Circular Proofs for Propositional Gentzen Calculus

by

Sander van Kasteel

student no. 3117839

supervised by

Rosalie Iemhoff

Universiteit Utrecht

September 2015

Acknowledgements

First of all, I would like to express my sincere gratitude to Rosalie Iemhoff for her guidance and contagious enthusiasm during the process of making this thesis. It was Rosalie who ignited within me the passion for both philosophical and mathematical logic in the first place. I am proud to call myself her student.

Furthermore, I would like to thank Albert Visser, Michael Moortgat and Jan Broersen for their teachings, which proved invaluable for the making of this thesis.

I am very grateful to Cees, Marianne, Mark and Esther for their continued support and reminding me of the important things in life. Be it from a house on the countryside of Zuid-Holland or from an apartment in the business district of Jakarta.

A special thanks to my friends Jeroen and Tom for providing me with a fruitful environment in the final weeks of writing, and René for his inspiring words in an hour of need.

Finally, every last little 'thank you' I have left go out to Noortje, without whom I would have surely slid down the rabbit hole. It is one thing to endure hearing the person next to you blab about formal logic, but to actually try to listen to and understand the gibberish takes a considerate amount of dedication. I still enjoy every moment of it and hope to continue returning the favor.

Contents

'Contrariwise,' continued Tweedledee, 'if it was so, it might be; and if it were so, it would be; but as it isn't, it ain't. That's Logic.'

– Lewis Carroll, *Through the Looking Glass*

Chapter 1

Introduction

Scientific research is a coin: heads is problem solving, tails is exploration. Problem solving is quite straightforward. We have a problem, generally phrased as *"I know of X, but I cannot explain X in my current framework"* . Thus, we exclaim "Let's use science!" and by putting our money where our mouth is, we hope to adjust our framework in such a way that we solve our problem. This is the bottom-up approach. Exploration, on the other hand, is the top-down approach. We have a framework and expand upon it to see where it takes us. The practical application of yielded results may not be immediately apparent, if they exist at all. Yet, if such an application is found it may turn out to be world changing. Furthermore, it is possible that at the end of our search we conclude that our framework is inconsistent and we thereby introduce a new problem to be solved.

Daniyar Shamkanov's *Circular Proofs for Gödel-Löb Logic* is an example of the latter method. In his paper, Shamkanov explores the possibility of expanding sequent-style proof systems by allowing circular proofs. He argues that this will allow us to express proofs derived by modal provability logic GL in proofs using only inference rules of standard modal logic K4. Even though the practical application of this theory remains to be seen, the notion of using circular proof structures in order to express a certain logic through rules of a weaker logic is intriguing indeed.

It should be noted that when considering the proof of Shamkanov's theory, two complications may arise. First of all, the sequent-style proof system used is Tait's system. Although this system is elegant in its few rules of inference, its documentation is sparse. Many logicians, myself included, feel more at home practicing sequential proofs through the use of the Gentzen system. Second, the proofs given by Shamkanov in the first three sections are fairly concise, thereby making them hard to understand for those without extensive experience in formal logic.

The desire to solve these complications is the main motivation of this thesis. My first goal is to give a proof that whenever Tait's system is used, it is equally possible to use Gentzen's system to obtain the same result. In Chapter 3, I present a proof that any logical expression provable in Tait can also be proved in Gentzen. Although this presents nothing new, it serves as a fine exercise to familiarise yourself with sequential calculi.

The second goal is to prove Shamkanov's assertion that every propositional formula that can be derived in GL by means of sequent calculus can also be derived by K4 by using a circular proof system. In his proof, Shamkanov takes a detour through the domain of infinite derivations. In Chapter 4 of this thesis, I shall present a different proof of his assertion by skipping infinite derivations altogether and translating immediately from standard to circular sequential proofs.

Chapter 2

Scientific Context

2.1 Logic and Artificial Intelligence

For ages, philosophers and scientists alike have tried to reduce the human mind to a mechanical system of some sort in order to better understand and otherwise operate upon its workings: from the clock in early days to binary computers today and, as some theorise, the quantum computer in the future. To craft human-like intelligence from such a system would be the most ambitious aim of the relatively young field of research that is Artificial Intelligence. But this ideal, known as *hard AI*, is rarely the subject of studies of those who interest themselves in this field. Examples of research in AI are such subjects as computer vision, speech technology and multi-agent systems. The latter has a direct link with formal logic, as reasoning about other agents' beliefs and knowledge and common knowledge is the scope of epistemic logic, a subfield of formal logic.

In [4, sec. 1.3], Richmond Thomason writes:

> The dominant goal (...) of philosophical logic is the extension of logical methods to nonmathematical reasoning domains. This goal has a theoretical dimension if (...) it requires reworking and extending logical formalisms.

Having a common goal in trying to formalise human reasoning and behaviour, AI and philosophic logic are two distinct fields of research with a flirtatious relationship. However, the methods in which they operate differ. In the entire period during which it has been practiced, the methodology of philosophical logic has been solely by thought experiments. This means that knowledge obtained by its practice is exclusively a priori: by reasoning within its framework, i.e. through formal logic. A successful application in the physical world is by no way guaranteed. The rise of computer science, and AI with it, brought a change to the way formal logic developed. Formal logic is the language in which computers (among other things) represent knowledge. Through computer simulations (based upon logical rules) computer science experiments with formal logic. If the results do not match the requirements, then either there is a problem with the representation of knowledge within the system or the rules are not sufficient for our goals. Et voilà: we obtained a method that yields a posteriori knowledge about formal logic.

In the specific case of AI, suppose we wish to create a system that acts in a certain way such that it can be described as 'intelligent'. We model its states and their relations based upon a system of formal logic that seems the most appropriate. If it acts exactly the way it was intended: great! But if it doesn't, we should consider whether the logical rules we used are 'correct' for the application (after checking for bugs of course). Subsequently, we can either stubbornly use

the same rules to catch exceptions or we alter the logical rules so that exceptions are already accounted for.

Compare this to formal logic: the strongest logic thinkable would be the one where every logical statement that is true is an axiom and it has no rules of inference. There are no exceptions in this system as everything has been accounted for. Furthermore, it is an unworkable monstrosity where every derivation is just a raw database crunch, and a big database it is! In logic, elegance lies not in power, but rather in using as little rules as possible to derive the highest amount of logical formulas. This is not unlike the goal of AI: intelligence does not lie in pre-defining what a system knows, but in asserting the rules with which newly obtained knowledge is analysed correctly.

In this way AI and logic have a symbiotic relationship with one another: AI relies heavily upon logic and on the other hand AI directs logic to the domain of human reasoning. For example, multi-agent systems required more than just classical logic to formalise a system of knowledge and belief. By expanding its formalisms, this led to the development of modal logic. It is the responsibility of formal logic to prove the consistency of newly proposed systems, while AI tests their effectiveness.

2.2 Circular Proofs and their Applications

The focus of this thesis is on the first three sections of *Circular Proofs for Gödel-Löb Logic*, in particular section 3 with its main theorem. For the purpose of this thesis, we can ignore the detour through the domain of regular ∞-derivations. Thus, through transitivity we obtain the short version of the theorem:

$$\mathsf{GL}_{\mathsf{Seq}} \vdash \Gamma \iff \mathsf{GL}_{\mathsf{circ}} \vdash \Gamma$$

In short, what Shamkanov expresses with this theorem is that a sequent has a GL-derivation[1] *if and only if* it has a circular $\mathsf{K4}$-derivation[2].

A circular derivation, as Shamkanov proposes, can be formed from an ordinary derivation tree by linking its non-axiomatic leafs with an identical interior node. The exact details of these back-links are discussed later. However, note that this means that these circular derivations no longer have a tree structure, but in fact the structure of a graph. The idea of deserting tree structures for formal proofs is not new. For example, *directed acyclic graphs* have been proposed as proof structures. The benefit of a *daglike* proof system is its increased efficiency, as derivations would require less steps (see [1, p. 162]). But as their name implies, these systems are different from Shamkanov's cyclic derivations at least in the sense that they do not allow cycles within their structures.

Until now, a formal proof system where not every branch of a derivation has to end at an axiom had not been seriously considered. In this, Shamkanov's proposal is interesting indeed. But why consider it at all? What are the applications? I think there are several. First of all, GL – being the logic of provability – plays a big role within computer science. For example, it is used to prove soundness and completeness of algorithms. Thereby, the influence of research in GL extends to developments in this field. As I am not a computer scientist myself, I cannot predict the influence that circular proofs may have for applications.

Second, I believe that Shamkanov's proposal may be of interest to AI in a philosophical sense. Thomason points out correctly in [4, sec. 1.5]:

[1] That is, it has a sequential proof though use of the axioms and rules of logic GL.

[2] Reader beware! Shamkanov uses $\mathsf{GL}_{\mathsf{circ}}$ to denote the domain of circular proofs using axioms and rules of logic $\mathsf{K4}$. To avoid confusion I chose $\mathsf{K4}_{\mathsf{circ}}$ to denote this domain for this thesis.

> (...) [L]ogic deals with reasoning – and relatively little of the reasoning we do is mathematical, while almost all of the mathematical reasoning that nonmathematicians do is mere calculation.

Allow me to extend this to the subject matter. Sequent-style proof systems are mainly used as a means to formalise mathematical proof. If we look at the structure of these systems, we see that the truth of a sequent is derived by inferring from axioms *and only from axioms*. But since we wish to extend proof systems to the non-mathematical domain, I am convinced we should ask ourselves the question: "How do we decide that we have provided sufficient arguments to consider a statement proved?" This question adresses a known philosophical problem that is discussed in the *Münchhausen Trilemma*. The trilemma recognises three patterns of reasoning:

- *Foundationalism*, which states that a proof is inferred solely from a foundation of basic beliefs (axioms). These basic beliefs need not be proved as they are uncontroversally true and independent of other beliefs.
- *Coherentism*, which states that proofs are structured like a web and the strength of a proof depends on the strength of the surrounding areas. This means that at some point within a proof, an argument is made that has already been considered. Thus, according to coherentism, reasoning is unavoidably circular.
- *Infinitism*, which states that every proof requires a further proof, which requires a further proof *ad infinitum* without repetition.

Sequent-based proof systems are undeniably foundationalistic in their nature, as every sequent in a correct sequent proof is is either inferred (transitively) from axioms or is an axiom itself. Thus, foundationalistic reasoning is a form of calculation. But as Thomason points out, human reasoning is rarely mathematical. It is hard to imagine that there are these atomic beliefs on which all of our reasoning is constructed. This is why coherentism is appealing as a pattern of reasoning: it rejects the idea of fundamental truths, while it retains a constructive approach by not giving in to notions of infinity. For further reading on foundationalism and coherentism and their arguments, I recommend [3, ch. 3].

So what does any of this have to do with circular proofs? I conjecture that adding circularity to a sequential proof system presents a logical method to the coherentist's account of reasoning. Of course, which one (if any) of the patterns of reasoning is still under debate. But if it is true that Shamkanov's circular proof system allows logicians and computer scientists to experiment with the notion of coherentism within sequential proofs, I believe there is much to learn about human reasoning on a logical level. That is what Artificial Intelligence is all about.

Chapter 3

Proof that we can use Gentzen instead of Tait

3.1 An introduction to Gentzen's Propositional Calculus

Sequent calculus was introduced in 1935 by Gerhard Gentzen as a formal representation of the derivability relation in natural deduction. The explanation of sequent calculus that follows here is gathered from [1, ch. 2], which I recommend to those who desire a more detailed account of the subject matter.

In this thesis we deal with Gentzen's System for propositional calculus, which is denoted $\boldsymbol{PK}$[1]. For propositional logic, formulas are built from the logical constants $\top$ and $\bot$, propositional variables $P_1, P_2, P_3, \ldots$, logical connectives $\vee$, $\wedge$ and $\neg$[2], and parentheses (,). Let us give a formal definition of the formulas used in propositional logic.

Definition 3.1. A *propositional formula* is built according to these rules:

- $\bot, \top, p$ are formulas for any variable p.
- If A and B are formulas, then so are $(A \vee B)$, $(A \wedge B)$ and $\neg A$.

Sequents are the entities on which propositional sequent calculus operates. The structure of sequents depend on the style of sequent calculus that we use. We define Gentzen style sequents here, while those in Tait style are dealt with in the next section.

Definition 3.2. A *sequent* in $\boldsymbol{PK}$ is of the form $(\Gamma \Rightarrow \Delta)$, where Γ and Δ are finite multisets of propositional formulas, where Γ is called the *antecedent* and Δ is called the *consequent*. $\Rightarrow$ is a new symbol. A sequent evaluates to true iff any formula in the antecedent evaluates to false or any formula in the consequent evaluates to true.

Example 3.3. The $\boldsymbol{PK}$-sequent

$$A_1, A_2, A_2 \Rightarrow B_1, B_2, B_3$$

evaluates to true iff the propositional formula

$$\neg (A_1 \wedge A_2 \wedge A_3) \vee B_1 \vee B_2 \vee B_3$$

[1] $\boldsymbol{P}$ stands for *propositional*, while $\boldsymbol{K}$ comes from the German word *klassisch* from classical logic.

[2] The implication connective $\rightarrow$ is not allowed, but we take $(A \rightarrow B)$ to stand for $(\neg A \vee B)$ for any formula A, B.

evaluates to true. Note that when we allow the implication connective to our notation, the formula

$$A_1 \wedge A_2 \wedge A_3 \to B_1 \vee B_2 \vee B_3$$

is logically equivalent.

Sequent calculi are proof systems, i.e. systems that construct proofs. Now let us examine the notion of proof as it applies to sequent calculi. The version of the sequent system in the following definition is intentionally left undefined, as it applies to both $\boldsymbol{PK}$ and $\boldsymbol{PT}$. The only difference being the axioms and inference rules.

Definition 3.4. A *proof* of a sequent S is a finite tree whose nodes are (labeled with) sequents, whose root (called *endsequent*) is S and is written at the bottom, whose leaves (or *initial sequents*) are logical axioms, such that each non-leaf sequent follows from the sequent(s) immediately above by one of the rules of inference as defined in the paragraphs for the respective proof system.

The *length* of a proof is defined as the amount of nodes in the longest path from the root to leaves of the tree. The *height of a node* is then defined as the distance from that node to the root of the proof. Furthermore, the *height of a rule* is defined as the height of the node that is the conclusion of that rule.

Inference rules and axioms of Gentzen Calculus The *logical axioms* are of the form

$$\mathsf{Ax}: \ \Gamma, A \Rightarrow A, \Delta \qquad\qquad \bot\mathsf{L}: \ \Gamma, \bot \Rightarrow \Delta$$

where A is any formula. The rules of inference are as follows (here Γ and Δ denote finite sequences of formulas).

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

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

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

$$\neg\mathsf{L}\ \frac{\Gamma \Rightarrow A, \Delta}{\Gamma, \neg A \Rightarrow \Delta} \qquad \neg\mathsf{R}\ \frac{\Gamma, A \Rightarrow \Delta}{\Gamma \Rightarrow \neg A, \Delta}$$

$$\Box_{\mathsf{K4}}\ \frac{\Box\Pi, \Pi \Rightarrow A}{\Gamma, \Box\Pi \Rightarrow \Box A, \Delta} \qquad \Box_{\mathsf{GL}}\ \frac{\Box\Pi, \Pi, \Box A \Rightarrow A}{\Gamma, \Box\Pi \Rightarrow \Box A, \Delta}$$

Note that structural weakening has been incorporated in the inference rules for modal operators. For a representative example of a $\boldsymbol{PK}$-proof, see (4.2.1).

We shall use $\mathsf{GL}_{\boldsymbol{PK}}$ to denote the system of sequent style proofs using all axioms and right and left inference rules of logical operators for $\boldsymbol{PK}$ plus rule $\Box_{\mathsf{GL}}$. Similarly, we use $\mathsf{K4}_{\boldsymbol{PK}}$ to denote sequent system with the same axioms and rules, except we add rule $\Box_{\mathsf{K4}}$ instead of $\Box_{\mathsf{GL}}$.

3.2 Distinct properties of Tait's Propositional Calculus

Tait's style of calculus is largely similar to Gentzen's with some exceptions which we discuss in this section. We shall use $\boldsymbol{PT}$ to denote Tait's propositional calculus.

Definition 3.5. A *sequent* in $\boldsymbol{PT}$ is a finite multiset of propositional formulas. A $\boldsymbol{PT}$-sequent evaluates to true iff at least one formula within the multiset evaluates to true.

Example 3.6. The $\boldsymbol{PT}$-sequent

$$A, B, C$$

evaluates to true iff the propositional formula

$$A \vee B \vee C$$

evaluates to true.

As I announced earlier, the definition of $\boldsymbol{PT}$-proofs is indifferent from Definition 3.4, with the exception of the axioms and rules of inference. Those of $\boldsymbol{PT}$ are given below and are gathered from [2, p. 3].

Inference rules and axioms of Tait Calculus The *logical axioms* are of the form

$$\mathsf{Ax} : \Gamma, A, \neg A \qquad\qquad \top : \Gamma, \top$$

where A is any formula. The rules of inference are as follows (here Γ denotes a finite sequence of formulas).

$$\wedge \frac{\Gamma, A \qquad \Gamma, B}{\Gamma, A \wedge B} \qquad\qquad \vee \frac{\Gamma, A, B}{\Gamma, A \vee B}$$

$$\Box_{\mathsf{K4}} \frac{\Gamma, \Diamond\Gamma, A}{\Diamond\Gamma, \Box A, \Delta} \qquad\qquad \Box_{\mathsf{GL}} \frac{\Gamma, \Diamond\Gamma, \Diamond\neg A, A}{\Diamond\Gamma, \Box A, \Delta}$$

As with $\boldsymbol{PK}$, structural weakening has been incorporated in $\boldsymbol{PT}$-rules $\Box_{\mathsf{K4}}$ and $\Box_{\mathsf{GL}}$.

Where Shamkanov uses $\mathsf{GL}_{\mathsf{Seq}}$, we shall use $\mathsf{GL}_{\boldsymbol{PT}}$ to denote Tait style sequent system with the axioms, inference rules for $\vee$ and $\wedge$ and rule $\Box_{\mathsf{GL}}$. The reason we do this is to avoid confusion with Gentzen's system.

3.3 From Tait to Gentzen

Now that we have an understanding of the workings of propositional sequent calculi and know how the styles of Gentzen and Tait differ from each other, we can explore how the styles are equivalent. More precisely, we can prove that $\boldsymbol{PK}$ is at least as powerful as $\boldsymbol{PT}$. Let us define our hypothesis formally:

Theorem 3.7. If multiset Δ is provable in $\boldsymbol{PT}$, then its corresponding sequent $(\Rightarrow \Delta)$ is provable in $\boldsymbol{PK}$.

First off, we need a definition in order to identify $\boldsymbol{PT}$-style sequents (which are multisets) with $\boldsymbol{PK}$-style sequents.

Definition 3.8. If Δ is some multiset, then $(\Rightarrow \Delta)$ is its corresponding sequent.

Furthermore, remember that the following equivalences hold by convention:

$$\begin{array}{rlrl} \top \equiv \neg\bot & \text{and conversely} & \bot & \equiv \neg\top \\ \Box A \equiv \neg\Diamond\neg A & \text{and conversely} & \Diamond A & \equiv \neg\Box\neg A \end{array}$$

Without further ado, let's get to the action!

Proof of Theorem 3.8. The theorem is proved by induction on length of proofs. We use n to denote the length of a proof. For our base case, we regard proofs in Tait with $n = 0$. These are the axioms. Suppose Δ is a multiset provable by a $\boldsymbol{PT}$-axiom, thus $\vdash_{\boldsymbol{PT}} \Delta$ holds. By Definition 3.8, Δ has a corresponding sequent $(\Rightarrow \Delta)$. If we can derive $(\Rightarrow \Delta)$ in Gentzen, $\vdash_{\boldsymbol{PK}} (\Rightarrow \Delta)$ holds. Tait has two axioms: Ax and $\top$ of which the corresponding sequents are derived by (3.3.1) and (3.3.2) respectively. Hence, for every possible $\boldsymbol{PT}$-derivation tree of length $n = 0$ we proved that there is a derivation in Gentzen.

$$\neg\,\mathsf{R}\,\frac{A \Rightarrow \Gamma, A}{\Rightarrow \Gamma, A, \neg A} \qquad (3.3.1) \qquad\qquad \neg\,\mathsf{R}\,\frac{\bot \Rightarrow \Gamma}{\Rightarrow \Gamma, \top} \qquad (3.3.2)$$

Now consider a $\boldsymbol{PT}$-derivation tree of length $n+1$. Suppose $\Delta_1, \ldots, \Delta_k$ are the assumptions of the lowest inference and their corresponding sequents $(\Rightarrow \Delta_1), \ldots, (\Rightarrow \Delta_k)$ are provable in Gentzen. In order to prove that the tree of length $n + 1$ is also provable in Gentzen we must consider every applicable $\boldsymbol{PT}$-rule and prove that if the corresponding sequent of its assumption is provable in Gentzen, the corresponding sequent of its conclusion is also provable in Gentzen. The corresponding sequents for $\boldsymbol{PT}$-rules $\vee$, $\wedge$, $\Box_{\mathsf{K4}}$ and $\Box_{\mathsf{GL}}$ are derived by (3.3.3), (3.3.4), (3.3.5) and (3.3.6) respectively. $\Box$

$$\vee\,\mathsf{R}\,\frac{\begin{array}{c}\boldsymbol{\pi} \\ \Rightarrow \Gamma, A, B\end{array}}{\Rightarrow \Gamma, A \vee B} \qquad (3.3.3) \qquad\qquad \wedge\,\mathsf{R}\,\frac{\begin{array}{c}\boldsymbol{\pi_1} \\ \Rightarrow \Gamma, A\end{array} \qquad \begin{array}{c}\boldsymbol{\pi_2} \\ \Rightarrow \Gamma, B\end{array}}{\Rightarrow \Gamma, A \wedge B} \qquad (3.3.4)$$

$$\begin{array}{r} (\neg\,\mathsf{R})^{|\Gamma|} \\ \Box_{\mathsf{K4}} \\ (\neg\,\mathsf{L})^{(2\cdot|\Gamma|)} \end{array} \frac{\begin{array}{c}\boldsymbol{\pi} \\ \Rightarrow \Gamma, \neg\Box\neg\Gamma, A\end{array}}{\begin{array}{c}\Box\neg\Gamma, \neg\Gamma \Rightarrow A \\ \hline \Box\neg\Gamma \Rightarrow \Box A, \Delta \\ \hline \Rightarrow \neg\Box\neg\Gamma, \Box A, \Delta\end{array}} \qquad (3.3.5) \qquad \begin{array}{r} \neg\,\mathsf{L} \\ (\neg\,\mathsf{L})^{(2\cdot|\Gamma|)} \\ \Box_{\mathsf{GL}} \\ (\neg\,\mathsf{R})^{|\Gamma|} \end{array} \frac{\begin{array}{c}\boldsymbol{\pi} \\ \Rightarrow \Gamma, \neg\Box\neg\Gamma, \neg\Box A, A\end{array}}{\begin{array}{c}\Box A \Rightarrow \Gamma, \neg\Box\neg\Gamma, A \\ \hline \Box\neg\Gamma, \neg\Gamma, \Box A \Rightarrow A \\ \hline \Box\neg\Gamma \Rightarrow \Box A, \Delta \\ \hline \Rightarrow \neg\Box\neg\Gamma, \Box A, \Delta\end{array}} \qquad (3.3.6)$$

To be precise, the converse of Theorem 3.7 also holds. That is to say, every sequent provable in Gentzen corresponds to a multiset that is provable in Tait. Therefore both systems are equally

powerful. However, for our purpose it is sufficient to know that Gentzen is at least as powerful as Tait. Now that that we have proved that this is the case, we can begin our investigation of circular derivations using Gentzen calculus.

Chapter 4

Transforming sequential proofs from regular GL to circular K4

4.1 The main theorem

First off, let us give a formal description of the theorem we try to prove.

Theorem 4.1. If formula Γ is provable in $\mathsf{GL}_{\boldsymbol{PK}}$, then it is provable in $\mathsf{K4}_{\mathsf{circ}}$. In terms of formal logic:

$$\mathsf{GL}_{\boldsymbol{PK}} \vdash \Gamma \Longrightarrow \mathsf{K4}_{\mathsf{circ}} \vdash \Gamma$$

In the previous section we explored $\mathsf{GL}_{\boldsymbol{PK}}$: it is the domain of $\boldsymbol{PK}$-derivations in logic GL. Before we get anywhere near proving our theorem, we also have to consider $\mathsf{K4}_{\mathsf{circ}}$, since so far our notion of circular sequential derivations (in any logic) has been rather vague. Luckily, Shamkanov provided an excellent definition of circular proofs for us in [2, p. 4].

Definition 4.2. A *circular derivation* is a pair (κ, d), where κ is an ordinary derivation and d is a *back-link function* assigning to some leaf x an interior node y with an identical sequent such that y lies on the path from the root to the leaf. A *circular proof* is a circular derivation such that every leaf is either connected by the back-link function, or is marked by an initial sequent.

The *domain* $\mathsf{K4}_{\mathsf{circ}}$, is then obtained from the domain of sequential proofs in standard modal logic K4, denoted $\mathsf{K4}_{\boldsymbol{PK}}$, by admitting circular proofs in K4.

This is all well and good, but we are still in the dark if we lack a clear idea of what it means for two sequents to be 'identical'. In order to shine a light on the subject, we require the notion of underlying sets, defined below.

Definition 4.3. For any multiset of formulas Γ the *underlying set*, denoted $\mathsf{Set}(\Gamma)$, is the regular set of its formulas. Furthermore, the underlying set of a $\boldsymbol{PT}$-sequent $\Gamma \Rightarrow \Delta$, denoted $\mathsf{Set}(\Gamma \Rightarrow \Delta)$, is the $\boldsymbol{PT}$-sequent with antecedent $\mathsf{Set}(\Gamma)$ and consequent $\mathsf{Set}(\Delta)$. In terms of formal logic:

$$\mathsf{Set}(\Gamma) := \{A \mid A \in \Gamma\}$$
$$\mathsf{Set}(\Gamma \Rightarrow \Delta) := \mathsf{Set}(\Gamma) \Rightarrow \mathsf{Set}(\Delta)$$

We shall now use the notion of underlying sets to specify the concept of sequent identity.

Definition 4.4. Two sequents S_1, S_2 are said to be identical iff they have the same underlying sets. In terms of formal logic:

$$S_1 \equiv S_2 \iff \mathsf{Set}(S_1) = \mathsf{Set}(S_2)$$

Now that we have a clear image of our subject matter, we can begin our proof. First, we shall explore a method to transform sequential derivations from provability logic GL to standard modal logic $\mathsf{K4}$. Subsequently, we argue that if this method fails to deliver, we can resolve it by assigning the back-link function. If this succeeds, we can conclude that we can send any proof in $\mathsf{GL}_{\boldsymbol{PK}}$ to a proof in $\mathsf{K4}_{\mathsf{circ}}$, thus proving Theorem 4.1.

4.2 Proof transformation: an algorithm

Below I present an algorithm that we shall use for the transformation of GL-proofs to $\mathsf{K4}$-proofs.

$\mathscr{T}$ = "On input derivation tree π:

1. Find the lowest, left-most[1]application of rule $\Box_{\mathsf{GL}}$ in π. If no such application is found, **return** π.
2. In π replace the application found at the previous step as indicated in figure 4.1. Name the newly formed derivation tree π'.
3. **return** π'."

Let us briefly examine the workings of algorithm $\mathscr{T}$. It uses a left-to-right breadth-first search algorithm to determine the lowest, left-most application of rule $\Box_{\mathsf{GL}}$. If such an application is found, it is replaced by an application of rule $\Box_{\mathsf{K4}}$. To complete the transformation, it creates a derivation tree for the assumption of the newly applied rule using only GL-rules. That this can be done follows from Proposition 4.5, which in turn follows immediately from Proposition 2.1 in [2, p. 3]. Verification is left to the reader.

Proposition 4.5. If $\vdash_{\mathsf{GL}} (\Box\Pi, \Pi, \Box A \Rightarrow A)$ holds, then $\vdash_{\mathsf{GL}} (\Box\Pi, \Pi \Rightarrow A)$ also holds.

Note that transformation by algorithm $\mathscr{T}$ only affects the applied rule and its underlying subtree. The rest of the derivation tree remains unchanged. Furthermore, by plugging in the derivation tree τ' new applications of rule $\Box_{\mathsf{GL}}$ (if any) will occur higher than the rule that has just been replaced.

By applying algorithm $\mathscr{T}$ iteratively, we replace $\Box_{\mathsf{GL}}$-applications one by one. When, at some iteration during this run, the yielded derivation tree did not change we stop the process. We shall use $\mathscr{T}^*$ to denote this process of proof transformation. First, let us explore the possibility of a succesful transformation of a proof by $\mathscr{T}^*$.

Lemma 4.6. Whenever running $\mathscr{T}^*$ on a proof in $\mathsf{GL}_{\boldsymbol{PK}}$ terminates, the resulting proof is in $\mathsf{K4}_{\mathsf{circ}}$.

Proof. Let us examine the way $\mathscr{T}^*$ works on an arbitrary derivation tree π. There are two options: either running $\mathscr{T}^*$ on π terminates or it does not. If we assume it terminates, as the lemma assumes, we have a finite chain of constructed trees $\pi \hookrightarrow \pi' \hookrightarrow \pi'' \hookrightarrow \ldots \hookrightarrow \pi_{\mathsf{K4}}$ where π_{K4} is a finite derivation tree in which only $\mathsf{K4}$-rules occur. In this case π_{K4} is a member of $\mathsf{K4}_{\boldsymbol{PK}}$. We know from Definition 4.2 that $\mathsf{K4}_{\boldsymbol{PK}}$ is a subdomain of $\mathsf{K4}_{\mathsf{circ}}$, therefore π_{K4} must also a member of $\mathsf{K4}_{\mathsf{circ}}$. □

[1]Priority in that order.

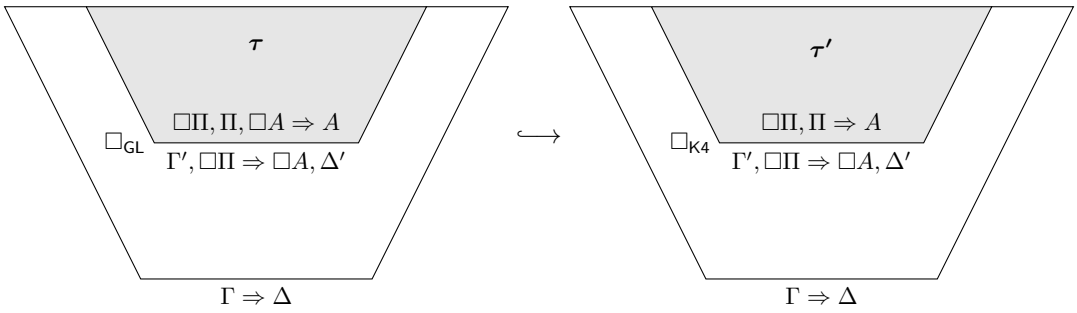

Figure 4.1: replacing a $\Box_{\mathsf{GL}}$-application with a $\Box_{\mathsf{K4}}$-application

What if running $\mathscr{T}^*$ on π does not terminate? This can only happen if from some point on in the process, $\Box_{\mathsf{GL}}$-applications are continuously replaced through introduction of one or more new applications of rule $\Box_{\mathsf{GL}}$. This is a problem, because we want the process to terminate, returning a well-formed $\mathsf{K4}$-derivation tree as output. Obviously, $\mathscr{T}^*$ is not complete in relation to all proofs in relation to $\mathsf{GL}_{\boldsymbol{PK}}$. If we consider that GL is a stronger logic than $\mathsf{K4}$, we understand why this is so.

Example 4.7. Consider (4.2.1) to be the GL-derivation tree that we wish to transform by iterative application of algorithm $\mathscr{T}$. One iteration yields derivation (4.2.2), while any number of iterations greater than 3 yields derivation (4.2.3). We see that the sequent $(\Box(\Box P \to P) \Rightarrow \Box P, P)$ repeats itself and our algorithm tries to derive it by applying rule $\Box_{\mathsf{K4}}$ and deriving its assumption, which results in a branch with sequent $(\Box(\Box P \to P) \Rightarrow \Box P, P)$, to which rule $\Box_{\mathsf{GL}}$ is applied. Therefore, this process will not terminate.

$$\to \mathsf{L}\,\frac{\Box(\Box P \to P), \Box P \Rightarrow \Box P, P \qquad \Box(\Box P \to P), \Box P, P \Rightarrow P}{\Box_{\mathsf{GL}}\,\dfrac{\Box(\Box P \to P), \Box P \to P, \Box P \Rightarrow P}{\to \mathsf{R}\,\dfrac{\Box(\Box P \to P) \Rightarrow \Box P}{\Rightarrow \Box(\Box P \to P) \to \Box P}}} \tag{4.2.1}$$

$$\to \mathsf{L}\,\frac{\Box(\Box P \to P), \Box P \Rightarrow \Box P, P \quad \Box(\Box P \to P), \Box P, P \Rightarrow P}{\Box_{\mathsf{GL}}\,\dfrac{\Box(\Box P \to P), \Box P \to P, \Box P \Rightarrow P}{\to \mathsf{L}\,\dfrac{\Box(\Box P \to P) \Rightarrow \Box P, P \qquad \Box(\Box P \to P), P \Rightarrow P}{\Box_{\mathsf{K4}}\,\dfrac{\Box(\Box P \to P), \Box P \to P \Rightarrow P}{\to \mathsf{R}\,\dfrac{\Box(\Box P \to P) \Rightarrow \Box P}{\Rightarrow \Box(\Box P \to P) \to \Box P}}}} \tag{4.2.2}$$

$$
\to\mathsf{R}\;\dfrac{\Box_{\mathsf{K4}}\;\dfrac{\to\mathsf{L}\;\dfrac{\Box_{\mathsf{K4}}\;\dfrac{\to\mathsf{L}\;\dfrac{\Box_{\mathsf{K4}}\;\dfrac{\dfrac{\text{[see (4.2.4)]}}{\Box(\Box P \to P), \Box P \to P \Rightarrow P}}{\Box(\Box P \to P) \Rightarrow \Box P, P} \qquad \Box(\Box P \to P), P \Rightarrow P}{\Box(\Box P \to P), \Box P \to P \Rightarrow P}}{\Box(\Box P \to P) \Rightarrow \Box P, P} \qquad \Box(\Box P \to P), P \Rightarrow P}{\Box(\Box P \to P), \Box P \to P \Rightarrow P}}{\Box(\Box P \to P) \Rightarrow \Box P}}{\Rightarrow \Box(\Box P \to P) \to \Box P} \tag{4.2.3}
$$

$$
\Box_{\mathsf{GL}}\;\dfrac{\to\mathsf{L}\;\dfrac{\Box(\Box P \to P), \Box P \Rightarrow \Box P, P \qquad \Box(\Box P \to P), \Box P, P \Rightarrow P}{\Box(\Box P \to P), \Box P \to P, \Box P \Rightarrow P}}{\Box(\Box P \to P) \Rightarrow \Box P, P} \tag{4.2.4}
$$

4.3 Proof that we can assign back-link functions if our transformation process does not halt

As suggested in the previous section, we will explore the results of $\mathscr{T}^*$ when it does not halt. We shall do this by proving the following theorem.

Theorem 4.8. Whenever running $\mathscr{T}^*$ on a proof π in $\mathsf{GL}_{\boldsymbol{PK}}$ does not terminate, after some number of iterations, we can construct a circular proof from it by assigning a back-link functions.

Before we examine this theorem any further we take an apparent sidetrack. We will consider the height of the lowest $\Box_{\mathsf{GL}}$-application in a proof and examine the influence of running $\mathscr{T}$ on such a proof. First, we shall assign to each derivation a pair to identify its relevant properties.

Definition 4.9. Find the lowest $\Box_{\mathsf{GL}}$-application in the derivation tree π. Let d denote the height of that rule. Let n denote the number of $\Box_{\mathsf{GL}}$-applications at height d. For every derivation that contains a $\Box_{\mathsf{GL}}$-application we assign a pair (d, n).

The effect of running $\mathscr{T}$ on a proof in terms of its pair is then described by the following lemma.

Lemma 4.10. Consider running $\mathscr{T}$ on a proof π with at least one $\Box_{\mathsf{GL}}$-application and associated pair (d, n), which transforms it to a new proof π' with an associated pair (e, m), where e is the updated value of d and m is the updated value of n. Then one of the following properties is true:

1. either $e = d$ and $0 < m < n$;
2. or $d < e$.

Proof. Let π be some proof with at least one $\Box_{\mathsf{GL}}$-application and let (d, n) be its associated pair. After we run $\mathscr{T}$ on π, we obtain π' whose associated pair we denote (e, m). Running $\mathscr{T}$ replaces the lowest, left-most $\Box_{\mathsf{GL}}$-application of π. This leaves us with two possibilities: either that application was the only one of its kind at that height or it was not.

Consider the case where it was not the only $\Box_{\mathsf{GL}}$-application on height d. Then e must be equal to d, as $\mathscr{T}$ transforms only one such application. Furthermore, $n > 1$ must hold and since $m = n - 1$, $0 < m < n$ must hold.

Now consider the case where the transformed application was the only $\Box_{\mathsf{GL}}$-application at height d. Then π' has no $\Box_{\mathsf{GL}}$-applications on height d. Since we use a breadth-first search algorithm, the next application must occur higher, so $d < e$ must hold. □

What we actually learn from Lemma 4.10 about the effect of running $\mathscr{T}$ on a proof in relation to the height of its lowest $\Box_{\mathsf{GL}}$-application, is that the lowest $\Box_{\mathsf{GL}}$-application of the newly obtained proof occurs at the same height or higher. This property will prove useful later.

Example 4.11. Consider (4.3.1), where the height of the lowest node labeled with a conclusion of a $\Box_{\mathsf{GL}}$-application is 3, while the amount of $\Box_{\mathsf{GL}}$-applications with their conclusions being nodes at that height is 2, so the pair associated with this proof is $(3, 2)$. If we run $\mathscr{T}$ on this proof, the lowest, left-most application of rule $\Box_{\mathsf{GL}}$ (the one with S_3 as its conclusion) is replaced. The height of the lowest $\Box_{\mathsf{GL}}$-application remains unchanged, but the number of $\Box_{\mathsf{GL}}$-applications is reduced by one. Therefore, the pair associated with the new proof is $(3, 1)$.

Note that we cannot be sure what the pair will be after applying $\mathscr{T}$ to the new proof with pair $(3, 1)$. After all, we are guaranteed that the newly formed subtree with endsequent S_3 is different from its predecessor. It is entirely possible that no $\Box_{\mathsf{GL}}$-application will occur higher up in the subtree at all.

$$\dfrac{\Box_{\mathsf{GL}}\dfrac{\dfrac{\vdots}{S_7}\quad\dfrac{\vdots}{S_8}}{\Box_{\mathsf{GL}}\dfrac{S_5}{S_3}}\qquad\dfrac{\dfrac{\vdots}{S_9}\quad\Box_{\mathsf{GL}}\dfrac{\vdots}{S_{10}}}{\Box_{\mathsf{GL}}\dfrac{S_6}{S_4}}}{\dfrac{S_2}{S_1}} \tag{4.3.1}$$

Proposition 4.12. Suppose we have an infinite chain of proofs $\pi \hookrightarrow \pi' \hookrightarrow \pi'' \hookrightarrow \ldots \hookrightarrow \pi_i \hookrightarrow \ldots$, generated by running $\mathscr{T}^*$ on a proof π without termination. Then for every integer m there is some proof π_i with pair (d_i, n_i) such that $d_i > m$.

Proposition 4.12 follows immediately from Lemma 4.10 and verification is left to the reader.

With the help of Proposition 4.12, we are almost ready to prove Theorem 4.8. Before we begin however, we require the notion of subformulas, defined below.

Definition 4.13. For a formula A, the *set of subformulas* of A (notation $\mathsf{Sub}(A)$) is defined by:

- A is a subformula of A.
- If $B \circ C$ is a subformula of A, then so are B, C for $\circ = \vee, \wedge, \rightarrow$

For a sequent $\Gamma \Rightarrow \Delta$, the *set of subformulas* of $(\Gamma \Rightarrow \Delta)$ is defined by:

$$\mathsf{Sub}(\Gamma \Rightarrow \Delta) := \bigcup \{\mathsf{Sub}(A) \mid A \in \Gamma \cup \Delta\}$$

Let us examine Theorem 4.8 further. By Definition 4.2, we know that we can assign a back-link function to two different nodes in a single path in a derivation when these nodes are identical. With the concept of subformulas, we can now prove that such an identity must occur at some point during non-halting $\mathscr{T}^*$.

Proof of Theorem 4.8. Let π be a $\boldsymbol{PK}$-proof of a sequent S for which our transformation process does not terminate. Let k denote the size of the set of subformulas of S, i.e. $|\mathsf{Sub}(S)| = k$. Now we can make two obervations:

1. In any deduction of a sequent $(\Gamma \Rightarrow \Delta)$, only subformulas of Γ and Δ occur[2].
2. There are at most 2^{2k} sequents $S_1, ..., S_{2^{2k}}$ such that $\forall i \forall j (i \neq j \rightarrow \mathsf{Set}(S_i) \neq \mathsf{Set}(S_j))$.

Now let $m = 2^{2k}+1$. We choose some π' constructed by the process with associated pair (d, n), such that $d > m$, i.e. the sequent that is the conclusion of the lowest left-most GL-application has height d. Now every branch in our derivation tree either ends in an axiom or its length is greater than m, in which case it must contain sequents S_i, S_j such that $\mathsf{Set}(S_i) = \mathsf{Set}(S_j)$ by the pigeonhole principle. Let S_j be the sequent at the higher node. By Definition 4.4, this means that S_i and S_j are identical. Subsequently, Definition 4.2 tells us that we can then use the back-link function to assign S_i to S_j, making S_j a leaf in the derivation tree. If we do this for every such branch, then what we are left with is a proof tree with circularity that contains no GL-applications. Therefore, our proof is in $\mathsf{K4}_{\mathsf{circ}}$. □

[2] This is an intrinsic property of our sequent calculus, called *subformula property* in [5, p. 66].

4.4 Proof of our main theorem

Let us see how our work thus far applies to Theorem 4.1. In the proof of Lemma 4.6 we pointed out that transforming a proof in $\mathsf{GL}_{\boldsymbol{PK}}$ with $\mathscr{T}^*$ yields two options: either it terminates or it does not. By Lemma 4.6, we know that if the process terminates, the resulting proof is in $\mathsf{K4}_{\mathsf{circ}}$. By Lemma 4.8, we know that if it does not terminate, after some number of iterations we can use Shamkanov's back-link function to construct a circular proof in $\mathsf{K4}_{\mathsf{circ}}$. Thus, every proof in $\mathsf{GL}_{\boldsymbol{PK}}$ can be transformed into a proof in $\mathsf{K4}_{\mathsf{circ}}$, thereby proving our theorem. □

Chapter 5

Conclusion and Discussion

5.1 Revision of the goals

Let us recapitulate the goals of this thesis:

- Reprove that Gentzen's Propositional Calculus is at least as powerful as Tait.
- Use Gentzen's Propositional Calculus to prove that all formulas with a standard sequential GL-derivation can also be derived by means of circular K4.

In Chapter 3.7 we successfully proved by induction that all formulas that are provable in Tait's Propositional Calculus are also provable in Gentzen's Propositional Calculus. Thereby we justified the use of Gentzen for the proof of the second goal.

In Chapter 4 we examined the definitions given by Shamkanov and adjusted them for the purpose of Gentzen Calculus where needed. We then examined an algorithm that transformed proofs from one domain to another and showed that it did not satisfy our needs on itself as it was not complete. Subsequently we showed that by applying Shamkanov's notion of back-link functions, we are able to fill in the gaps of our algorithm. Thereby we proved that for every proof in provability logic GL, we can construct a proof in standard modal logic K4 by admitting circularity to our derivations.

5.2 Recommendations

As I described in the introduction, the scope of this thesis was the first three sections of Daniyar Shamkanov's paper *Circular Proofs for Gödel-Löb Logic*. Furthermore, we only proved one direction of Shamkanov's theorem. Though this proof is interesting in itself, it would be satisfying to see the theorem proved in the other direction for Gentzen Calculus as well.

Furthermore, Shamkanov dives deeper into the subject in his fourth section. Due to limited resources (time and knowledge), I have not considered that part of his paper. It would undoubtedly be interesting to see how it applies to Gentzen Calculus.

Lastly, I am curious to see if and especially how Circular Proofs will affect Artificial Intelligence and Computer Science. In Section 2.2 I conjectured that it could provide us a fresh new angle to the subject of reasoning. Although it would have been interesting to explore this idea further, proving the main theorem turned out to be a lot more technical than I anticipated. Unfortunately, this left me with too little time for the deeper philosophical aspect of the subject. Come what may, I am optimistic.

Chapter 6

Bibliography

[1] S. Cook and P. Nguyen, *Logical Foundations of Proof Complexity*, Cambridge University Press, First ed., 2010.

[2] D. S. Shamkanov, *Circular Proofs for Gödel-Löb Logic*, Mathematical Notes, 96 (2014), pp. 575–585.

[3] M. Steup, *Epistemology*, in The Stanford Encyclopedia of Philosophy, E. N. Zalta, ed., Spring 2014 ed., 2014.

[4] R. Thomason, *Logic and Artificial Intelligence*, in The Stanford Encyclopedia of Philosophy, E. N. Zalta, ed., Winter 2014 ed., 2014.

[5] A. S. Troelstra and H. Schwichtenberg, *Basic Proof Theory*, Cambridge University Press, Second ed., 2000.