Utrecht University

Graduate School of Natural Sciences
Artificial Intelligence

Schütte Countermodels and Sequent Calculi

Using the Schütte method to compare different sequent calculi on countermodel construction

Master Thesis

Author:
Michiel Braat
Anr: 6079881
ECTS-credits: 45

Supervisor:
Prof. dr. Rosalie Iemhoff
Second Examiner:
Dr. Giuseppe Greco

28 February 2019

Abstract

In this thesis the Schütte style completeness proof is used as a tool to compare different sequent calculi. The Schütte style countermodel construction is the most important part of this kind of completeness proof. This countermodel construction method is used to compare three different sequent systems for the modal logics K, T, $\mathsf{K4}$, and $\mathsf{S4}$: A basic modal sequent calculus system, a tree-hypersequent system and a labeled sequent system. In the first part of the thesis we define what is meant with a Schütte countermodel and the question is answered whether the possible Schütte style countermodels produced using these sequents calculi are exactly the same sets of countermodels for each type calculus or whether it depends on the sequent system what models can be constructed. It is proven that each Schütte countermodel that can be constructed in one sequent system can also be constructed in the other types of systems. This shows that the set of constructable Schütte countermodels for these calculi are the same. As an additional part of the thesis, we explore Schütte countermodel construction for the intuitionistic modal logic $\mathbf{DY}_K$.

Contents

Chapter 1

Introduction

1.1 The Schütte style completeness proof

Not being able to find a proof for a proposition and finding a counterexample are generally two different things. If it is not possible to find proof for something using some method, it does not guarantee there exists a counterexample to it. However, in the case of symbolic logic, this relation can sometimes be proven. For analytic calculi, the completeness theorem explicitly states the duality between either being able to construct a proof with the calculus for a proposition or the existence of a countermodel. An analytic calculus is complete for a certain semantics if every tautology in that semantics can be proven using that analytic calculus. Consequently, the modus tollens of completeness states that every formula that cannot be proven in the analytic calculus cannot be a tautology in that semantics, which means that there exists a countermodel to the formula.

A completeness proof for an analytic calculus, together with soundness, is important to show the adequacy of the calculus for a semantics. However, completeness proofs are not always constructive. They do not show how to *find* a proof for every proposition and when one should give up the search for such a proof because the proposition is not a tautology. An example is the Henkin style proof which "gives no way to obtain derivability from validity, nor does it show how to construct a countermodel for underivable propositions."[19, p.2].

For sequent calculi, however, there is a style of completeness proof that is constructive. It shows how one can construct countermodels from a failed proof search. This is the "Schütte style" completeness proof, in which a combination of the sequents in a failed derivation can be used to create a countermodel. Because of this close relation between derivations, sequents and countermodels in the Schütte proof there has been renewed interest in this style of completeness proof for sequent calculi [21], also in combination with sequent systems for multi-modal logics [10].

The Schütte style completeness proof can be found scattered around in the literature, but they are not generally named as such. Examples where we can find them are [8, 17, 21, 23, 25]. However, its origin can be found earlier. In the 1950's this kind of proof was first given for first-order logic independently by Beth [3], Hintikka [13], Kanger [15] and Schütte [24]. Naming these completeness proofs 'Schütte style' is not something that is widely used in literature. It can for example also be found in literature named as a weak proof [6] or the standard 'tree' or 'tableaux' completeness proof [25], referring to the importance of the tree structure of sequent derivation in the completeness proof and the usage of it for tableaux systems. In this thesis, we will use the name Schütte proof for this specific style of completeness proof.

The fact that the Schütte style completeness proof shows the completeness of a calculus is the main point of the proof. However, the completeness itself is not central in this thesis, but the *method* used in the proof to construct the countermodels. The Schütte method to create countermodels will be used for comparison of three different sequent systems for the modal logics K, T, $\mathsf{K4}$ and $\mathsf{S4}$: the basic (modal) sequent calculus system, a tree-hypersequent system, and a labeled sequent system.

What exactly the requirements are for a completeness proof to be a Schütte proof, or when a countermodel is a Schütte countermodel is not yet clearly defined, or widely agreed on in literature. Therefore, this thesis also aims to set the first steps in this direction. This is done by looking closer into the specific characteristic of these proofs, and define what steps in a completeness proof are required for a proof to be called a Schütte proof.

1.2 Gentzen calculi for modal logic

Gentzen systems for modal logics have historically been plagued with problems. It is not easy to create a satisfactory sequent calculus for even some basic modal logics like S5 [20]. However, in recent years several generalizations of the basic Gentzen calculus have been proposed which do succeed in describing most of the standard modal logics. These developments include the formulation of labeled approaches, hypersequent approaches and display calculi for modal logics.

Two important directions into which the Gentzen systems have developed are tree-hypersequents [23] and labeled sequents [18]. Both approaches incorporate the relational structure of the semantics in the calculus. Labeled sequents refer to the semantics directly, while tree-hypersequents use an extra layer of syntax to incorporate the semantics implicitly. Labeled systems use world-indicating labels and relational atoms in their sequents. While tree-hypersequents use multiple, related sequents to encode the relational structure of the semantics.

With the development of numerous new Gentzen systems, the question of what makes a good sequent calculus becomes more relevant. One would like to be able to rate and compare the newly described calculi in their aptness as an analytic calculi for describing a logic. With this, questions arise like: What are good properties to have as a sequent calculus? It is important to know how the different Gentzen systems compare with each other for different reasons. Firstly, it can show the superiority of one of the calculi. Secondly, it can show what system is preferable to use in a specific context. And thirdly, it can answer questions on the similarity of two systems, or give insight in where two calculi differ.

There are already some ways in which one can compare different sequent calculi with each other. For example, Avron [2] lists some properties a general proof-theoretical framework, like the Gentzen calculus, should aim for. Among these six properties are for example the sub-formula property, locality of the rules and the diversity of logics that can be described with the framework. A more in-depth treatment of different properties of sequent calculi can be found in Poggiolesi [22]. And with that, developing new properties for comparison of different systems is an important topic of research.

1.3 Research question

In this thesis, the aim is to use the Schütte countermodel construction as a new measure to compare sequent systems. Different sequent systems have slightly different Schütte completeness proofs, and with it, different countermodel construction methods. Is it possible to look at these methods, compare them and also compare the sets of possible Schütte countermodels these sequent systems can produce.

In the thesis, the Schütte countermodels are explained and properly defined. Three sequent calculi systems are compared based on their Schütte countermodel production. These calculi are the tree-hypersequent system of Poggiolesi [22] ($\mathbf{Ths}_*$), the labeled sequent calculus of Negri [18] ($\mathbf{Tls}_*$) and the basic sequent systems for modal logic ($\mathbf{G3m}_*$) which are extended forms of **G3** [27]. The question is answered whether the possible Schütte style countermodels produced using these sequent systems are exactly the same set of countermodels or whether it depends on the sequent system what models can be constructed. In the thesis it is proven these three sequent systems can exactly construct the same Schütte countermodels for basic sequents of the logics K, T, K4 and S4. This is done by proving the following theorem:

Theorem 1.1. *$\mathfrak{M}$ is a $\mathbf{G3m}_*$ Schütte model for the sequent $\Gamma \Rightarrow \Delta$, if and only if $\mathfrak{M}$ is a $\mathbf{Ths}_*$ Schütte model for the equivalent tree hypersequent $\Gamma \Rightarrow \Delta$ if and only if $\mathfrak{M}$ is a $\mathbf{Tls}_*$ Schütte model for the equivalent labeled sequent $x : \Gamma \Rightarrow x : \Delta$, where $*$ is either K, T, K4 or S4.*

The research done in the thesis is important on three levels. First, it shows it is possible to use meta-proofs of sequent systems, like the Schütte style completeness proof, and use them to closely compare different systems. This is possible because these meta-proofs about different systems can be similar, but differ on small but essential parts, which lay bare differences between systems. Secondly, this research creates an overarching way to look at different Schütte completeness proofs and identify different steps in the proof, which are common over different instances of this kind of proof. With that, this research is able to give more insight in exactly what a Schütte proof encompasses. And finally, this thesis shows the similarity between three different sequent systems. It shows that even though the labeled and tree-hypersequent are much more syntax-heavy still produce the same set of Schütte countermodels as the basic modal sequent systems, and thus are in this aspect not better. One might say they are even worse in this aspect because they achieve the same with much more syntax.

The thesis is structured as follows. In chapter 2, the basics of propositional modal logic are explained including Kripke semantics. In chapter 3 the necessary information about the three different sequent systems is presented, this includes a general introduction of the systems as well as some important lemmas like the admissibility of structural rules. Chapter 4 includes a discussion on what exactly the Schütte method is, and when a model can be called a Schütte countermodel. In chapter 5 we look at the calculus specific parts which we encounter in their respective Schütte proofs. And finally, in chapter 6 the theorem is proven that all the three systems exactly are able to create the same Schütte models. Chapter 7 concludes this research but in chapter 8 we present some extra exploration of the Schütte method in non-classical modal logic. Here we prove that the modal Dyckhoff calculus is complete with respect to intuitionistic Kripke models using a kind of Schütte method related to the one used in [26].

Chapter 2

Modal Logic

2.1 Modal propositional logic

Modal logics differ from classical propositional logic by the addition of an unary operator: the modality, which is written as a box ($\Box$). This modality can be interpreted as many things based on what kind of logic is used, but most basic interpretation is that of necessity. So $\Box\phi$ is intuitively interpreted as 'it is necessary that ϕ'. Most of the time this box operator is accompanied by a diamond operator $\Diamond$ which is the dual of the box and $\Diamond\phi$ means $\neg\Box\neg\phi$ and then is interpreted as 'it is possible that ϕ'. However many different logics can be constructed with these operators, together with those of classical propositional logic, which might lead to different intuitive interpretations of the operators like 'it is admissible' or 'it is known that'. The total language of formulas can by described by:

$$[\mathfrak{L}_{modal}] \quad \phi ::= p \mid \ \bot \ \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \phi \to \phi \mid \Box\phi \mid \Diamond\phi\,,$$

where p is an atomic proposition of the set of used propositional variables A ($p \in A$) and $\bot$ is falsum. Because some connectives can be rewritten in other connectives we will not always use the full language for brevity sake.

2.2 Kripke semantics

The accompanying semantics that is used in this thesis is the relational semantics of Kripke. In this semantics we have a set of worlds, in which different propositions can be true. These worlds however have a relation between them. For some worlds we can move from that world to other worlds: they are accessible by that world.

The interpretation of the necessity and possibility operators then are linked to this relation between the worlds. If a proposition is necessary in a world, then it is true in every world that is accessible from that world. If a proposition is possible in a world, then there is a world which is accessible from that world where that proposition is true.

When we formalize this, it is nice to first define Kripke frames. Then we will see that different kinds of modal logics are complete to Kripke models with specific frame conditions.

Definition 2.1. A Kripke Frame is a tuple $\mathfrak{F} =< W, R >$ such that:

1. W is a non empty set (of possible worlds).
2. $R \subseteq (W \times W)$ is a binary relation on W. If wRw' then we say world w' is accessible from w.

With the use of this definition we can then define what a full Kripke model is.

Definition 2.2. A Kripke Model is a tuple $\mathfrak{M} =< W, R, V >$ such that:

1. $< W, R >$ is a Kripke frame.
2. V is a function assigning a truth value to each atomic formula p for each world $w \in W$. $V(w, p) \in \{0, 1\}$.

With this and the following definitions for the truth values of the connectives we can define the truth of a formula in a model.

Definition 2.3. let $\mathfrak{M} =< W, R, V >$ be a Kripke model with $w \in W$ and A the set of propositional variables with $p \in A$. Then the truth of a formula is inductively defined relative to model $\mathfrak{M}$ and world w in the following way.

1. $\mathfrak{M}, w \vDash p \iff V(w, p) = 1$.
2. $\mathfrak{M}, w \vDash \perp \iff$ is not the case
3. $\mathfrak{M}, w \vDash \psi \wedge \phi \iff \mathfrak{M}, w \vDash \psi$ and $\mathfrak{M}, w \vDash \phi$.
4. $\mathfrak{M}, w \vDash \psi \vee \phi \iff \mathfrak{M}, w \vDash \psi$ or $\mathfrak{M}, w \vDash \phi$.
5. $\mathfrak{M}, w \vDash \psi \rightarrow \phi \iff \mathfrak{M}, w \vDash \psi$ implies $\mathfrak{M}, w \vDash \phi$.
6. $\mathfrak{M}, w \vDash \Box\psi \iff$ For all $w' \in W$, if wRw', then $\mathfrak{M}, w' \vDash \psi$.
7. $\mathfrak{M}, w \vDash \Diamond\psi \iff$ There is a w' for which wRw' and $\mathfrak{M}, w' \vDash \psi$.

Validity in Kripke semantics is defined as follows:

Definition 2.4. Let ψ be a modal formula, C a class of Kripke frames and $\mathfrak{M}_c$ a model with a frame of class C then:

$$\vDash_C \psi \iff \mathfrak{M}_c \vDash \psi \text{ for all models } \mathfrak{M}_c \text{ based on frames of class C.}$$

Different normal modal logic's then are defined semantically by their accessibility relation. The most basic one being K which has no restrictions on the kind of accessibility relation. The logic K can be axiomatized by by the axioms and rules in table 2.1. We can then define the soundness and completeness of K as:

Definition 2.5. A formula is a theorem of K if and only if it is true in all Kripke models:

$$\vDash \psi \iff \vdash_{\mathsf{K}} \psi\,.$$

Axioms:	
1	Any axiomitization of classical propositional logic
2	$\Box(A \to B) \to (\Box A \to \Box B)$
3	$\Diamond A \leftrightarrow \neg\Box\neg A$
Rules:	
1	Modus Ponens: from $A \to B$ and A, deduce B
2	Necesiation rule: from A, deduce $\Box A$

Table 2.1: Axiomitization of K

By extending K with different axioms we can create different modal logics which relate to specific classes of Kripke frames. The classes of these frames are based on specific geometric relations that the accessibility relation satisfies. In table 2.2 we can see what axioms relate to which class frames.

Name	Axiom	Frame Property
D	$\Box A \to \Diamond A$	Seriality: $\forall w\ \exists w'\ wRw'$
T	$\Box A \to A$	Reflexivity: $\forall w\ wRw$
B	$A \to \Box\Diamond A$	Symmetry: $\forall w\ \forall w'\ wRw' \to w'Rw$
4	$\Box A \to \Box\Box A$	Transitivity: $\forall w\ \forall w'\ \forall w''\ wRw' \wedge w'Rw'' \to wRw''$

Table 2.2: Axioms related to frame properties

With these properties and axioms we can create different logics which corresponds to different frame classes. The naming is as follows. If we extend K with the axiom T, we get the logic KT, if we extend it then with axiom D, we get KDT etc. Only the logics KT4, KTB4 and KT have for historic reasons a different name: S4, S5 and T respectively.

There are also other axioms and with them come other modal logics that can be constructed. These do not necessarily have to coincide with specific frame properties of the relational semantics. In this study we will not go into these and we will limit ourselves only to the logics K, T, K4 and S4.

Chapter 3

Three Different Sequent Calculi

3.1 Basic modal sequent calculus

Sequent calculi for classical and intuitionistic propositional logic are originally formulated by Gentzen [11]. For these logics, sequent calculus systems are recognized as one of the most elegant and basic analytic calculi. In this thesis we will use a multiset variant of the sequent calculus.

Definition 3.1. A sequent S is an expression of the form $\Gamma \Rightarrow \Delta$ where Γ and Δ are finite multisets of formulas. Γ is the antecedent, Δ is the succedent. We will also refer to the antecedent (succedent) of a sequent S with $S^a(S^s)$.

Definition 3.2. The interpretation of a sequent $\Gamma \Rightarrow \Delta$ is

$$I(\Gamma \Rightarrow \Delta) \equiv \bigwedge \Gamma \rightarrow \bigvee \Delta \,.$$

When Γ or Δ is empty the interpretation is $\bigwedge \emptyset \equiv \top$ and $\bigvee \emptyset \equiv \bot$.

Some examples of interpretations of sequents are the following:

$$\begin{aligned} I(p \rightarrow q, p \Rightarrow q) &= ((p \rightarrow q) \wedge p) \rightarrow q \,, \\ I(\Rightarrow \Box p) &= \Box p \,, \\ I(\Box p \Rightarrow \Box q, p) &= \Box p \rightarrow (\Box q \vee p) \,. \end{aligned}$$

A prominent cut free version of a sequent calculus for classical propositional logic is **G3** which can be found in Troelstra and Schwichtenberg [27]. Because the modal sequent calculi that will be presented are all extensions of, **G3** will be presented first.

Definition 3.3. Sequent calculus **G3**:
Axioms:

$$\frac{}{p, \Gamma \Rightarrow \Delta, p}\ \mathsf{Ax} \qquad \frac{}{\bot, \Gamma \Rightarrow \Delta}\ \mathsf{Ax}$$

Propositional Rules:

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

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

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

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

For all rules we use the customary definitions of what the principal and auxiliary formulas are. The principal formula is the formula in the conclusion which is created form the auxiliary formula(s) of the premise(s) with the connective which is introduced in that specific rule. Γ and Δ are the context.

Definition 3.4. A proof of a sequent S is a finite tree with at its nodes sequents which are connected according to the rules of the sequent system. The root of the tree is the sequent S and the leafs are sequents which have the form of axioms. We call such a tree a prooftree.

If the sequent S is derivable by the prooftree $\mathcal{D}$, and $\mathcal{D}$ has a depth of at most n, we write $\mathcal{D} \vdash_n S$.

Definition 3.5. A rule is depth preserving invertible if the following is true. When the conclusion of a rule has a proof, then each premise of that rule has also a proof of at most the same depth.

Notice that the propositional rules of all the sequent calculi **G3** are depth preserving invertible. This can easily be shown with induction on the depth of the derivation. For reference, this can be found in Troelstra and Schwichtenberg [27] as the proof of proposition 3.5.4.

Besides this contraction rules are depth preserving admissible in **G3**. This will not be proven, however we will prove this for the modal sequent calculi which extend **G3**.

3.1.1 Sequent calculi for K, T, K4 and S4

The sequent calculus **G3** can be extended to $\mathbf{G3m}_K$ for the modal logic K by adding an additional rule. $\mathbf{G3m}_K$ then can be extended to the logic $\mathbf{G3m}_T$ by adding another rule.

Definition 3.6. The sequent calculus $\mathbf{G3m}_K$ can be constructed by extending **G3** by the following rule:

$$\frac{\Gamma \Rightarrow A}{\Sigma, \Box\Gamma \Rightarrow \Phi, \Box A}\ K\Box$$

The sequent calculus $\mathbf{G3m}_T$ can be constructed by extending $\mathbf{G3m}_K$ by the following rule:

$$\frac{\Gamma, \Box A, A \Rightarrow \Delta}{\Gamma, \Box A \Rightarrow \Delta}\ T\Box$$

Remark 3.7. In the rule $K\Box$, $\Box\Gamma$ can only contain formulas of the form $\Box B$ and Σ does not contain any formulas with the box operator as its main connective[1].

The sequent calculi for the logics K4 and S4 can in a similar way be derived from **G3**. The only difference is that, to create transitivity in the system, the $K\Box$ rule need to be changed into the $4\Box$ rule.

Definition 3.8. The sequent calculus $\mathbf{G3m}_{K4}$ can be constructed by extending **G3** by the following rule:

$$\frac{\Box\Gamma, \Gamma \Rightarrow A}{\Sigma, \Box\Gamma \Rightarrow \Phi, \Box A}\ 4\Box$$

The sequent calculus $\mathbf{G3m}_{S4}$ can be constructed by extending $\mathbf{G3m}_{K4}$ by the following rule:

$$\frac{\Gamma, \Box A, A \Rightarrow \Delta}{\Gamma, \Box A \Rightarrow \Delta}\ T\Box$$

Remark 3.9. In the rule $4\Box$, $\Box\Gamma$ can only contain formulas of the form $\Box B$ and Σ does not contain any formulas with the box operator as its main connective.

For the rules $K\Box$ and $4\Box$ the formulas in $\Box\Gamma$, are also seen as auxiliary formulas.

Remark 3.10. To address all four of the different modal sequent calculi based on **G3** we will write $\mathbf{G3m}_*$.

[1] Aometimes an additional condition on Σ is set saying it can only contain atoms, for example in [4]. This assures saturation of the propositional rules between applications of this rule. We do however not add this condition because this kind of saturation is defined somewhere else more specifically. See definition 5.1.

Although the $K\Box$ and the $4\Box$ rules are not invertible some nice properties of **G3** are still present in the modal variants. Most importantly for us later, the contraction rules are still depth preserving admissible.

Proposition 3.11. *For all the logics* $\mathbf{G3m}_*$*, contraction is depth preserving admissible:*

$$\textit{If } \mathbf{G3m}_* \vdash_n A, A, \Gamma \Rightarrow \Delta, \textit{ then } \mathbf{G3m}_* \vdash_n A, \Gamma \Rightarrow \Delta\,.$$
$$\textit{If } \mathbf{G3m}_* \vdash_n \Gamma \Rightarrow A, A, \Delta, \textit{ then } \mathbf{G3m}_* \vdash_n \Gamma \Rightarrow A, \Delta\,.$$

Proof. Proof based on induction on the depth n of the derivation $\mathcal{D}$. We will only proof the first rule, the second is analogous.

If $\mathcal{D}$ is of depth 1:
$A, A, \Gamma \Rightarrow \Delta$ is an axiom. Therefore, $A, \Gamma \Rightarrow \Delta$ is also an axiom and thus derivable.

Induction hypothesis:
If $n > 1$, the induction hypotheses tells us that the admissibility of contraction holds for $n - 1$, and we only have to verify for the last applied rule.

In case of a propositional rule:
If the last rule was a rule in which A is not the principal formula, then we can apply the induction hypothesis directly to show that $A, \Gamma \Rightarrow \Delta$ is also derivable.

If, in the last rule was a propositional rule, like $L\wedge$ with A as its principal formula, then the last rule looks like this:

$$\frac{B \wedge C, B, C, \Gamma \Rightarrow \Delta}{B \wedge C, B \wedge C, \Gamma \Rightarrow \Delta}\ L\wedge$$

Because of the depth preserving invertibility of the propositional rules, $\mathbf{G3m}_* \vdash_{n-1} B, C, B, C, \Gamma \Rightarrow \Delta$. We can then apply the induction hypothesis for both B and C.

The argument for the rules $L\vee$, $L\neg$ and $L \rightarrow$ is analogous to this and we will not be treated here.

If the last rule that was used is $K\Box$:
If $A \in \Sigma$, the application of $K\Box$ looks like this:

$$\frac{\Gamma \Rightarrow B}{\Sigma', A, A, \Box\Gamma \Rightarrow \Phi, \Box B}\ K\Box$$

In which case $\Sigma', A, \Box\Gamma \Rightarrow \Phi, \Box B$ also derivable.

If A is $\Box B \in \Box\Gamma$, the application of $K\Box$ looks like this:

$$\frac{\Gamma', B, B \Rightarrow A}{\Sigma, \Box\Gamma', \Box B, \Box B \Rightarrow \Phi, \Box A}\ K\Box$$

We then can apply the induction hypothesis on B.

If the last rule that was used was $T\Box$:
If $A \in \Gamma$ then we can apply the induction hypothesis directly, even if A is of the form B. If A is $\Box B$ and the principal formula, the last rule looks like this:

$$\frac{\Box B, \Box B, B, \Gamma \Rightarrow \Delta}{\Box B, \Box B, \Gamma \Rightarrow \Delta}\ T\Box$$

Because of the induction hypothesis $\Box B, B, \Gamma \Rightarrow \Delta$ is also derivable with a $\mathcal{D}$ with depth $n-1$. And thus applying the rule $\Box T$ to this sequent gives us a derivation of $\Box B, \Gamma \Rightarrow \Delta$ of depth n.

If the last rule that was used was $4\Box$:
This case is analogous to $K\Box$. If $A \in \Sigma$, then we can use the induction hypothesis directly. If A is $\Box B$ and in $\Box\Gamma$, the premise contains both B and $\Box B$ twice, and we can use the induction hypothesis twice. ■

3.1.2 Problems with the basic modal Gentzen systems

These modal sequent calculi are not as elegant as their propositional original. This is because the new modal rules do not posses some of the nice properties the propositional rules of **G3** do posses such as symmetry and invertibility.

For example the sequent calculus $\mathbf{G3m}_K$ is not symmetric. This is because it does not have a right and left introduction rule for the modal operator, but only one rule: $K\Box$. Besides this, the rule $K\Box$ and $4\Box$ also has implicit weakening in it. This makes the $K\Box$ and $4\Box$ rule not invertible and proof search a trickier business in the modal variants of **G3** than in normal version.

Rule $T\Box$ and $4\Box$ also include implicit contractions. This challenges the termination of the proof searches for the sequent calculi which include these rules. However, the calculi are terminating, but for this to happen additional strategies need to be used during the proof search such as loop checking in the case of the transitivity rule $4\Box$, or in the case of reflexivity, keeping track of what modal formula have been already used as prime formulas in the $T\Box$ rule.

Because of these not so nice properties of the modal rules for the basic sequent system, and the fact it has not yet been possible to capture the symmetry axiom for logics like B and S5 in the basic sequent calculus, many different extensions of the sequent framework have been proposed. Two of them are the ones that will be discussed in the next sections.

3.2 Labeled sequent calculus

One extension of the basic sequent calculus is the labeled approach by Negri [18]. We will present here a variant of this kind of system. This system does not use sequents and formulas as their basic components, but *labeled* sequents by having prefixed formulas, and relational atoms in the sequents instead of normal formulas. By introducing labels in the sequents the possible world semantic of Kripke can be referenced more explicit in the sequents themselves: different labels represent different worlds. This solves some of the problems that were present for the $\mathbf{G3m}_*$ systems.

Definition 3.12. A prefixed formula is a formula A with a prefixed x and is written in the following way: $x : A$. Two prefixes x and y can be used to construct relational atom xRy. A labeled sequent is a sequent $\Gamma \Rightarrow \Delta$, in which Γ is a multiset that can contain relational atoms and prefixed formula and Δ is a multiset that can contain only prefixed formulas. If a multiset Γ, contains only formulas prefixed with the label x we will also write it as $x : \Gamma$.

In this version of the labeled sequent calculus we only do not consider relational atoms existing in the succedent of sequents. This is contrary to the original calculus of Nergi, where it is possible to have relational atoms in the succedent. However, Negri comments that "no rule removes an atom of the form xRy from the right-hand side of sequents, and such atoms are never active in the logical rules"[18, p.531] and that axioms that require relation atoms in the succedent of a sequent "can as well be left out from the calculus without impairing completeness of the system."[18, p.531]. Because of this relational atoms have been left out of the succedents of sequents all-together.

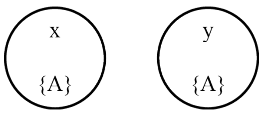


FIGURE 3.1: A model that satisfies the labeled sequent $x : A \Rightarrow y : A$.

3.2.1 Tree-labeled sequents

A problem with the labeled approach of Negri is that it lacks an interpretation function of the sequents to modal formulas. The reason for this is that the language of labeled sequents is richer than that of normal sequents. Because of the prefixes, the sequents say something about labeled formulas instead of formulas proper. It is for example possible to construct the labeled sequent $x : A \Rightarrow y : A$. However, it is not possible for this sequent to be written down in one modal formula. This is because there is no relation between the worlds x and y. The sequent $x : A \Rightarrow y : A$ is for example satisfied in the model in figure 3.1. But it is not possible to say something in one formula about these two worlds because they are not related. The labeled sequent is able to express something about the model a formula is not able to express which shows why it it impossible to have a proper interpretation of labeled sequents to modal formulas. A way to solve this is to only consider *tree* labeled sequents. This is for example also done in [16] and [12].

Definition 3.13. Let LAB be all the labels that occur in a labeled sequent S and $\boldsymbol{R}$ be the union of all the relational atoms in S^a.

A tree-labeled sequent is a labeled sequent such that the the relation $\boldsymbol{R}$ over the labels LAB, is an outward directed rooted tree. We call the label x which is at the root of this tree the root label.

Example 3.1. *The following sequents are examples of sequents which are tree-labeled:*

$$x : A \Rightarrow x : B \wedge A\,,$$
$$xRy, yRz, x : A, y : A, \Rightarrow z : B \wedge A\,,$$
$$xRy, xRz, x : A, y : A \Rightarrow z : B \wedge A\,.$$

The following sequents are examples of sequents which are not tree-labeled:

$$xRx, x : A \Rightarrow x : B \wedge A\,,$$
$$xRy, yRz, zRx, x : A, y : A, \Rightarrow z : B \wedge A\,.$$

When we only consider tree-labeled sequents it is possible to interpret the labeled sequents as formulas. It is now possible to do so, because we can interpret the relations between the worlds in the sequent from the point of view of the root label as modal relations.

Definition 3.14. The interpretation of a labeled sequent $\Gamma \Rightarrow \Delta$ can be constructed as follows. For each label we have two different interpretation functions:

$$I^x(\Gamma, xRy_1, xRy_2, .., xRy_n \Rightarrow \Delta) \equiv I_x(\Gamma \Rightarrow \Delta) \vee \Box I^{y_1}(\Gamma \Rightarrow \Delta) \vee \Box I^{y_2}(\Gamma \Rightarrow \Delta) \vee .. \vee \Box I^{y_n}(\Gamma \Rightarrow \Delta)\,.$$
$$I_x(\Gamma, x : \Sigma \Rightarrow \Delta, x : \Pi) \equiv \bigwedge \Sigma \rightarrow \bigvee \Pi\,.$$

The interpretation $I^x(\Gamma \Rightarrow \Delta)$ where x is the root label is then the interpretation of the whole labeled sequent.

When Σ or Π is empty the interpretation is $\bigwedge \emptyset \equiv \top$ or $\bigvee \emptyset \equiv \bot$.

Intuitively this interpretation says that we should see the total labeled sequent as multiple smaller sequents, one for each label. For these smaller sequents the normal interpretations of a sequent applies. These individual interpretations are then 'glued together' in one big disjunction and adding a modal operator between two sequents for each relational atom.[2]

3.2.2 Tree-labeled sequent calculi for K, T, K4 and S4

Definition 3.15. Sequent calculus $\mathbf{Tls}_K$:
Axioms:

$$\frac{}{x : p, \Gamma \Rightarrow \Delta, x : p}\ \mathsf{Ax}$$

Propositional Rules:

$$\frac{\Gamma, x : A, x : B \Rightarrow \Delta}{\Gamma, x : A \wedge B \Rightarrow \Delta}\ L\wedge \qquad \frac{\Gamma \Rightarrow \Delta, x : A \quad \Gamma \Rightarrow \Delta, x : B}{\Gamma \Rightarrow \Delta, x : A \wedge B}\ R\wedge$$

$$\frac{\Gamma, x : A \Rightarrow \Delta \quad \Gamma, x : B \Rightarrow \Delta}{\Gamma, x : A \vee B \Rightarrow \Delta}\ L\vee \qquad \frac{\Gamma \Rightarrow \Delta, x : A, x : B}{\Gamma \Rightarrow \Delta, x : A \vee B}\ R\vee$$

$$\frac{\Gamma \Rightarrow \Delta, x : A \quad \Gamma, x : B \Rightarrow \Delta}{\Gamma, x : A \to B \Rightarrow \Delta}\ L\to \qquad \frac{\Gamma, x : A \Rightarrow \Delta, x : B}{\Gamma \Rightarrow \Delta, x : A \to B}\ R\to$$

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

Modal Rules:

$$\frac{\Gamma, x : \Box A, xRy, y : A \Rightarrow \Delta}{\Gamma, x : \Box A, xRy \Rightarrow \Delta}\ L\Box \qquad \frac{\Gamma, xRy \Rightarrow \Delta, y : A}{\Gamma \Rightarrow \Delta, x : \Box A}\ R\Box*$$

*in $R\Box$, y has to be a fresh label (not occurring in Γ or Δ).

We can extend this sequent system for K with specific rules based on the frame properties corresponding to T, K4 and S4 in Kripke semantics to get labeled sequent calculi for these logics.

Definition 3.16. Sequent calculi $\mathbf{Tls}_T$, $\mathbf{Tls}_{K4}$ and $\mathbf{Tls}_{S4}$ can be created by extending $\mathbf{Tls}_K$ with the following rules based on the frame properties associated with these logics in Kripke semantics:

$$\frac{\Gamma, x : \Box A, x : A \Rightarrow \Delta}{\Gamma, x : \Box A \Rightarrow \Delta}\ \mathit{ref} \qquad \frac{\Gamma, x : \Box A, xRy, y : \Box A \Rightarrow \Delta}{\Gamma, x : \Box A, xRy \Rightarrow \Delta}\ \mathit{trans}$$

Definition 3.17. For the rules $L\wedge$, $R\wedge$, $L\vee$, $R\vee$, $L\to$, $R\to$, $L\neg$, $R\neg$ and *ref* we call the prefix x the *active prefix* of the rule.

[2] The smaller implications are connected with disjunctions in the interpretation because we still want, just as in a normal sequent, that we can derive a sequent if and only if at least one of the succedent formulas can be derived when all the antecedent formulas are true.

For the rules $L\Box$, $R\Box$ and *trans* we call the prefixes x and y the *active prefixes* and xRy the *active relational atom* of the rule, x is called the *first* prefix and y the *second* prefix.

The transitivity and reflexivity rules used here are not the original rules used by Negri in [18]. The reason to use other transitivity and reflexivity rules is that these rules are a bit easier to work with when comparing this calculus to **Ths** and **G3m**. Why this is, and what changes when the original rules of Nergi are used is explained in 6.5.

Even though these calculi are not especially constructed for tree-labeled sequents, they do preserve the tree labeling property when these rules are used in an bottom-up fashion. This means that if we only want to limit our view to tree-labeled sequents, we can safely use these sequent systems.

Proposition 3.18. *If* $\Gamma \Rightarrow \Delta$ *is an rooted tree-labeled sequent and a conclusion of one of the rules of* **Tls***, then the premises are also rooted tree-labeled sequents.*

Proof. Check for each rule whether this is true.
For the Propositional rules, *ref*, *trans* and $L\Box$ this is trivial because they do not add relational atoms.

The rule $R\Box$ adds a relational atom xRy such that $x \in LAB_{\textsf{conclusion}}$ and $y \notin LAB_{\textsf{conclusion}}$. Therefore the added relational atom xRy adds a branch from x to the new label y, which preserves the tree structure of $\boldsymbol{R}_{conclusion}$ to $\boldsymbol{R}_{premise}$. ■

These labeled calculi have some nice properties which the Gentzen sequent systems **G3m** do not have such as symmetry of the modal rules and invertibility of all the rules. However, the $L\Box$ rule includes implicit contraction which complicates simple termination of the calculus even for $\mathbf{Tls}_K$. Besides this, weakening and contraction are both height preserving admissible. This is proven in propositions 4.4 and 4.12 in [18].

3.3 Tree-hypersequent Calculus

In this section the tree-hypersequent system of Poggiolesi [23] is presented. This system is very similar to Brünnler's deep sequent system [5]. The system is an expansion of Avron's hypersequent system [2]. In the hypersequent system not only single sequents are used as nodes in derivations, but multiple sequents are used together in one derivation step. The tree-hypersequent system is an extension of hypersequents where a tree ordering between the sequents is created. The syntax places these sequents in a tree structure. This tree structure will be encoding the accessibility relation between different world nodes in the semantics when creating the Schütte countermodels later on.

Via this tree structure the connection with the tree-labeled calculi can be seen. Because these calculi work with tree-hypersequents some additional notation is needed to encode the tree structure between sequents. Besides that we establish some conventions which make describing the rules of the system more concise.

- ";" and "/" are two meta-linguistic symbols that connect two sequents in the hypersequent. "/" indicates a parent-child relation between the two sequents and ";" indicates an equal relation.
- $\Gamma \Rightarrow \Delta$, ... denote (multiset) sequents.
- G, H, ... denote tree-hypersequents.
- $\underline{X}, \underline{Y}$, ... denote multisets of tree-hypersequents.

First let us see what tree-hypersequents are.

Definition 3.19. A tree-hypersequent THS is inductively defined in the following way:

- if S is a sequent, then S is a tree-hypersequent ,
- if S is a sequent and $G_1, ...G_n$ are tree-hypersequents, then $S/G_1; ....; G_n$ is a tree-hypersequent .

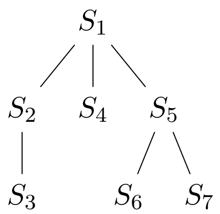
The interpretation of tree hypersequents is similar to that of tree-labeled sequents.

Definition 3.20. The interpretation of a tree-hypersequent is inductively defined in the following way

- $I(\Gamma \Rightarrow \Delta) \equiv_{def} \bigwedge_{i\in\Gamma} A_i \rightarrow \bigvee_{i\in\Delta} B_i$:
- $I(\Gamma \Rightarrow \Delta/G_1; ....; G_n) \equiv_{def} I(\Gamma) \vee \Box I(G_1) \vee .... \vee \Box I(G_n)$.

Example 3.2. *The tree structure of the following tree-hypersequent is visually shown:*

$$S_1/((S_2/S_3); S_4; (S_5/(S_6; S_7))) \,.$$



The rules of the $\mathbf{Ths}_*$ calculi only focus on one or two sequents that are in the tree-hypersequent. To do this without describing the whole context of the tree-hypersequent we use the notation $G[\Gamma \Rightarrow \Delta]$ or $G[H]$. Informally this represents a focus function. $G[\,]$ should be seen as a tree-hypersequent G with one hole in it. It becomes a real tree-hypersequent if this hole gets filled in with a sequent S, or a tree hypersequent H. If a rule changes something in the sequents of the tree-hypersequents on which in focused, the other sequents in the tree-hypersequent remain unchanged. If we for example have the tree hypersequent G: $S_1/((S_2/S_3); S_4; (S_5/(S_6; S_7)))$, and we want to focus on sequent Γ_4 we can do this by describing G as $G[S_4]$. Which means that we look at sequent S_4, which is somewhere in the tree-hypersequent G. More information and a formal definition of this notation can be found in section 6.1 of [23].

3.3.1 Tree-hypersequent calculi for K, T, K4 and S4

Definition 3.21. Sequent calculus $\mathbf{Ths}_K$:
Axioms:

$$\frac{}{G[p, \Gamma \Rightarrow \Delta, p]}\ \mathsf{Ax}$$

Propositional Rules:

$$\frac{G[\Gamma \Rightarrow \Delta, A]}{G[\neg A, \Gamma \Rightarrow \Delta]}\ L\neg \qquad \frac{G[A, \Gamma \Rightarrow \Delta]}{G[\Gamma \Rightarrow \Delta, \neg A]}\ R\neg$$

$$\frac{G[A, B, \Gamma \Rightarrow \Delta]}{G[A \wedge B, \Gamma \Rightarrow \Delta]}\ L\wedge \qquad \frac{G[\Gamma \Rightarrow \Delta, A] \quad G[\Gamma \Rightarrow \Delta, B]}{G[\Gamma \Rightarrow \Delta, A \wedge B]}\ R\wedge$$

$$\frac{G[A, \Gamma \Rightarrow \Delta] \quad G[A, \Gamma \Rightarrow \Delta]}{G[A \vee B, \Gamma \Rightarrow \Delta]}\ L\vee \qquad \frac{G[\Gamma \Rightarrow \Delta, A, B]}{G[\Gamma \Rightarrow \Delta, A \vee B]}\ R\vee$$

$$\frac{G[\Gamma \Rightarrow \Delta, A] \quad G[B, \Gamma \Rightarrow \Delta]}{G[A \to B, \Gamma \Rightarrow \Delta]}\ L\to \qquad \frac{G[\Gamma, A \Rightarrow \Delta, B]}{G[\Gamma \Rightarrow \Delta, A \to B]}\ R\to$$

Modal Rules:

$$\frac{G[\Box A, \Gamma \Rightarrow \Delta/(A, \Sigma \Rightarrow \Phi/\underline{X})]}{G[\Box A, \Gamma \Rightarrow \Delta/(\Sigma \Rightarrow \Phi/\underline{X})]}\ L\Box \qquad \frac{G[\Gamma \Rightarrow \Delta/ \Rightarrow A]}{G[\Gamma \Rightarrow \Delta, \Box A]}\ R\Box$$

Definition 3.22. Sequent calculi $\mathbf{Ths}_T$, $\mathbf{Ths}_{K4}$ and $\mathbf{Ths}_{S4}$ can be created by extending $\mathbf{Ths}_K$ with the following rules based on the frame properties associated with these logics in Kripke semantics:

$$\frac{G[\Box A, A, \Gamma \Rightarrow \Delta]}{G[\Box A, \Gamma \Rightarrow \Delta]}\ \mathit{ref} \qquad \frac{G[\Box A, \Gamma \Rightarrow \Delta/(\Box A, \Sigma \Rightarrow \Phi/\underline{X})]}{G[\Box A, \Gamma \Rightarrow \Delta/(\Sigma \Rightarrow \Phi/\underline{X})]}\ \mathit{trans}$$

Just as the Tree-labeled sequent calculi these calculi have some nice properties like invertibility of the rules and height preserving admisibility of contraction and weakening. But is also has the same kind of problems as the tree-labeled sequent calculi. For example the $L\Box$ rule also has implicit contraction in it, which makes termination of the calculus not straightforward.

3.4 Initial comparison of the three calculi

The $\mathbf{G3m}_*$ calculi are very concise in syntax and have a close resemblance to the original Gentzen systems for propositional logic. This makes them quite easy to use and read derivations in them. However, the downside is that the calculus does not preserve its 'history' when used in backwards proof search. When a modal rule like the $K\Box$ is used, the additional boxed formulas that are in the antecedent of the conclusion are 'weakened away' and do not occur in the premise of that rule. If we then also want to unfold these formulas, it is necessary to use backtracking.

This is not needed in the other two calculi. Here we preserve the formulas because the usage of the $R\Box$ rule creates a new related sequent in $\mathbf{Ths}_*$ or an extra related label in $\mathbf{Tls}_*$. This preservation of sequents or labels in the modal rules makes all of them invertible and makes backtracking unnecessary in these calculi. On the other hand, it makes them have a lot of formula's accumulating in the sequents when doing backward proof search. This makes nodes in derivations of these two calculi get big very fast, and hard write down concise.

Although there is no backtracking needed in $\mathbf{Ths}_*$ and $\mathbf{Tls}_*$, the drawback of this is that the $L\Box$ rule in both systems includes implicit contraction of the $\Box A$ formula in the antecedent. Because of this even $\mathbf{Ths}_K$ and $\mathbf{Tls}_K$ are not straightforwardly terminating, while this is the case for $\mathbf{G3m}_K$. However, this is easily fixed by limiting the application of the $L\Box$ rule. In general, $\mathbf{Ths}_*$ and $\mathbf{Tls}_*$ are very similar and one can translate rules and derivations of the one into the other, as is shown by Gore and Ramanayake [12].

More intricate differences between the these calculi will become clear when we start looking at how the Schütte counter model construction works for these three calculi in chapter 5. But before that we will first explain the Schütte method for proving completeness and defining what a Schütte countermodel is.

Chapter 4

Proof Procedures and the Schütte Method

To show that the three systems produce the same Schütte models, it is necessary to first define what a Schütte model is. Because Schütte models are models that are constructed in these specific Schütte style completeness proofs, we will first characterize these Schütte proofs, and list three essential parts in the proof that need to be present in a completeness proof if it wants to be categorized as a Schütte proof. These parts are a proof procedure, saturation and countermodel construction. As we will see some parts of this overall method are the same regardless which calculus is used. However, some parts of the proof are calculus depended. In this chapter we will look at the shared parts of the Schütte proof.

4.1 Example of a Schütte proof

As a start, we begin by looking at an example of a Schütte proof to get an idea of what they look like. The example we will treat is the proof of completeness of $\mathbf{Tls}_K$ for the semantics of Kripke models. The Schütte proof we will treat here is based on the proof of theorem 5.4 in [19], and a similar proof method can be found in chapter 8 (#4-#6) of [9] for prefixed tableaus. A difference is that these sources also include proofs for the Kripke semantics of the logics $\mathsf{K4}$, T and $\mathsf{S4}$. These will not be treated here for the sake of keeping it short.

The proof rests on the idea that the backward application of the rules in a proof procedure eventually unfolds every complex formula that is present in the to be proven sequent. Because all formulas get unfolded, atomic propositions accumulate in the branches of the derivation tree. When all leaves of the derivation eventually are axioms then we have a proof. However, if such a proof does not exist, the application of the rules to these branches of the derivation tree goes on forever. If this is the case for a branch in the derivation it is possible to use this branch to construct a Schütte countermodel. This means that, just as every Schütte proof, we will show completeness using modus tollens. We will proof that if a sequent is underivable, we can construct a countermodel by directly using the failed derivation tree.

Lemma 4.1 (Königs lemma)**.** *An infinite tree with only finitely branching vertexes has an infinite branch.*

Theorem 4.2. *Let $S = \Gamma \Rightarrow \Delta$ be a labeled sequent in* $\mathbf{Tls}_K$*. Either the sequent is derivable, or one can construct a Schütte countermodel which refutes the interpretation of the sequent.*

Proof. 1. Systematic proof procedure for sequent S

Stage 0: Put S at the root of the tree.

Stages n>0 have two cases:

Case 1: Every leaf of the derivation is an axiom. This makes our tree a proof of S and we stop the procedure.

Case 2: Not every leave of the derivation is an axiom. We continue the construction of the derivation tree by applying rules to the leafs of the derivation tree which are not axioms following the order of the stages.

There are 10 different stages, one for each rule of $\mathbf{Tls}_K$. At stage 10+1, return to stage 1.

Stage 1 $L\wedge$:
For each leaf sequent which is not an axiom apply rule $L\wedge$ backwards as long as possible. Meaning, if the sequent is of the form

$$x_1 : A_1 \wedge B_1, ..., x_m : A_m \wedge B_m, \Gamma' \Rightarrow \Delta$$

where there is no formula with conjunction as its main connective in Γ', we will write

$$x_1 : A_1, x_1 : B_1, ..., x_m : A_m, x_m : B_m, \Gamma' \Rightarrow \Delta$$

above it and thus apply rule $L\wedge$ backwards m times.

Stage 2 $R\wedge$:
For each leaf sequent which is not an axiom apply rule $R\wedge$ backwards as long as possible. Meaning, if the sequent is of the form

$$\Gamma \Rightarrow x_1 : A_1 \wedge B_1, ..., x_m : A_m \wedge B_m, \Delta'$$

where there is no formula with conjunction as its main connective in Δ'. We will write 2^m premise sequents

$$\Gamma \Rightarrow x_1 : C_1, ..., x_m : C_m, \Delta$$

above it, where C_i is either A_i or B_i. And thus we apply rule $R\wedge$ backwards m times.

Stage 3 $L\vee$**:**
For each leaf sequent which is not an axiom apply rule $L\vee$ backwards as long as possible. This is done in the same way as in stage 2.

Stage 4 $R\vee$**:**
For each leaf sequent which is not an axiom apply rule $R\vee$ backwards as long as possible. This is done in the same way as in stage 1.

Stage 5 $L \rightarrow$**:**
For each leaf sequent which is not an axiom apply rule $L \rightarrow$ backwards as long as possible. Meaning, if the sequent is of the form

$$x_1 : A_1 \rightarrow B_1, ..., x_m : A_m \rightarrow B_m, \Gamma' \Rightarrow \Delta$$

where there is no formula with implication as its main connective in Γ'. We will write 2^m premise sequents of the form

$$x_{i_1} : B_{i_1}, ..., x_{i_k} : B_{i_k}, \Gamma' \Rightarrow x_{j_{k+1}} : A_{j_{k+1}}, ..., x_{j_m} : A_{j_m}, \Delta$$

above it, where $i_1, ..., i_k \in \{1, ..., m\}$ and $j_{k+1}, ..., j_m \in \{i, ..., m\} - \{i_1, ..., i_k\}$. We thus apply rule $L\wedge$ backwards m times.

Stage 6 $R \rightarrow$**:**
For each leaf sequent which is not an axiom apply rule $R \rightarrow$ backwards as long as possible. Meaning, if the sequent is of the form

$$\Gamma \Rightarrow x_1 : A_1 \rightarrow B_1, ..., x_m : A_m \rightarrow B_m, \Delta'$$

where there is no formula with implication as its main connective in Δ'. We will write

$$x_1 : A_1, ..., x_m : A_m, \Gamma \Rightarrow x_1 : B_1, ..., x_m : B_m, \Delta'$$

above it and thus apply rule $R \rightarrow$ backwards m times.

Stage 7 $L\neg$**:**
For each leaf sequent which is not an axiom apply rule $L\neg$ backwards as long as possible. Meaning, if the sequent is of the form

$$x_1 : \neg A_1, ..., x_m : \neg A_m, \Gamma' \Rightarrow \Delta$$

and there is no formula with negation as its main operator in Γ'. We will write

$$\Gamma' \Rightarrow x_1 : A_1, ..., x_m : A_m, \Delta$$

above it and thus apply rule $L\neg$ backwards m times.

Stage 8 $R\neg$:
For each leaf sequent which is not an axiom apply rule $R\neg$ backwards as long as possible. This is done in the same way as in stage 7.

Stage 9 $L\Box$:
For each leaf sequent which is not an axiom apply rule $L\Box$ backwards as long as possible. Meaning, if the sequent is of the form

$$x_1 : \Box A_1, ..., x_m : \Box A_m, x_1 R y_1, .., x_m R y_m, \Gamma' \Rightarrow \Delta$$

where there is no pair of labeled formula with the necessity operator as its main operator and relational atom in Γ' of the form $x : \Box A$ and xRy[1]. We will write

$$y_1 : A_1, ..., y_m : A_m, x_1 : \Box A_1, ..., x_m : \Box A_m, x_1 R y_1, .., x_m R y_m, \Gamma' \Rightarrow \Delta$$

above it and thus apply rule $L\Box$ backwards m times.

Stage 10 $R\Box$:
For each leaf sequent which is not an axiom apply rule $R\Box$ as much as possible. Meaning, if the sequent is of the form

$$\Gamma \Rightarrow x_1 : \Box A_1, ..., x_m : \Box A_m, \Delta'$$

where there is no formula with the necessity operator as its main operator in Δ'. We will write

$$x_1 R y_1, .., x_m R y_m, \Gamma \Rightarrow y_1 : A_1, ..., y_m : A_m, \Delta'$$

above it, where $y_1, ..., y_m$ are fresh labels. We thus apply rule $R\Box$ backwards m times.

If for any stage n, the sequent is neither an axiom and the rule of that stage is not applicable we will write the sequent itself above it.

If the reduction tree is finite, this means that each leaf is an axiom, and it is thus a proof of the sequent at the root of the tree. If the reduction tree is infinite, there is an infinite branch in the tree. This is based on the fact that all of our rules are finitely branching, all of them have at most two premises. Because of this and Königs lemma we know that the tree must have a finite branch to be a finite tree. This means that one of the leafs does not result in an axiom and the tree is thus not a proof of the sequent at the root.

[1] Where the label of the boxed formula is the same as the first active prefix in the relational atom.

2 Construction of the countermodel
It is now possible to construct a countermodel to that sequent by using a branch which is infinite.

Let $S_0, S_1, S_2, ...$ be such an infinite branch.

Consider the sets of labeled formulas and relational atoms

$$\mathbf{\Gamma} \equiv \bigcup_{0 \leq i} S_i^a \qquad \mathbf{\Delta} \equiv \bigcup_{0 \leq i} S_i^s$$

based on such branch.

A Kipke model is constructed which forces all formulas in $\mathbf{\Gamma}$ and no formula in $\mathbf{\Delta}$.

Consider the following Kripke model $\mathfrak{M} =< W, R, V >$:

- W is the set of labels in $\mathbf{\Gamma}$ and $\mathbf{\Delta}$.
- R consists of all relational atoms in $\mathbf{\Gamma}$.
- For all atoms p: $V(p, w) = 1 \iff w : p \in \mathbf{\Gamma}$.

We can now show with induction on the complexity of the formulas that $\mathfrak{M}, w \vDash A$ if $w : A \in \mathbf{\Gamma}$ and $\mathfrak{M}, w \nvDash A$ if $w : A \in \mathbf{\Delta}$.

Case $w : A = w : p$ and $w : A \in \mathbf{\Gamma}$:
By the definition of the Kripke model if $w : p \in \mathbf{\Gamma}$ then p is also forced in the model in world w and thus $\mathfrak{M}, w \vDash p$.

Case $w : A = w : p$ and $w : A \in \mathbf{\Delta}$:
If $w : p \in \mathbf{\Delta}$ then $w : p$ cannot also be in $\mathbf{\Gamma}$, otherwise the branch would have an axiom in it which is a contradiction. This is because there is no way in the logic to remove an atomic formula in the branch if we look from the root to the leaf, the atomic propositions are upward cumulative. So if $w : p$ is also somewhere in $\mathbf{\Delta}$ there has to be a sequent in the branch such that $w : p, \Gamma \Rightarrow w : p, \Delta$ which is an axiom. This means that if $w : p \in \mathbf{\Delta}$, then $\mathfrak{M}, w \nvDash p$.

Induction hypothesis:
If A is of the form $B \circ C$ where $\circ \in \{\wedge, \vee, \rightarrow\}$ or A is of the form $\circ B$ where $\circ \in \{\Box, \neg\}$, we know that for the formulas B and C that $\mathfrak{M}, w \vDash B, C$ if $w : B, w : C \in \mathbf{\Gamma}$ and $\mathfrak{M}, w \nvDash B, C$ if $w : B, w : C \in \mathbf{\Delta}$.

Case $w : A = w : B \wedge C$ and $w : A \in \mathbf{\Gamma}$:
If $w : B \wedge C \in \mathbf{\Gamma}$, then by the fact that our procedure was infinite and all rules are applied

as much as possible, it must be the case that $w : B$ and $w : C \in \mathbf{\Gamma}$ by the use of rule $L\wedge$. By the induction hypothesis $\mathfrak{M}, w \models B$ and $\mathfrak{M}, w \models C$. Therefore $\mathfrak{M}, w \models B \wedge C$.

Case $w : A = w : B \vee C$ and $w : A \in \mathbf{\Gamma}$:
If $w : B \vee C \in \mathbf{\Gamma}$, then by the fact that our procedure was infinite and all rules are applied as much as possible, it must be the case that $w : B$ or $w : C \in \mathbf{\Gamma}$ by the use of rule $L\vee$. By the induction hypothesis then $\mathfrak{M}, w \models B$ or $\mathfrak{M}, w \models C$. Therefore $\mathfrak{M}, w \models B \vee C$.

Case $w : A = w : B \rightarrow C$ and $w : A \in \mathbf{\Gamma}$:
If $w : B \rightarrow C \in \mathbf{\Gamma}$, then by the fact that our procedure was infinite and all rules are applied as much as possible, it must be the case that $w : C \in \mathbf{\Gamma}$ or $w : B \in \mathbf{\Delta}$ by the use of rule $L \rightarrow$. By the induction hypothesis then $\mathfrak{M}, w \not\models B$ or $\mathfrak{M}, w \models C$. Therefore $\mathfrak{M}, w \models B \rightarrow C$.

Case $w : A = w : \neg B$ and $w : A \in \mathbf{\Gamma}$:
If $w : \neg B \in \mathbf{\Gamma}$, then by the fact that our procedure was infinite and all rules are applied as much as possible, it must be the case that $w : B \in \mathbf{\Delta}$ by the use of rule $L\neg$. By the induction hypothesis then $\mathfrak{M}, w \not\models B$. Therefore $\mathfrak{M}, w \models \neg B$.

Case $w : A = w : \Box B$ and $w : A \in \mathbf{\Gamma}$:
If $w : \Box B \in \mathbf{\Gamma}$, we consider all the relational atoms $wRx \in \mathbf{\Gamma}$. If there are none, then no world is accessible from world w, so $w \models \Box B$ trivially holds. If there are relational atoms $wRx \in \mathbf{\Gamma}$, then by the fact that our procedure was infinite and all rules are applied as much as possible, it must be the case that $x : B \in \mathbf{\Gamma}$ by the use of rule $L\Box$ for every such relational atom. By the induction hypothesis then $\mathfrak{M}, x \models B$ for every world x for which wRx. Therefore $\mathfrak{M}, w \models \Box B$.

Case $w : A = w : B \wedge C$ and $w : A \in \mathbf{\Delta}$:
If $w : B \wedge C \in \mathbf{\Delta}$, then by the fact that our procedure was infinite and all rules are applied as much as possible, it must be the case that $w : B$ or $w : C \in \mathbf{\Delta}$ by the use of rule $R\wedge$. By the induction hypothesis $\mathfrak{M}, w \not\models B$ or $\mathfrak{M}, w \not\models C$. Therefore $\mathfrak{M}, w \not\models B \wedge C$.

Case $w : A = w : B \vee C$ and $w : A \in \mathbf{\Delta}$:
If $w : B \vee C \in \mathbf{\Delta}$, then by the fact that our procedure was infinite and all rules are applied as much as possible, it must be the case that $w : B$ and $w : C \in \mathbf{\Delta}$ by the use of rule $R\vee$. By the induction hypothesis then $\mathfrak{M}, w \not\models B$ and $\mathfrak{M}, w \not\models C$. Therefore $\mathfrak{M}, w \not\models B \vee C$.

Case $w : A = w : B \rightarrow C$ and $w : A \in \mathbf{\Delta}$:
If $w : B \rightarrow C \in \mathbf{\Delta}$, then by the fact that our procedure was infinite and all rules are applied as much as possible, it must be the case that $w : C \in \mathbf{\Delta}$ and $w : B \in \mathbf{\Gamma}$ by the use of rule $R \rightarrow$. By the induction hypothesis then $\mathfrak{M}, w \models B$ and $\mathfrak{M}, w \not\models C$. Therefore $\mathfrak{M}, w \not\models B \rightarrow C$.

Case $w : A = w : \neg B$ and $w : A \in \mathbf{\Delta}$:
If $w : \neg B \in \mathbf{\Delta}$, then by the fact that our procedure was infinite and all rules are applied as much as possible, it must be the case that $w : B \in \mathbf{\Gamma}$ by the use of rule $R\neg$. By the induction hypothesis then $\mathfrak{M}, w \vDash B$. Therefore $\mathfrak{M}, w \nvDash \neg B$.

Case $w : A = w : \Box B$ and $w : A \in \mathbf{\Delta}$:
If $w : \Box B \in \mathbf{\Gamma}$, then by the fact that our procedure was infinite and all rules are applied as much as possible, it must be the case that $x : B \in \mathbf{\Gamma}$ for some x by the use of rule $R\Box$, and we have the relational atom $wRx \in \mathbf{\Gamma}$. By the induction hypothesis then $x \nvDash B$ for a world x which is accessible from world w. Therefore $\mathfrak{M}, w \nvDash \Box B$.

For each formula in the to be proven sequent, it is the case that if the formula is in the antecedent, then it is forced in the model, and if it is in the succedent of the sequent, then it is not forced in the model. Therefore, the model is a countermodel to the interpretation of that sequent ■

Corollary 4.3 (completeness)**.** *If a formula A is valid in every Kripke model, then the sequent $\Rightarrow x : A$ it is derivable in* $\mathbf{Tls}_K$

We see here that the proof contains two important parts. First, there is the proof procedure which either finds a proof for the sequent or creates at least one infinite branch. Secondly, there is the countermodel construction, where this infinite branch is used to construct the worlds of the model using the labels in the branch, and forcing the atomic propositions in the antecedents of the sequents in the branch. These two steps are the main parts of a Schütte proof. However, to prove that the created model is indeed a countermodel, it is especially important that the rules were used on all formulas encountered such that we could use induction on the complexity of the formulas in the sequent. This was ensured because the proof procedure used the rules of $\mathbf{Tls}_K$ as much as possible. We will see this essential part of the proof's argument later come back as a separate property of a derivation's branch: saturation.

4.2 What is a Schütte proof?

In a Schütte proof of completeness for a sequent system there are three steps which are important. Two of which we saw already explicitly in the example. First is the *proof procedure*, this is the procedure which is used to create a derivation tree which either results in a proof, or will have the information for us to create countermodels. All derivation trees consist, like proofs, of sequent, tree-hypersequent or labeled sequent nodes which are connected by the rules of the calculus, where the root node is the 'to be proven' sequent. In the proof of completeness for $\mathbf{Tls_K}$ in the previous section we saw one of these procedures.

Next it is important that from this proof procedure we can derive a branch which is *rule-saturated* if there is no proof for the root sequent. The the branch itself needs to be saturated for all the rules, meaning that we used the rules in all possible ways at least once. Generally this means that the proof procedure is designed in such a way that each branch will either end with an axiom or be saturated for the rules, just as in the example. But besides rule-saturation one can also define a more semantic version of saturation which is based on the interpretation of the different connectives in Kripke semantics in a sequent. Because of the closeness of the rules of sequent calculi to the semantic interpretation of the connectives in Kripke semantics, most of the time this kind of saturation follows directly from the rule application. In this thesis we will treat this kind of saturation separate, because by separating these it is possible to show what is common to the saturation in all the different calculi, and where the calculi differ: their rules. To do this we will introduce a intermediate step between the rule-saturation of the branch and the countermodel construction. We will introduce what we will call the saturation of a *Schütte set of sequents* or a *Schütte set*.

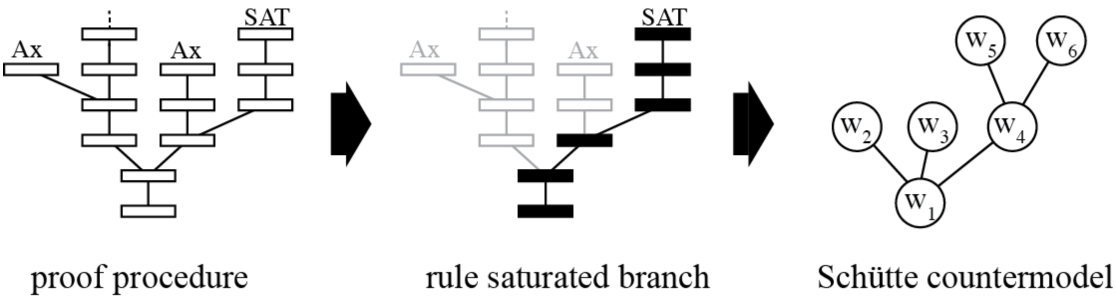


FIGURE 4.1: Full process of creating Schütte countermodels.

The last part in a Schütte proof is to show that this saturated Schütte set of sequents can be used to construct a *Schütte model* that refutes the original sequent, meaning it is a countermodel to the formula which is the interpretation of that sequent. Because of the saturation, it is provable that this model is actually a countermodel. Figure 4.1 shows these three steps in order. In the next three sections (4.2.1-4.2.3) we will define all these concepts.

4.2.1 Proof procedure

Definition 4.4 (proof procedure, failed leaf, failed branch)**.** A procedure is called a **proof procedure** for a sequent system if and only if the following conditions are met

1. The initialization of the procedure includes placing the 'to be proven' sequent at the root of the derivation tree

2. The steps in the proof procedure consist only of applying rules of the sequent system to the leaves in a bottom-up fashion.

If the leaves of the finished derivation tree are all axioms of the sequent system, then the derivation tree is a proof of the root sequent.

If at least one of the leaves of the derivation tree is not an axiom of the system, then the derivation tree if not a proof of the root sequent. These leaves are called **failed leaves**, and the branches from the root sequent to these leaves are called **failed branches**. It can also be the case that a failed branch is infinite, as is the case in the example in the previous section.

Remark 4.5. For the sequent calculi $\mathbf{G3m}_*$ we will allow for a certain amount of backtracking to take place. This is, however, limited in the following way:

In a derivation in $\mathbf{G3m}_*$ it is allowed to backtrack to a sequent which has occurred earlier in the branch of the derivation tree if and only if that sequent is a conclusion of either the $K\Box$ or $4\Box$ rule. And after backtracking, the $K\Box$ or $4\Box$ rule is immediately applied backwards again such that the principal formula $\Box B$ of the newly applied $K\Box$ or $4\Box$ rule has not been used as a principal formula for a $K\Box$ or $4\Box$ rule, applied to this sequent yet.[2]

In a Schütte proof such a procedure is used to show that if there is no proof found by the procedure, the derivation tree can be used to construct a model that refutes the root sequent, and thus showing completeness. For such a Schütte model not the whole derivation tree is needed, we only need one branch of the derivation three. It is important that the branch is a failed branch. This also means that with one derivation it might be possible to create multiple different countermodels, when there are multiple different failed branches in the derivation tree.

4.2.2 Saturation

The notion of saturation is important in Schütte completeness proofs because it will ensure we have generated enough information in a derivation's branch, by using the calculus' rules, to construct a countermodel. We do see notions of saturation for example in [19], [10] and [17]. However, in some Schütte style proofs saturation is not explicitly mentioned such as in our example of in section 4.1, but these proof implicitly include

[2] $\mathbf{G3m}_*$ does not really need the backtracking action itself. It is also possible to directly apply the $K\Box$ or $4\Box$ rule multiple times to a sequent S. Once with each formula $\Box B \in S^s$ as principal. This is done in a disjunctive fashion meaning that if one of these premises is provable, then the sequent S is provable. If it is done this way, there is no need for backtracking. This is for example done in section 2.2.1 of Bílková [4, p.15] and in Chapter 8 of this thesis with the calculus $\mathbf{DY_K}$. Here, however, we will use backtracking to keep the similarity with $\mathbf{Ths}_*$ and $\mathbf{Tls}_*$

saturation in some form. In the Schütte proof in section 4.1, the fact that all rules are used exhaustively in infinite steps makes it saturated.

A branch of a derivation needs to be saturated for all the rules of the logic to be used as a countermodel. But what does saturation exactly mean? First there is rule-saturation. If a branch of a derivation is rule-saturated it means that it is saturated for all the rules of the calculus. All the rules of the calculus are used in as many different ways as possible. But because we are looking at different sequent systems with different rules and sometimes additional syntax besides that of normal sequents, what counts as a rule-saturated branch is different for each calculus. Therefore, we will treat this part of the Schütte method in the next chapter, where we dive deeper into the specific sequent calculi again.

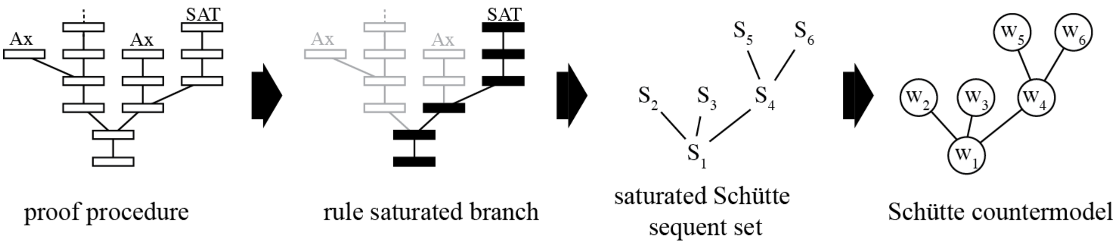


FIGURE 4.2: Full process of creating Schütte countermodels, with addition of the Schütte set of sequents intermediate step.

However it is possible to also define another type of saturation which is not based on the rules of the calculus, but based on the truth conditions of Kripke models.

This kind of saturation is inspired by the definition of saturation described in definition 8.3 and 8.5 of [17, p.57-59] and the proof of theorem 2 of [1, p.936]. This kind of saturation is not based on branches of a derivation, but saturation of sequents and saturation of sets of related sequents. It is possible to create these sets of related sequents from saturated branches, and then create countermodels from these sets. Because this kind of saturation *is* the same for all of the three calculi we will use this as an intermediate step between the saturated branch and the Schütte countermodel, as shown in figure 4.2.

Definition 4.6 (saturated sequent)**.** A sequent S is a **saturated sequent** if the following holds for all formula A,B and atomic proposition p:

1. If $A \wedge B \in S^a$, then $A \in S^a$ and $B \in S^a$.
2. If $A \wedge B \in S^s$, then $A \in S^s$ or $B \in S^s$.
3. If $A \vee B \in S^a$, then $A \in S^a$ or $B \in S^a$.
4. If $A \vee B \in S^s$, then $A \in S^s$ and $B \in S^s$.
5. If $A \to B \in S^a$, then $A \in S^s$ or $B \in S^a$.

6. If $A \to B \in S^s$, then $A \in S^a$ and $B \in S^s$.

7. If $\neg A \in S^a$, then $A \in S^s$.

8. If $\neg A \in S^s$, then $A \in S^a$.

9. There is no atomic proposition p such that $p \in S^a$ and $p \in S^s$

These sequents are saturated for the propositional truth conditions, but not for the truth conditions for the modal operators. To create Kripke models we need more than saturated sequents therefore we will also need a *Schütte set of sequents.* This set needs than also to be saturated for the modal operator.

Definition 4.7 (Schütte set)**.** A Schütte set of sequent $\mathfrak{S} =< \mathcal{S}, \mathcal{R} >$ is a related set of sequent such that:

1. $\mathcal{S}$ is a non empty set of sequents.

2. $\mathcal{R} \subseteq (\mathcal{S} \times \mathcal{S})$ is a binary relation on $\mathcal{S}$.

Definition 4.8 (Saturated Schütte set)**.** A Schütte set of sequents $\mathfrak{S} =< \mathcal{S}, \mathcal{R} >$ is **saturated** if and only if the following conditions are met for every sequent $S \in \mathcal{S}$:

1. S is a saturated sequent

2. If $\Box A \in S^a$, then for all sequents $S_1 \in \mathcal{S}$ such that $S\mathcal{R}S_1$ it is the case that $A \in S_1^a$

3. If $\Box A \in S^s$, then there is a sequent $S_1 \in \mathcal{S}$ such that $S\mathcal{R}S_1$ and $A \in S_1^s$

Remark 4.9. For the modal logics T, $\mathsf{K4}$ and $\mathsf{S4}$ we need extra saturation conditions:

1. $\mathcal{R}$ is reflexive for T and $\mathsf{S4}$

2. $\mathcal{R}$ is transitive for $\mathsf{K4}$ and $\mathsf{S4}$

As said, there is, besides this definition of a saturated Schütte set of sequents, also a way of talking about the saturation of a branch of a derivation for the rules of that calculus. However, we can use the idea of a saturated set of sequents as an additional condition for a proof procedure to be a candidate for a Schütte style completeness proof and the construction of Schütte countermodels.

Definition 4.10. A proof procedure is a possible proof procedure of a Schütte countermodel construction if the following condition is met:

1. Each failed branch in the procedure is saturated for the rules of the calculus.

2. For each failed branch in the procedure the sequents in that failed branch can be combined (in a way specific to the sequent calculus) such that the created Schütte set is a saturated Schütte set.

We will see in Chapter 5 that for the three different type of calculi **G3m**$_*$, **Tls**$_*$ and **Ths**$_*$, if a branch is rule-saturated for the rules of that calculus, it is also possible to use this branch to construct a saturated Schütte set of sequents making condition 2 of the definition follow from the first condition for these calculi. We will at the moment not show explicitly how to construct these saturated Schütte sets of sequents from a failed branch in a proof search for different calculi. This is shown in chapter 5.

4.2.3 Schütte countermodel construction

If a rule-saturated failed branch of a proof procedure can be used to construct a saturated Schütte set of sequents then it can also be used to create a Schütte countermodel. We will now show that we can use such a saturated Schütte set to create a countermodel. If we have a Schütte set, we use it to create a Kripke model which has for each sequent in the set a world corresponding to it, in which all the atomic propositions in the antecedent of that sequent are forced. The relation function between these worlds in the constructed Kripke model is the same as the relation between the sequents in the Schütte set of sequents. This will mean that each world forces all propositions in the antecedent of its corresponding sequent, but none of the propositions in the succedent of the sequent. The relation between a Schütte set of sequents, and a Schütte model is very close, and this idea of sequents *becoming* worlds in the model is important to keep in mind in the next chapter.

Definition 4.11 (Construction of a Schütte countermodel from a saturated Schütte set of sequents)**.** If $\mathfrak{S} =< \mathcal{S}, \mathcal{R} >$ is a saturated Schütte set of sequents then a countermodel $\mathfrak{M}$ can be constructed which is a countermodel so that for each sequent $s \in S$. The construction is as follows:

Construct the following Kripke model $\mathfrak{M} =< W, R, V >$

- $W = \mathcal{S}$; For each $S_i \in \mathcal{S}$ we add an world w_i to W.
- R is a relation over W such that $w_i R w_j \iff S_i \mathcal{R} S_j$.
- For all atoms p: $V(p, w_i) = 1 \iff p \in S_i^a$.

Lemma 4.12. *If $\mathfrak{S} =< \mathcal{S}, \mathcal{R} >$ is a saturated Schütte set of sequents, then the model $\mathfrak{M}$ constructed from $\mathfrak{S}$ according to the construction of definition 4.11 is a countermodel to the interpretation of each of the sequents in $S_i \in \mathcal{S}$. More precisely: $\mathfrak{M}, w_i \nvDash I(S_i)$ for each $S_i \in \mathcal{S}$.*

Proof. We will show that for each sequent $S_i \in \mathcal{S}$ of $\mathfrak{S}$: $\mathfrak{M}, w_i \models A$ if $A \in S_i^a$ and $\mathfrak{M}, w_i \not\models A$ if $A \in S_i^s$. This is shown by induction on the complexity of the formulas.

If A is atomic:
By the construction of the model $\mathfrak{M}$, if $A \in S_i^a$ then $V(A, w_i) = 1$ and $\mathfrak{M}, w_i \models A$. If $A \in S_i^s$ then A cannot be also in S_i^a, based on the fact s_i is a saturated sequent and the requirement (7) of definition 4.6. Because of this and the specific construction of $\mathfrak{M}$, $V(A, w_i) = 0$ and $\mathfrak{M}, w_i \not\models A$

If A is $B \wedge C$ and $A \in S_i^a$:
If $B \wedge C \in S_i^a$, then based on the fact S_i is a saturated sequent and the requirement (1) of definition 4.6 also B and $C \in S_i^a$. By the induction hypothesis $\mathfrak{M}, w_i \models B$ and $\mathfrak{M}, w_i \models C$. Therefore $\mathfrak{M}, w_i \models B \wedge C$.

If A is $B \wedge C$ and $A \in S_i^s$:
If $B \wedge C \in S_i^s$, then based on the fact s_i is a saturated sequent and the requirement (2) of definition 4.6 also $B \in S_i^s$ or $C \in S_i^s$. By the induction hypothesis $\mathfrak{M}, w_i \not\models B$ or $\mathfrak{M}, w_i \not\models C$. Therefore $\mathfrak{M}, w_i \not\models B \wedge C$.

Similar arguments when A is $B \vee C$, $B \rightarrow C$ or $\neg B$ based on the fact the sequent S_i is saturated.

If A is $\Box B$ and $A \in S_i^a$:
If $\Box B \in S_i^a$, then based on the fact that $\mathfrak{S}$ is a saturated Schütte set of sequents and the requirement (2) of definition 4.8 and the construction of the relation R in definition 4.11, all worlds w_j for which $w_i R w_j$, $B \in S_j^a$. By the induction hypothesis for all w_j $\mathfrak{M}, w_j \models B$. Therefore $\mathfrak{M}, w_i \models \Box B$.

If A is $\Box B$ and $A \in S_i^s$:
If $\Box B \in S_i^s$, then based on the fact that $\mathfrak{S}$ is a saturated Schütte set of sequents and the requirement (3) of definition 4.8 and the construction of the relation R in definition 4.11, there is a world w_j for which $w_i R w_j$ and $B \in S_j^s$. By the induction hypothesis $\mathfrak{M}, w_j \not\models B$. Therefore $\mathfrak{M}, w_i \not\models \Box B$.

The interpretation of a sequent $I(S_i) \equiv \bigwedge S_i^a \rightarrow \bigvee S_i^s$. Because $\mathfrak{M}, w_i \models A$ for all $A \in S_i^a$ and $\mathfrak{M}, w_i \not\models A$ for all $A \in S_i^s$, $\mathfrak{M}, w_i \not\models \bigwedge S_i^a \rightarrow \bigvee S_i^s$. ■

If we have created this saturated Schütte set of sequents from a branch of a proof search of a calculus, the completeness of that calculus for models with the proper frame properties follows as a corollary. This is because the sequent at the root of the branch is incorporated in one of the sequents of the Schütte set, which means the Schütte countermodel also refutes the root sequent.

We have seen that a Schütte countermodel is a countermodel created from a saturated Schütte set of sequents which is derived from a rule-saturated failed branch of a proof procedure. It depends on the logic when a branch is rule-saturated and how the Schütte set is constructed out of the branch. In the next chapter we will look at how one can construct a Schütte set of sequent from a branch in a derivation, and that if such a branch is rule-saturated that constructed Schütte set is also saturated.

Chapter 5

Calculus Specific Parts of the Schütte Method

This chapter looks at the calculus specific parts of the Schütte method for countermodel construction. This includes (1) the construction of the Schütte set of sequents from a failed derivation branch and (2) rule saturation for a derivation branch. We will also show that if a failed branch is rule-saturated, it also constructs a saturated Schütte set of sequents with the defined constructions.

In this Chapter, we will also explicitly see where the calculi differ. Because of this one example is treated for all the three different calculi. An example for which a derivation, a Schütte set of sequents and a Schütte countermodel are created. For this purpose, one invalid sequent will be used which is the following:

$$\Box p \Rightarrow \Box p \wedge \Box q, \Box r$$

5.1 G3m

5.1.1 Rule-saturation

Definition 5.1 (Rule-saturation for $\mathbf{G3m}_*$)**.** A branch in a derivation in $\mathbf{G3m}_*$ is saturated for the rules of $\mathbf{G3m}_*$ if all its rules are used in all possible ways. This means the following:

1. The propositional rules are used at least once for each *different* formula, which has $\wedge$, $\vee$, $\neg$ or $\rightarrow$ as its main connective, which occur in the derivation branch between applications of $4\Box$ or $K\Box$ rules. And this formula is used as the principal formula in that propositional rule.

2. In $\mathbf{G3m}_T$ and $\mathbf{G3m}_{S4}$, the rule $T\Box$ is used at least once for each *different* formula $\Box A \in S^a$ which occur in the derivation branch between applications of $4\Box$ or $K\Box$ rules.

3. the rule $K\Box$ or $4\Box$ is used once for *every different* formula $\Box A \in S^s$ occurring in the derivation branch[1].

Example 5.1. *The following derivation is a derivation for $\Box p \Rightarrow \Box p \wedge \Box q, \Box r$ in $\mathbf{G3m}_K$ and has a branch which is saturated for all the rules:*

$$
\dfrac{\dfrac{\overline{p \Rightarrow p}\ Ax}{\Box p \Rightarrow \Box p, \Box r}\ K\Box \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\lightning}{p \Rightarrow r}}{\Box p \Rightarrow \Box q, \Box r}\ K\Box}{p \Rightarrow q}\ Backtracking}{\Box p \Rightarrow \Box q, \Box r}\ K\Box}{\Box p \Rightarrow \Box p \wedge \Box q, \Box r}\ R\wedge
$$

This derivation is not a proof because the second branch of the derivation tree does not end in an axiom. Therefore we can use this branch in the construction of a $\mathbf{G3m}_K$ Schütte countermodel.

5.1.2 Schütte set of sequents construction

Before we can define the construction of a Schütte set from a derivation branch, we first need to define a specific way to split up a $\mathbf{G3m}_*$ branch into segments. This is necessary because specific parts of the branch will be used to make different sequents in the Schütte set, and eventually worlds in the model.

Definition 5.2 (Branch segmentation marking)**.** Let $B = S_1, S_2, ..., S_n$ or $B = S_1, S_2, ...$ be a branch in a derivation in $\mathbf{G3m}_*$. This branch can be partitioned into segments u by cutting the branch between the premise and conclusion of applied $K\Box$ or $4\Box$ rules (depending on which calculus it is). We can mark these segments of a branch in a bottom-up fashion giving us a set of sequent markers U and a binary relation on the segments Q.

The marking of the segments is started with $U_0 = \emptyset$ and $Q_0 = \emptyset$.

We start with marking sequent S_1 and give it the segment-mark o, where $U_1 = U_0 \cup \{o\}$ and $Q_1 = Q_0$.

Suppose S_i is marked met segment-marker u, then we can also mark sequent S_{i+1} with the following segment-marking rules:

[1] This can only be achieved via backtracking.

1. If the $K\Box$ or $4\Box$ is used between S_i and S_{i+1}, sequent $S_i + 1$ will be marked with a new marker u', where $U_{i+1} = U_i \cup \{u'\}$ and $Q_{i+1} = Q_i \cup \{(u, u')\}$.

2. If a propositional rule or $T\Box$ is used between S_i and S_{i+1}, the sequent S_{i+1} will get the same sequent marking u as S_i, where $U_{i+1} = U_i$ and $Q_{i+1} = Q_i$.

3. If backtracking is used between S_i and S_{i+1}, it is the case that S_{i+1} is the same sequent as S_j and $j < i$. This means that S_j has already a segment-mark. Suppose this mark is u'. We will then give the sequent S_{i+1} also the mark u', where $U_{i+1} = U_i$ and $Q_{i+1} = Q_i$.

The final set of segment-markers U and the binary relation Q such that

$$U = \bigcup_{0 \le i \le n} U_i \qquad \text{or with an infinite branch} \qquad U = \bigcup_{0 \le i} U_i\,.$$

$$Q = \bigcup_{0 \le i \le n} Q_i \qquad \text{or with an infinite branch} \qquad Q = \bigcup_{0 \le i} Q_i\,.$$

This results in segments where each segment consists only of sequents which are connected with each other via propositional rules and possible the $T\Box$ rule. Besides this, each segment starts with either the root sequent, or a sequent which is a premise of the $K\Box$ or $4\Box$ rule and ends with a sequent which is either the conclusion of the $K\Box$ or $4\Box$ rule, a backtracking action or the leaf node of the whole branch.

Remark 5.3. If we assume a limit on the $T\Box$ rule by adding the constraint that we use the $T\Box$ rule only finitely many times on the same principal formula within a segment then, because there are only finitely many different principal formulas, this lets us only use the $T\Box$ rule finitely many times, and thus makes each segment finite.

Definition 5.4 (Schütte set of sequents construction for $\mathbf{G3m}_*$)**.** A Schütte set of sequents $\mathfrak{S}^{\mathbf{G3m}} =< \mathcal{S}, \mathcal{R} >$ can be constructed from a possibly infinite branch B of a failed proof tree in $\mathbf{G3m}_*$ in the following way:

Let $B = S_1, S_2, ..., S_n$ or $B = S_1, S_2, ...$ be such a branch of sequents. This branch can be partitioned into segments with marks U and binary relation Q by definition 5.2.

For a segment u let $S_1, S_2, ..., S_m$ be all the sequents marked with u in the branch B. For each segment-mark $u \in U$ a sequent S_u can be created such that the antecedent and succedent of S_u are the following *sets* of formulas:

$$S_u^a = \{A | A \in S_j^a, 1 \le j \le m\} \qquad S_u^s = \{A | A \in S_j^a, 1 \le j \le m\}$$

With these sequents the Schütte set $\mathfrak{S}^{\mathbf{G3m}} =< \mathcal{S}, \mathcal{R} >$ can be created where:

$$\mathcal{S} = \{S_u | u \in U\}$$

The binary relation $\mathcal{R}$ over the sequents in this set is defined in the following way:

$S_u \mathcal{R} S_{u'}$ if and only if $(u, u') \in Q$

Besides this, if we are dealing with $\mathbf{G3m}_T$ we take the reflexive closure of $\mathcal{R}$, if we are dealing with $\mathbf{G3m}_{K4}$ we take the transitive closure of $\mathcal{R}$ and if we are dealing with $\mathbf{G3m}_{S4}$ we take the reflexive and transitive closure of $\mathcal{R}$.

We can now use the Schütte set construction of definition 5.4 for the construction of the set which comes with the failed saturated branch in example 5.1. The branch in this case the sequence $B =< \Box p \Rightarrow \Box p \wedge \Box q, \Box r >, < \Box p \Rightarrow \Box q, \Box r >, < p \Rightarrow q; \Box p \Rightarrow \Box q, \Box r >$ $, < p \Rightarrow r >$. This branch can be separated into three segments:

1. $u_1 =< \Box p \Rightarrow \Box p \wedge \Box q, \Box r >, < \Box p \Rightarrow \Box q, \Box r >, < \Box p \Rightarrow \Box q, \Box r >$,
2. $u_2 =< p \Rightarrow q >$,
3. $u_3 =< p \Rightarrow r >$.

From these three segments we can create the following Schütte set of sequents $\mathfrak{S}^{\mathbf{G3m}} =< \mathcal{S}, \mathcal{R} >$, where $\mathcal{S} = \{S_{u_1}, S_{u_2}, S_{u_3}\}$ with:

$$S_{u_1} = \Box p \Rightarrow \Box p \wedge \Box q, \Box q, \Box r \quad ,$$

$$S_{u_2} = p \Rightarrow q \quad ,$$

$$S_{u_3} = p \Rightarrow r$$

and the binary relation $\mathcal{R} = \{S_{u_1} \mathcal{R} S_{u_2}, (S_{u_1} \mathcal{R} S_{u_3}\}$. This Schütte set can then be used to create the countermodel seen in figure 5.1 which refutes every sequent of the set S in their respective world, and therefore also refutes the sequent $\Box p \Rightarrow \Box p \wedge \Box q, \Box r$.

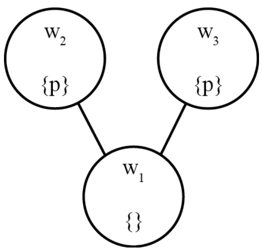


FIGURE 5.1: Schütte countermodel of the sequent $\Box p \Rightarrow \Box p \wedge \Box q, \Box r$.

Proposition 5.5. *If a failed branch in a derivation in* $\mathbf{G3m}_*$ *is saturated for the rules of* $\mathbf{G3m}_*$ *then the Schütte set of sequents* $\mathfrak{S}^{\mathbf{G3m}} =< \mathcal{S}, \mathcal{R} >$ *constructed from it by definition 5.4 is a saturated set of sequents (definition 4.8).*

Proof. This is proven by looking at the conditions of definition 4.8 individually. Starting with the conditions of definition 4.6.

Condition 1 of definition 4.6:
Suppose $B \wedge C$ is in an antecedent of a sequent in the branch segment u. Because of the condition 1 of definition 5.1 we know that the segment is saturated for the $L\wedge$ rule. This means that the rule $L\wedge$ is applied in the segment in at least the following way:

$$\frac{\Gamma, B, C \Rightarrow \Delta}{\Gamma, B \wedge C \Rightarrow \Delta} \; L\wedge$$

According to the construction of definition 5.4, we know therefore that for a sequent $S_u \in \mathcal{S}$ that if $B \wedge C \in S_u^a$, then $B, C \in S_u^a$. This satisfies condition 1 of definition 4.6.

Condition 2 of definition 4.6:
Suppose $B \wedge C$ is in an succedent of a sequent in the branch segment u. Because of the condition 1 of definition 5.1 we know that the segment is saturated for the $R\wedge$ rule. This means that the rule $R\wedge$ is applied in the branch in at least one of the following ways:

$$\frac{\Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, B \wedge C} \; R\wedge \qquad \frac{\Gamma \Rightarrow \Delta, C}{\Gamma\Delta, B \wedge C} \; R\wedge$$

According to the construction of definition 5.4, we know therefore that for a sequent $S_u \in \mathcal{S}$ that if $B \wedge C \in S_u^s$ either $B \in S_u^s$ or $C \in S_u^s$. This satisfies condition 2 of definition 4.6.

The proofs for conditions 3-8 of definition 4.6 and the other propositional connectives are similar to conditions 1 and 2.

Condition 9 of definition 4.6:
Suppose p is in a sequent of segment u in the branch. Because the branch is failed (and atoms are upward cumulative in a segment of the branch) we know that p cannot be in both an antecedent and succedent of any sequent in the branch segment. Because of the construction of definition 5.10, we know therefore that if $p \in S_u^a$ that $p \notin S_u^s$, and if $p \in S_u^s$ that $p \notin S_u^a$ which satisfies condition 7 of definition 4.6.

This shows that condition 1 of 4.8 holds.

Condition 2 of definition 4.8:
Suppose $\Box B$ is in an antecedent of a sequent in the segment u. Because in the segment, formulas of the form $\Box B$ are upwards cumulative we know that each formula $\Box B$ in the antecedent of any sequent in the segment, it is also in the top sequent of the segment. According to the construction of definition 5.4 the only sequents $S'_u \in \mathcal{S}$ for which $S_u \mathcal{R} S_{u'}$ are in the Schütte sequents created from segments which are connected to the top sequent

in branch u with the $K\Box$ or $4\Box$ rule in the following way:

$$\frac{\Gamma \Rightarrow A}{\Sigma, \Box\Gamma \Rightarrow \Phi, \Box A}\ K\Box$$

Where for all formulas of the form $\Box B$ in the antecedent it is the case that $\Box B \in \Box\Gamma$, therefore B is in the antecedent of the root sequent of segment u' and thus $B \in S^a_{u'}$ which satisfies condition 2 of definition 4.8.

Condition 2 of definition 4.8:
Suppose $\Box B$ is in a succedent of a sequent in the segment u. Because in the segment formulas of the form $\Box B$ are upwards cumulative we know that each formula $\Box B$ in the succedent of any sequent in the segment, it is also in the top sequent of the branch. Because of the saturation for the $K\Box$ or $4\Box$ rule we know that the $K\Box$ or $4\Box$ rule is applied at least in the following way:

$$\frac{\Gamma \Rightarrow B}{\Sigma, \Box\Gamma \Rightarrow \Phi, \Box B}\ K\Box$$

Where the sequent $\Gamma \Rightarrow B$ is the bottom sequent of a new segment u'. Because of the construction of $\mathfrak{S}^{\mathbf{G3m}}$ according to definition 5.4 we know there is the sequent $S_{u'}$ for which $S_u \mathcal{R} S_{u'}$ and $B \in S^s_{u'}$ which satisfies condition 3 of definition 4.8. ■

Remark 5.6. If we have a saturated branch in $\mathbf{G3m}_T$, $\mathbf{G3m}_{K4}$ or $\mathbf{G3m}_{S4}$ the transitivity and reflexivity conditions hold trivially. Besides this, the saturated application of the rules $T\Box$ and $4\Box$ ensure that condition 2 of definition 4.8 still holds.

5.2 Ths

5.2.1 Rule-saturation

Definition 5.7. A branch in a derivation in $\mathbf{Ths}_*$ is saturated for the rules of $\mathbf{Ths}_*$ if all its rules are used in all possible ways at least once. This means the following:

1. All propositional rules are used at least once for each different formula in each different sequent that occurs in the tree-hypersequents in the branch.

2. The rule $L\Box$ is used at least once for every different combination of parent-child sequents and different formula $\Box A$ in the antecedent of the parent sequent in the tree-hypersequents in the branch.

3. The rule $R\Box$ is used at least once for every sequent and $\Box A$ in the succedent of that sequent that occurs in the three-hypersequents in the branch.

4. in $\mathbf{Ths}_T$ and $\mathbf{Ths}_{S4}$ the rule *ref* is used at least once for each different sequent and each formula $\Box A$ in occurring in the antecedent of that sequent in the tree-hypersequents in the branch.

5. in $\mathbf{Ths}_{K4}$ and $\mathbf{Ths}_{S4}$ the rule *trans* is used at least once least once for every different combination parent-child sequents and different formula $\Box A$ in the antecedent of the parent sequent in the tree-hypersequents in the branch.

Remark 5.8. With 'different sequent' in the tree-hypersequents in the branch, we mean the intuitive meaning of different sequents which occur throughout the tree-hypersequents in a branch. These are exactly different in their marking as in definition 5.10 which we see later on.

Example 5.2. *The following derivation is a derivation of the example sequent* $\Box p \Rightarrow \Box p \wedge \Box q, \Box r$ *in* $\mathbf{Ths}_K$

$$
\dfrac{
\dfrac{\dfrac{\dfrac{}{\Box p \Rightarrow \Box r / p \Rightarrow p}\,Ax}{\Box p \Rightarrow \Box r / \Rightarrow p}\,L\Box}{\Box p \Rightarrow \Box p, \Box r}\,R\Box
\qquad
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\lightning}{\Box p \Rightarrow / p \Rightarrow r; p \Rightarrow q}}{\Box p \Rightarrow / \Rightarrow r; p \Rightarrow q}\,L\Box}{\Box p \Rightarrow \Box r / p \Rightarrow q}\,R\Box}{\Box p \Rightarrow \Box r / \Rightarrow q}\,L\Box}{\Box p \Rightarrow \Box q, \Box r}\,R\Box
}{\Box p \Rightarrow \Box p \wedge \Box q, \Box r}\,R\wedge
$$

We see that the second branch of this derivation is not an axiom and that it is saturated for all the rules. It is therefore possible to use this branch for the construction of a Schütte countermodel.

5.2.2 Schütte set of sequents construction

Just as with $\mathbf{G3m}_*$ we need a preliminary definition before we can immediately create the Schütte set. Now we do not have to segment parts of the derivation branch, but it is necessary to track the different sequents which are in the tree-hypersequents in the branch. We do this by adding markers to the sequents in the tree-hypersequent along the branch.

Definition 5.9 (Tree-hypersequent branch marking)**.** Let $B = G_1, G_2, ..., G_n$ or $B = G1, G_2, ...$ be a branch of tree-hypersequents connected with the rules of $\mathbf{Ths}_*$. The sequents in the tree-hypersequents in this branch can be recursively marked in a bottom-up fashion, giving us a set of markers M and a binary relation over these markers P.

The marking is started with $M_0 = \emptyset$ and $P_0 = \emptyset$.

We start the marking at G_1. Because G_1 is only a single sequent of the from $\Gamma \Rightarrow \Delta$ we give this sequent the mark o. Where $M_1 = M_0 \cup \{o\}$ and $P_1 = P_0$

Suppose G_i is marked then we can construct a marking for G_{i+1} in the following manner:

For each next tree-hypersequent up in the branch we have the following marking rules:

1. If the $R\Box$ rule is used such that:

$$\dfrac{G[\overbrace{\Gamma \Rightarrow \Delta}^{m} / \overbrace{\Rightarrow A}^{m'}]}{G[\underbrace{\Gamma \Rightarrow \Delta, \Box A}_{m}]}\ R\Box$$

   The newly created sequent in the tree-hypersequent in the premise is marked with the fresh marker m' and all other sequents in the tree-hypersequent in the premise take over their marks from the conclusion tree-hypersequent. $M_{i+1} = M_i \cup \{m'\}$ and $P_{i+1} = P_i \cup (m, m')$

2. If another rule than $R\Box$ is used: All sequents in the tree-hypersequent in the premise take over their marks from the conclusion tree-hypersequent. $M_{i+1} = M_i$ and $P_{i+1} = P_i$

The final set of markers M and the binary relation P such that

$$M = \bigcup_{0 \leq i \leq n} M_i \qquad \text{or with an infinite branch} \qquad M = \bigcup_{0 \leq i} M_i$$

$$P = \bigcup_{0 \leq i \leq n} P_i \qquad \text{or with an infinite branch} \qquad P = \bigcup_{0 \leq i} P_i$$

Definition 5.10 (Combined Sequents $\mathbf{Ths}_*$)**.** The set of combined sequents from a branch B of a failed proof tree in $\mathbf{Ths}_*$ can be constructed in the following way:

Let $B = G_1, G_2, ..., G_n$ or $B = G_1, G_2, ...$ be such a branch of tree-hypersequents. The sequents in the three-hypersequents in this branch can be marked in a bottom-up fashion according to definition 5.9 where we also gain a set of markers M, and a binary relation over these markers P.

We can then split every tree-hypersequent G_i of the branch in its separate marked sequents so that $S_{m,i}$ is the sequent in G_i which is marked with the mark m. For each mark which is created in the marking of the branch we can create the following two sets:

$$\mathbf{\Gamma_m} = \{A | A \in S^a_{(m,i)} \ \wedge \ 1 \leq i \leq n\} \qquad \text{or with an infinite branch:} \qquad \mathbf{\Gamma_m} = \{A | A \in S^a_{(m,i)} \ \wedge \ 1 \leq i\}$$

$$\mathbf{\Delta_m} = \{A | A \in S^s_{(m,i)} \ \wedge \ 0 \leq i \leq n\} \qquad \text{or with an infinite branch:} \qquad \mathbf{\Delta_m} = \{A | A \in S^s_{(m,i)} \ \wedge \ 1 \leq i\}$$

We can then create the sequents $S_m = \mathbf{\Gamma_m} \Rightarrow \mathbf{\Delta_m}$ for each marker $m \in M$. With these sequents we can create the Schütte set $\mathfrak{S}^{\mathbf{Ths}} =< \mathcal{S}, \mathcal{R} >$ such that

$$\mathcal{S} = \{S_m | m \in M\}$$

The relation function $\mathcal{R}$ is defined such that: $S_x \mathcal{R} S_y$ if and only if, $(x, y) \in P$. Besides this, if we are dealing with $\mathbf{Ths}_T$ we take the reflexive closure of $\mathcal{R}$, if we are dealing with $\mathbf{Ths}_{K4}$ we take the transitive closure of $\mathcal{R}$ and if we are dealing with $\mathbf{Ths}_{S4}$ we take the reflexive and transitive closure of $\mathcal{R}$.

With the saturated branch in example 5.2 it is possible to create a Schütte set with definition 5.10. If we mark the sequents with the sequentmarking of definition 5.9 the branch then consists of the following marked tree-hypersequents[2]:

$< \Box p \Rightarrow \Box p \wedge \Box q, \Box r >_{m_1}$

$< \Box p \Rightarrow \Box q, \Box r >_{m_1}$

$< \Box p \Rightarrow \Box r >_{m_1} / <\Rightarrow q >_{m_2}$

$< \Box p \Rightarrow \Box r >_{m_1} / < p \Rightarrow q >_{m_2}$

$< \Box p \Rightarrow>_{m_1} / <\Rightarrow r >_{m_3}; < p \Rightarrow q >_{m_2}$

$< \Box p \Rightarrow>_{m_1} / < p \Rightarrow r >_{m_3}; < p \Rightarrow q >_{m_2}$

Besides this we obtain the binary relation $P = \{(m_1, m_2), (m_1, m_3)\}$. With the construction of definition 5.10 for the sequent set construction we create a sequent set $\mathfrak{S}^{\mathbf{Ths}} =< \mathcal{S}, \mathcal{R} >$ where:

$$\mathcal{S} = \{S_1, S_2, S_3\}$$
$$\mathcal{R} = \{S_1 \mathcal{R} S_2, S_1 \mathcal{R} S_3\}$$

with:

$S_1 = \Box p \Rightarrow \Box p \wedge \Box q, \Box q, \Box r$

$S_2 = p \Rightarrow q$

$S_3 = p \Rightarrow r$

If we use this set of saturated sequents to create a Schütte model we get the same model as with $\mathbf{G3m}_K$, the model in figure 5.1.

Proposition 5.11. *If a failed branch in a derivation in* $\mathbf{Ths}_*$ *is saturated for the rules of* $\mathbf{Ths}_*$ *then the Schütte set of sequents* $\mathfrak{S}^{\mathbf{Ths}} =< \mathcal{S}, \mathcal{R} >$ *constructed from it by definition 5.10 is a saturated (definition 4.8).*

Proof. This is proven by looking at the conditions of definition 4.8 individually. Starting with the conditions of definition 4.6.

Condition 1 of definition 4.6:

Suppose $B \wedge C$ is in the antecedent of a sequent marked with m in the branch. Because

[2]For readability the individual sequents in the tree-hypersequents are surrounded by "<>" and marked.

of the condition 1 of definition 5.7 we know that the branch is saturated for the $L\wedge$ rule. This means that the rule $L\wedge$ is applied in the branch in at least the following way:

$$\frac{G[\overbrace{\Gamma, B, C \Rightarrow \Delta}^{m}]}{G[\underbrace{\Gamma, B \wedge C \Rightarrow \Delta}_{m}]}\ L\wedge$$

According to the construction of definition 5.10, we know therefore that for the sequent $S_m \in \mathcal{S}$, if $B \wedge C \in S^a_m$ that $B, C \in S^a_m$. This satisfies condition 1 of definition 4.6.

Condition 2 of definition 4.6:

Suppose $B \wedge C$ is in the succedent of a sequent marked with m in the branch. Because of the condition 1 of definition 5.7 we know that the branch is saturated for the $R\wedge$ rule. This means that the rule $R\wedge$ is applied in the branch in at least one the following ways:

$$\frac{G[\overbrace{\Gamma \Rightarrow \Delta, B}^{m}]}{G[\underbrace{\Gamma \Rightarrow \Delta, B \wedge C}_{m}]}\ R\wedge \qquad \frac{G[\overbrace{\Gamma \Rightarrow \Delta, C}^{m}]}{G[\underbrace{\Gamma \Rightarrow \Delta, B \wedge C}_{m}]}\ R\wedge$$

According to the construction of definition 5.10, we know therefore for the sequent $S_m \in \mathcal{S}$ that if $B \wedge C \in S^s_m$ that either $B \in S^s_m$ or $C \in S^s_m$. This satisfies condition 2 of definition 4.6.

The proof condition 3-6 of definition 4.6 for the other propositional connectives is similar.

Condition 7 of definition 4.6:

Suppose p is in a sequent marked with m in the branch. Because the branch is failed (and atoms are upward cumulative in the branch) we know that p cannot be in both an antecedent and succedent of a sequent marked with m in the branch. Because of the construction of definition 5.10, we know therefore that if $p \in S^a_m$ that $p \notin S^s_m$, and if $p \in S^s_m$ that $p \notin S^a_m$ which satisfies condition 7 of definition 4.6.

This shows that condition 1 of definition 4.8 holds.

Condition 2 of definition 4.8:

Suppose $\Box B$ is in the antecedent of a sequent marked with m in the branch. Because of the condition 2 of definition 5.7 we know that the branch is saturated for the $R\Box$ rule. This means that the rule $R\Box$ is applied in the branch at least once for each sequent marked m' which is an immediate child of m in the branch in the following way:

$$\frac{G[\overbrace{\Gamma, \Box B \Rightarrow \Delta}^{m} / \overbrace{\Gamma', B \Rightarrow \Delta'}^{m'}]}{G[\underbrace{\Gamma, \Box B \Rightarrow \Delta}_{m} / \underbrace{\Gamma' \Rightarrow \Delta'}_{m'}]}\ L\Box$$

According to the construction of definition 5.10, we therefore know that if $\Box B \in S_m^a$ that for all sequents m' such that $S_m \mathcal{R} S_{m'}$, $B \in S_{m'}^a$. This satisfies condition 2 of definition 4.8.

Condition 3 of definition 4.8:
Suppose $\Box B$ is in the succedent of a sequent marked with m in the branch. Because of the condition 3 of definition 5.7 we know that the branch is saturated for the $L\Box$ rule. This means that the rule $L\Box$ is applied in the branch at least once in the following way:

$$\frac{G[\overbrace{\Gamma \Rightarrow \Delta}^{m} / \overbrace{\Rightarrow A}^{m'}]}{G[\underbrace{\Gamma \Rightarrow \Delta, \Box A}_{m}]} R\Box$$

According to the construction of definition 5.10, we know therefore that if $\Box B \in S_m^s$ there is a sequent m' such that $S_m \mathcal{R} S_{m'}$ and $B \in S_{m'}^s$. This satisfies condition 3 of definition 4.8. ■

Remark 5.12. If we have a saturated branch in $\mathbf{Ths}_T$, $\mathbf{Ths}_{K4}$ or $\mathbf{Ths}_{S4}$ the transitivity and reflexivity conditions hold trivially. Besides this, the saturated application of the rules *ref* and *trans* ensures that condition 2 of definition 4.8 still holds.

5.3 Tls

5.3.1 Rule-saturation

Definition 5.13. A branch in a derivation in $\mathbf{Tls}_*$ is saturated of the rules of $\mathbf{Tls}_*$ if all its rules are used in all possible ways. This means the following:

1. All propositional rules are used at least once for each different prefixed formula that occurs in the branch.
2. The rule $L\Box$ is used at least once for every different combination of prefixed formula $x : \Box A$ and relational atom xRy which occur in the antecedent of a sequent in the branch.
3. The rule $R\Box$ is used at least once for every different prefixed formula $x : \Box$ which occur in the succedent of a sequent in the branch.
4. In $\mathbf{Tls}_T$ and $\mathbf{Tls}_{S4}$ the rule *ref* is used at least once for each different formula $x : \Box A$ which occurs in the branch.
5. In $\mathbf{Tls}_{K4}$ and $\mathbf{Tls}_{S4}$ the rule *trans* is used at least once for every different combination of prefixed formula $x : \Box A$ and relational atom xRy which occur in the antecedent of a sequent in the branch.

Example 5.3. *The following derivation is a derivation for the example sequent* $\Box p \Rightarrow \Box p \wedge \Box q, \Box r$ *in* $\mathbf{Tls}_K$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{xRy, x:\Box p, y:p \Rightarrow x:\Box r, y:p}\,Ax}{xRy, x:\Box p \Rightarrow x:\Box r, y:p}\,L\Box}{x:\Box p \Rightarrow x:\Box p, x:\Box r}\,R\Box \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\lightning}{xRy, xRz, x:\Box p, y:p, z:p \Rightarrow y:q, z:r}\,L\Box}{xRy, xRz, x:\Box p, y:p \Rightarrow y:q, z:r}\,R\Box}{xRy, x:\Box p, y:p \Rightarrow x:\Box r, y:q}\,L\Box}{xRy, x:\Box p \Rightarrow x:\Box r, y:q}\,R\Box}{x:\Box p \Rightarrow x:\Box q, x:\Box r}\,R\Box}{x:\Box p \Rightarrow x:\Box p \wedge \Box q, x:\Box r}\,R\wedge$$

As with the example derivation in $\mathbf{G3m}_K$ and $\mathbf{Ths}_K$, the second branch does not end in an axiom and is saturated. This branch can be used to create a Schütte countermodel. Notice that this derivation has much in common with the derivation in $\mathbf{Ths}_K$. But, except for the syntax of '/' and ';' between the different sequents, here the sequents are mixed together in one sequent and the labels show how we can pick them apart. The relational atoms take the place of the meaning of the '/' syntax.

5.3.2 Schütte set of sequents construction

In **G3m** it was necessary to create a segment-marking of the branch, in **Ths** was necessary to create a sequent marking of the Tree-hypersequents but for **Tls** it is not necessary to create additional markings because the markings are already explicitly apparent in the labeled sequents. These markings are the labels and thus the labels will be used to create different sequents for the Schütte set of sequents.

Definition 5.14 (Combined Sequents **Tls**)**.** The set of combined sequents from a branch B of a failed proof tree in $\mathbf{Tls}_*$ can be constructed in the following way:

Let $B = S_1, S_2, ..., S_n$ or $B = S_1, S_2, ...$ be such a branch of labeled sequents. From this branch two sets $\mathbf{\Gamma}$ and $\mathbf{\Delta}$ of labeled formulas can be created, and a set of relational atoms $\mathbf{R}$ such that:

$\mathbf{\Gamma} = \{x : A \mid x : A \in S_i^a, 1 \leq i \leq m\}$ or with an infinite branch: $\mathbf{\Gamma} = \{x : A \mid x : A \in S_i^a, 1 \leq i\}$

$\mathbf{\Delta} = \{x : A \mid x : A \in S_i^s, 1 \leq i \leq m\}$ or with an infinite branch: $\mathbf{\Delta} = \{x : A \mid x : A \in S_i^s, 0 \leq i\}$

$\mathbf{R} = \{xRy \mid xRy \in S_i^a, 1 \leq i \leq m\}$ or with an infinite branch: $\mathbf{R} = \{xRy \mid xRy \in S_i^a, 1 \leq i\}$

Construct the set of combined sequents by making the following combined sequents $S_x = \Gamma_x \Rightarrow \Delta_x$ for each label x that is occurring in the sets $\mathbf{\Gamma}$ and $\mathbf{\Delta}$ such that:

$$\Gamma_x = \{A | x : A \in \mathbf{\Gamma}\}$$
$$\Delta_x = \{A | x : A \in \mathbf{\Delta}\}$$

With these sequents we can create the the Schütte set $\mathfrak{S} =< \mathcal{S}, \mathcal{R} >$ where

$$\mathcal{S} = \{S_x | x \text{ is a label occurring in } \mathbf{\Gamma}, \mathbf{\Delta} \text{ or } \mathbf{R}\}$$

The relation function $\mathcal{R}$ over this set of combined sequents is defined as:

$$S_x \mathcal{R} S_y \text{ if and only if } xRy \in \mathbf{R}$$

Besides this, if we are dealing with $\mathbf{Tls}_T$ we take the reflexive closure of $\mathcal{R}$, if we are dealing with $\mathbf{Tls}_{K4}$ we take the transitive closure of $\mathcal{R}$ and if we are dealing with $\mathbf{Tls}_{S4}$ we take the reflexive and transitive closure of $\mathcal{R}$.

It is now possible to use the saturated branch of example 5.3 to create a Schütte set of sequents, and a countermodel. If we first create the sets $\mathbf{\Gamma}$, $\mathbf{\Delta}$ and $\mathbf{R}$.

$$\begin{aligned} \mathbf{\Gamma} &= \{x : \Box p, y : p, z : p\} \\ \mathbf{\Delta} &= \{x : \Box p \wedge \Box q, x : \Box q, x : \Box r, y : q, z : r\} \\ \mathbf{R} &= \{xRy, xRz\} \end{aligned}$$

With this create a Schütte sequent set $\mathfrak{S}^{\mathbf{Tls}} =< \mathcal{S}, \mathcal{R} >$ where:

$$\begin{aligned} \mathcal{S} &= \{S_x, S_y, S_z\} \\ \mathcal{R} &= \{S_x \mathcal{R} S_y, S_x \mathcal{R} S_z\} \end{aligned}$$

With:
$S_x = \Box p \Rightarrow \Box p \wedge \Box q, \Box q, \Box r$
$S_y = p \Rightarrow q$
$S_z = p \Rightarrow r$

If we use this set of saturated sequents to create a Schütte model we get again the same model as with G3m, the model in figure 5.1 where $x = 1, y = 2$ and $z = 3$.

Proposition 5.15. *If a branch in a derivation in* $\mathbf{Tls}_*$ *is saturated for the rules of* $\mathbf{Tls}_*$ *then the Schütte set of sequents* $\mathfrak{S}^{\mathbf{Tls}} =< \mathcal{S}, \mathcal{R} >$ *constructed from it by definition 5.14 is saturated (definition 4.8).*

Proof. This is proven by looking at the conditions of definition 4.8 individually. Starting with the conditions of definition 4.6.

Condition 1 of definition 4.6:
Suppose $x : B \wedge C$ is in an antecedent of a sequent in the branch. Because of the condition 1 of definition 5.13 we know that the branch is saturated for the $L\wedge$ rule. This means

that the rule $L\wedge$ is applied in the branch in at least the following way:

$$\frac{\Gamma, x : B, x : C \Rightarrow \Delta}{\Gamma, x : B \wedge C \Rightarrow \Delta} L\wedge$$

According to the construction of definition 5.14, we know therefore that for a sequent $S_x \in \mathcal{S}$, if $B \wedge C \in S_x^a$ that $B, C \in S_x^a$. This satisfies condition 1 of definition 4.6.

Condition 2 of definition 4.6:
Suppose $x : B \wedge C$ is in an succedent of a sequent in the branch. Because of the condition 1 of definition 5.13 we know that the branch is saturated for the $R\wedge$ rule. This means that the rule $R\wedge$ is applied in the branch in at least one of the following ways:

$$\frac{\Gamma \Rightarrow \Delta, x : B}{\Gamma \Rightarrow \Delta, x : B \wedge C} L\wedge \qquad \frac{\Gamma \Rightarrow \Delta, x : C}{\Gamma \Rightarrow \Delta, x : B \wedge C} L\wedge$$

According to the construction of definition 5.14, we know therefore that for a sequent $S_x \in \mathcal{S}$, if $B \wedge C \in S_x^s$ that either $B \in S_x^s$ or $C \in S_x^s$. This satisfies condition 1 of definition 4.6.

The proof condition 3-6 of definition 4.6 for the other propositional connectives is similar.

Condition 7 of definition 4.6:
Suppose $x : p$ is in a sequent in the branch. Because the branch is failed (and atoms are upward cumulative in the branch) we know that $x : p$ cannot be in both an antecedent and succedent of a sequent in the branch. Because of the construction of definition 5.14, we know therefore that for each sequent $S_x \in \mathcal{S}$, if $p \in S_x^a$ then $p \notin S_x^s$, and if $p \in S_x^s$ then $p \notin S_x^a$ which satisfies condition 7 of definition 4.6.

This shows that condition 1 of definition 4.8 holds. **Condition 2 of definition 4.8:**
Suppose $x : \Box B$ is in an antecedent of a sequent in the branch. Because of the condition 3 of definition 5.13 we know that the branch is saturated for the $R\Box$ rule. This means that the rule $R\Box$ is applied in the branch at least once for each relational atom in xRy which occur in an antecedent of a sequent in the branch in the following way:

$$\frac{\Gamma, x : \Box B, xRy, y : B \Rightarrow \Delta}{\Gamma, x : \Box B, xRy \Rightarrow \Delta} L\Box$$

According to the construction of definition 5.14, we know therefore that for each sequent $S_x \in \mathcal{S}$, if $\Box B \in S_x^a$ that for all sequents S_y such that $S_x \mathcal{R} S_y$, $B \in S_y^a$. This satisfies condition 2 of definition 4.8.

Condition 3 of definition 4.8:
Suppose $x : \Box B$ is in an succedent of a sequent in the branch. Because of the condition 2 of definition 5.13 we know that the branch is saturated for the $L\Box$ rule. This means

that the rule $L\Box$ is applied in the branch at least once in the following way:

$$\frac{\Gamma, xRy \Rightarrow \Delta, y : B}{\Gamma \Rightarrow \Delta, x : \Box B} R\Box$$

According to the construction of definition 5.14, we know therefore for each sequent $S_x \in \mathcal{S}$, that if $\Box B \in S^s_x$ that there is a sequent S_y such that $S_x \mathcal{R} S_y$ and $B \in S^s_y$. This satisfies condition 3 of definition 4.8. ∎

Remark 5.16. If we have a saturated branch in $\mathbf{Tls}_T$, $\mathbf{Tls}_{K4}$ or $\mathbf{Tls}_{S4}$ the transitivity and reflexivity conditions hold trivially. Besides this, the saturated application of the rules *ref* and *trans* ensure that condition 2 of definition 4.8 still holds.

5.4 Differences in Schütte set of sequents construction

In the construction of the Schütte sets there is one difference between the three different calculi which is most important. This difference is in what parts of the sequents in the reduction tree the sequent in the Schütte set, which eventually will represent worlds in the countermodel. What needs to be combined of the derivation tree to create a Schütte sequent for these three calculi differs in what 'level' of syntax we need to look at in these derivation trees.

First, we have **G3m**, where whole segments of the branch need to be combined together to create the Schütte sequents and countermodel worlds. This is the highest syntactic 'level' at which these Schütte sequents are constructed from. It is on the level of the derivation. In **G3m**, one segment of a derivation stands for one world in a model, where the $K\Box$(or $4\Box$) rule splits these segments. From this perspective it is obvious that we only have one modal rule which expresses the whole relation between different worlds [3]. The fact that whole parts of a derivation represent a world also gives rise to the need of backtracking to be able to create countermodels and the problem of having a $\mathbf{G3m}_B$ calculus with also symmetric relation between worlds.

The Tree-hypersequent calculi have different sequents which represent the Schütte sequents. Because the calculus is not about sequent themselves but hypersequents, These sequents representing Schütte sequents and worlds in the model can be written all along the branch of the derivation. This makes it possible to have multiple worlds present beside each other in one node of the derivation tree. The sequent is the 'level' of syntax in which the tree-hypersequent operates. Because of this, it is important in the Schütte set of sequents construction to mark the sequents along the derivation to show what Schütte sequent they eventually belong to.

[3] Sometimes we have an extra rule, the $T\Box$ rule, but this rule is only about the relation of that world with itself

The labeled sequent calculi go one step of syntax smaller to represent the Schütte sequents. Labeled sequent calculi label the formula in the sequents. By going one step smallest than sequents themselves it is possible to put all the labeled formula in one sequent per node in the derivation tree. Here the labels of the formula will indicate to what Schütte sequent the formula belongs, and in which world in the Schútte model it will be forced or not forced (depending on whether the formula is in the antecedent or succedent).

These three calculi differ in what level of syntax they encode the Schütte sequents or worlds of the model. Is it on the derivation level, sequent level or formula level. And by this several different things are possible in the last two calculi which are not possible in the first. For example, the tree-hypersequent calculi and the labeled calculi are able to also create calculi for symmetric logics, while this is not possible for the normal modal sequent calculus **G3m**. This is because multiple parts of the model (multiple Schütte sequents) can be represented in one node of a derivation in these calculi, and any transference of information between these worlds can be encoded in rules for these calculi. This is not possible for the **G3m** calculi, which only can transfer information in one direction, because if we get to another world, the information about the last world is not in the leaf nodes of the derivation anymore.

In the next chapter, we will see that this difference in approach to constructing the Schütte sequents, whether it is done either on the derivation, sequent or formula level, is only a matter of taste. We will show that these derivation branches can be translated into one another and still produce the same Schütte sets of sequents and eventually Schütte countermodels.

Chapter 6

Schütte Model Equivalence of G3m, Ths and Tls

In this chapter, we will prove the main theorem of this thesis which shows the close relationship between the three different sequent systems based on Schütte models.

Theorem 6.1. $\mathfrak{M}$ *is a* $\mathbf{G3m}_*$ *Schütte model for the sequent* $\Gamma \Rightarrow \Delta$*, if and only if* $\mathfrak{M}$ *is a* $\mathbf{Ths}_*$ *Schütte model for the equivalent tree hypersequent* $\Gamma \Rightarrow \Delta$ *if and only if* $\mathfrak{M}$ *is a* $\mathbf{Tls}_*$ *Schütte model for the equivalent labeled sequent* $x : \Gamma \Rightarrow x : \Delta$*, where* $*$ *is either* K*,* T*,* K4 *or* S4*.*

We prove this by showing that one can translate branches of derivations in one calculus to another and still keep exactly the same Schütte set which is created from that branch, and thus also keeps the same Schütte countermodel. Because we have three different calculi it is not needed to proof al six different directions, we can do just three directions and this is enough to proof al six. We will prove the directions shown in figure 6.1.

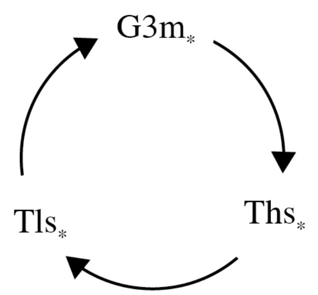


FIGURE 6.1: Directions in which the main theorem will be proven.

Notice that the theorem only talks about basic sequents or their translation into the tree-hypersequent or labeled framework. The theorem will not hold if we also include tree-hypersequents with more than one sequent or labeled sequents with more than one label. This is mainly done because nodes with multiple sequents do not exist in **G3m** and are therefore not properly translatable in this kind of calculi.

6.1 Translation of failed branches

6.1.1 G3m to Ths

To transform a branch of a derivation in $\mathbf{G3m}_*$ to a branch in $\mathbf{Ths}_*$, we use a translation of the applied rules in the branch in a bottom-up fashion starting at the root sequent of the branch. However, to know to which sequent(s) in a tree-hypersequent the translated rule needs to be applied, we make use of the segment-marking for the $\mathbf{G3m}_*$ branch according to definition 5.2 and the sequent marking of definition 5.9 for the tree-hypersequents in the translated $\mathbf{Ths}_*$ branch.

Definition 6.2 (Translation Function $\mathbf{G3m}_* \rightarrow \mathbf{Ths}_*$)**.** Here we define the translation function $\mathcal{F}$ from $\mathbf{G3m}_*$ to $\mathbf{Ths}_*$. The function uses as input a saturated failed branch of a derivation tree of a sequent $S_1 = \Gamma \Rightarrow \Delta$ in $\mathbf{G3m}_*$ and it returns a derivation branch in $\mathbf{Ths}_*$ for the tree-hypersequent $G_1 = \Gamma \Rightarrow \Delta$.

Let $B^{G3m} = S_1, S_2, ..., S_n$ or $B^{G3m} = S_1, S_2, ...$ be such a saturated branch of sequents from a derivation in $\mathbf{G3m}_*$.

The function works in a step by step translation of the branch $\mathbf{G3m}_*$ starting at S_1 and then translating applications of rules in the branch B^{G3m} in a bottom-up fashion. During the steps of the function, we keep track of the part of B^{G3m} that is translated as B^t, and the translation itself as B^{Ths}. The function stops if it reaches the top of the branch. The function $\mathcal{F}$ starts at step 0.

Step 0**:**
Sequent $S_1 = \Gamma \Rightarrow \Delta$ at the root of the branch B^{G3m} is translated into the same tree-hypersequent $G_1 = \Gamma \Rightarrow \Delta$. We mark the sequent $\Gamma \Rightarrow \Delta$ which G_1 solely consists of with the same marking as the segment-mark of S_1. Create the translation branch $B_0^{Ths} = G_1$. Also create the already translated subbranch $B_0^t = S_1$. Go to step 1.

Step $i > 0$**:**
Suppose sequent S_j is the last sequent of B^t. What is done in this step depends on which rule is applied backward from sequent S_j to S_{j+1} of B^{G3m}. We therefore distinguish different cases for each applied rule.

Case 1. $L\wedge$:
The rule $L\wedge$ rule is used between S_j and S_{j+1} where both sequents are in segment m in

the following way:[1]

$$\begin{array}{rc} m: & \Gamma, A, B \Rightarrow \Delta \\ \hline m: & \Gamma, A \wedge B \Rightarrow \Delta \end{array} \; L\wedge$$

This is translated in an application of the $L\wedge$ rule of $\mathbf{Ths}_*$ to the sequent m in the last tree-hypersequent of the branch B^{Ths} in the following way:

$$\frac{G[\overbrace{\Gamma, A, B \Rightarrow \Delta}^{m}]}{G[\underbrace{\Gamma, A \wedge B \Rightarrow \Delta}_{m}]} \; L\wedge$$

We add this bottom-up application of the $L\wedge$ rule to the translation branch B^{Ths}_{i-1} to get the branch B^{Ths}_i. We add S_{j+1} to the already translated subbranch such that $B^t_i = B^t_{i-1}, S_{j+1}$. Then go to step i+1.

Case 2. $R\wedge$:

The rule $R\wedge$ rule is used between S_j and S_{j+1} where both sequents are in segment m in one of the following ways, based on which of the two premise sequents of the rule is in the branch:

$$\begin{array}{rc} m: & \Gamma \Rightarrow \Delta, A \\ \hline m: & \Gamma \Rightarrow \Delta, A \wedge B \end{array} \; R\wedge_1 \qquad\qquad \begin{array}{rc} m: & \Gamma \Rightarrow \Delta, B \\ \hline m: & \Gamma \Rightarrow \Delta, A \wedge B \end{array} \; R\wedge_2$$

This is translated in an application of the $R\wedge$ rule of $\mathbf{Ths}_*$ to the sequent m in the last tree-hypersequent of the branch B^{Ths} in one of the the following ways:

$$\frac{G[\overbrace{\Gamma \Rightarrow \Delta, A}^{m}]}{G[\underbrace{\Gamma \Rightarrow \Delta, A \wedge B}_{m}]} \; R\wedge_1 \qquad\qquad \frac{G[\overbrace{\Gamma \Rightarrow \Delta, B}^{m}]}{G[\underbrace{\Gamma \Rightarrow \Delta, A \wedge B}_{m}]} \; R\wedge_2$$

We add this application of the $R\wedge$ rule to the translation branch B^{Ths}_{i-1} to get the branch B^{Ths}_i. We add S_{j+1} to the already translated subbranch such that $B^t_i = B^t_{i-1}, S_{j+1}$. Then go to step i+1.

In cases 3. $L\vee$, 4. $R\vee$, 5. $L\rightarrow$, 6. $R\rightarrow$, 7. $L\neg$ and 8. $R\neg$:

If one of the other propositional rules is used between S_j and S_{j+1}, the translation function works in a similar way as in the case of $L\wedge$ and $R\wedge$.

Case 9. $K\Box$:

The $K\Box$ rule is used between S_j and S_{j+1} where sequent S_j is in segment m and sequent S_{j+1} is in segment m' in the following way:

$$\begin{array}{rc} m': & B_1, B_2, ..., B_l \Rightarrow A \\ \hline m: & \Sigma, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow \Phi, \Box A \end{array} \; K\Box$$

[1] The m : before the sequents show the segment-marks of the sequents in the rule

This is translated in an application of the $R\Box$ rule and l applications of the $L\Box$ rule of $\mathbf{Ths}_*$ to the sequent m in the last tree-hypersequent of the branch B^{Ths} in the following way:

$$
\begin{array}{cl}
G[\overbrace{\Sigma, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow \Phi}^{m} \,/\, \overbrace{B_1, B_2, ..., B_l \Rightarrow A}^{m'}] & \\
\hline
\uparrow & L\Box_l \\
\hline
G[\Sigma, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow \Phi / B_1, B_2 \Rightarrow A] & L\Box_{...} \\
\hline
G[\Sigma, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow \Phi / B_1 \Rightarrow A] & L\Box_2 \\
\hline
G[\Sigma, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow \Phi / \Rightarrow A] & L\Box_1 \\
\hline
G[\underbrace{\Sigma, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow \Phi, \Box A}_{m}] & R\Box
\end{array}
$$

where the new sequent created with the $R\Box$ rule is marked with the marker m'. We add this application of the $R\Box$ rule and the l applications of the $L\Box$ rule to the translation branch B^{Ths}_{i-1} to get the branch B^{Ths}_i. We add S_{j+1} to the already translated subbranch such that $B^t_i = B^t_{i-1}, S_{j+1}$. Then go to step i+1.

Case 10. $4\Box$:
The $4\Box$ rule is used between S_j and S_{j+1} where sequent S_j is in segment m and sequent S_{j+1} is in segment m' in the following way:

$$
\begin{array}{rcl}
m': & \Box B_1, \Box B_2, ..., \Box B_l, B_1, B_2, ..., B_l \Rightarrow A & \\
\cline{2-2}
m: & \Sigma, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow \Phi, \Box A & 4\Box
\end{array}
$$

This is translated in an application of the $R\Box$ rule and l applications of the $L\Box$ and *trans* rule of $\mathbf{Ths}_*$ to the sequent m in the last tree-hypersequent of the branch B^{Ths} in the following way:

$$
\begin{array}{cl}
G[\overbrace{\Sigma, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow \Phi}^{m} \,/\, \overbrace{B_1, B_2, ..., B_l, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow A}^{m'}] & \\
\hline
G[\Sigma, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow \Phi / B_1, B_2, ..., B_l, \Box B_1, \Box B_2, ... \Rightarrow A] & trans_l \\
\hline
\uparrow & L\Box_l \\
\hline
G[\Sigma, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow \Phi / B_1, B_2, \Box B_1, \Box B_2 \Rightarrow A] & L\Box_{...} \\
\hline
G[\Sigma, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow \Phi / B_1, B_2, \Box B_1 \Rightarrow A] & trans_2 \\
\hline
G[\Sigma, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow \Phi / B_1, \Box B_1 \Rightarrow A] & L\Box_2 \\
\hline
G[\Sigma, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow \Phi / B_1 \Rightarrow A] & trans_1 \\
\hline
G[\Sigma, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow \Phi / \Rightarrow A] & L\Box_1 \\
\hline
G[\underbrace{\Sigma, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow \Phi, \Box A}_{m}] & R\Box
\end{array}
$$

where the new sequent created with the $R\Box$ rule is marked with the marker m'. We add this application of the $R\Box$ rule and the l applications of the $L\Box$ and *trans* rule to the translation branch B^{Ths}_{i-1} to get the branch B^{Ths}_i. We add S_{j+1} to the already translated subbranch such that $B^t_i = B^t_{i-1}, S_{j+1}$. Then go to step i+1.

11. $T\Box$:

The rule $T\Box$ rule is used between S_j and S_{j+1} where both sequents are in segment m in the following way:

$$\begin{array}{ll} m: & \dfrac{\Gamma, \Box A, A \Rightarrow \Delta}{\Gamma, \Box A \Rightarrow \Delta} \; T\Box \\ m: & \end{array}$$

This is translated in an application of the *ref* rule of $\mathbf{Ths}_*$ to the sequent m in the last tree-hypersequent of the branch B^{Ths} in the following way:

$$\frac{G[\overbrace{\Gamma, \Box A, A \Rightarrow \Delta}^{m}]}{G[\underbrace{\Gamma, \Box A \Rightarrow \Delta}_{m}]} \; ref$$

We add this application of the *ref* rule to the translation branch B^{Ths}_{i-1} to get the branch B^{Ths}_i. We add S_{j+1} to the already translated subbranch such that $B^t_i = B^t_{i-1}, S_{j+1}$. Then go to step $i + 1$.

12. Backtracking:

If backtracking is used between S_j and S_{j+1}, $B^{Ths}_{i-1} = B^{Ths}_i$. We add S_{j+1} to the already translated subbranch such that $B^t_i = B^t_{i-1}, S_{j+1}$. Then go to step $i + 1$.

The function stops when S_j is the last sequent in the branch B^{G3m} and the translation of B^{G3m} is B^{Ths}_{i-1}.

Lemma 6.3. *Suppose the translation function $\mathcal{F}$ is used on a branch B^{G3m}, then it is the case that after step i of function $\mathcal{F}$ the top sequent of the branch B^t_i which has the segment-mark m is exactly the same sequent as the sequent marked with m in the top tree-hypersequent of the branch B^{Ths}_i.*

Proof. We prove this with induction on the step i of the function $\mathcal{F}$.

If i = 0:

The lemma holds trivially.

Induction hypothesis:

If we are at step $i = n + 1$, we know that at step $i = n$, that the top sequent of the branch B^t_n which has the segment-mark m is exactly the same sequent as the sequent marked with m in the top tree-hypersequent of the branch B^{Ths}_n.

If i=n+1:

We discern cases by the different cases 1-11 of the step $i < 0$ of the translation function $\mathcal{F}$.

Case 1:

If the function executed case 1, it translated the $L\wedge$ rule of $\mathbf{G3m}_*$. Because of the induction hypothesis we know that the top sequent of branch B^t_n with segment-mark m, is the same as the sequent marked m in the top tree-hypersequent of branch B^{Ths}_n. This means that if the rule is translated in the following way:

$$\begin{array}{l} m\colon \\ m\colon \end{array} \dfrac{\Gamma, A, B \Rightarrow \Delta}{\Gamma, A \wedge B \Rightarrow \Delta} L\wedge \quad \rightsquigarrow \quad \dfrac{G[\overbrace{\Gamma, A, B \Rightarrow \Delta}^{m}]}{G[\underbrace{\Gamma, A \wedge B \Rightarrow \Delta}_{m}]} L\wedge$$

we know that the conclusions sequent of the $\mathbf{G3m}_*$ $L\wedge$ rule and the focused sequent m in the tree hypersequent of the conclusion of the $\mathbf{Ths}_*$ $L\wedge$ rule are the same. This make the premise sequents in these rules also the same. This means that in this case, the top sequent of B^t_{n+1} is the same sequent as the sequent in the top tree-hypersequent of B^{Ths}_{n+1} with the same mark.

The cases 2, 3, 4, 5, 6, 7, 8 and 11 are similar to case 1.

Case 9:
If the function executed case 9, it translated the $K\Box$ rule of $\mathbf{G3m}_*$. Because of the induction hypothesis we know that the top sequent of branch B^t_n with segment-mark m, is the same as the sequent marked m in the top tree-hypersequent of branch B^{Ths}_n. This means that if the rule is translated in the following way:

$$\begin{array}{r} m' : \\ m : \end{array} \dfrac{B_1, B_2, ..., B_l \Rightarrow A}{\Sigma, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow \Phi, \Box A} K\Box$$

$$\rightsquigarrow$$

$$\dfrac{\dfrac{\dfrac{\dfrac{G[\overbrace{\Sigma, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow \Phi}^{m} / \overbrace{B_1, B_2, ..., B_l \Rightarrow A}^{m'}]}{\uparrow} L\Box_l}{G[\Sigma, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow \Phi / B_1, B_2 \Rightarrow A]} L\Box_{...}}{G[\Sigma, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow \Phi / B_1 \Rightarrow A]} L\Box_2$$

$$\dfrac{\dfrac{G[\Sigma, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow \Phi / B_1 \Rightarrow A]}{G[\Sigma, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow \Phi / \Rightarrow A]} L\Box_1}{G[\underbrace{\Sigma, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow \Phi, \Box A}_{m}]} R\Box$$

we know that the conclusions sequent of the $K\Box$ rule and the focused sequent m in the tree hypersequent of the conclusion of the $R\Box$ rule are the same. This makes the premise sequent of the $K\Box$ rule with segment-mark m' and the sequent with mark m' in the premise tree-hypersequent of the rule $L\Box_l$ also the same. This means that in this case, the top sequent of B^t_{n+1} is the same sequent as the sequent in the top tree-hypersequent of B^{Ths}_{n+1} with the same mark.

Case 10 is similar to case 9.

Case 12:

If the function executed case 12, the translation added nothing to $B^T hs_n$. However, we know that backtracking was used in the branch B^{G3m} to a previous segment which is the conclusion of a $K\Box$ or a $4\Box$ rule. Suppose this is sequent S_j of the branch B^t_n with segment-mark m.

Because we can only backtrack to a sequent which is the conclusion of the $K\Box$ or $4\Box$ rule, and because we can only apply the $K\Box$ or $4\Box$ rule immediately after this (see remark 4.5), we know that above the sequent S_j of the branch B^t there is no rule other than $K\Box$ or $4\Box$ which applies to a sequent with segment-mark m.

Therefore, we know that in the translation branch B^{Ths}_{n+1} that after the translation of S_j, there is no rule applied to the sequent marked with m in the tree hypersequents of the branch other than the $L\Box$, $R\Box$ and *trans* rules. These rules do not change the sequent marked with m. Because of this and the induction hypothesis, we know that in B^t_{n+1} there is the sequent m which is exactly the same as the sequent S_j. ■

This lemma secures that it is always possible to apply the translated rule at each step of the function. But it does not show yet that both branches give the same Schütte sets.

Lemma 6.4. *Suppose the translation function $\mathcal{F}$ is used on a branch B^{G3m}, then it is the case that after step i of function $\mathcal{F}$, the branch B^t_i constructs the same Schütte set $\mathfrak{S}_i =< \mathcal{S}, \mathcal{R} >$ using the construction of definition 5.4 as the translation branch B^{Ths}_i using the construction of definition 5.10.*

Proof. We prove this with induction on the step i of the function $\mathcal{F}$.

If $i = 0$:
Both branches in B^t_0 and B^{Ths}_0 only consist of the (tree-hyper)sequent $\Gamma \Rightarrow \Delta$. Therefore it is trivial to check that the set of sequents that are created with the definitions 5.10 and 5.4 are the same.

Induction hypothesis:
If we are at step $i = n+1$, we know that at step $i = n$, that the top the branch B^t_i constructs the same schütte set $\mathfrak{S}_n =< \mathcal{S}, \mathcal{R} >$ as the translation branch B^{Ths}_n.

If $i = n + 1$
We discern cases by the different cases 1-12 of the step $i < 0$ of the translation function $\mathcal{F}$.

Case 1:
The last rule that was applied in B^t_{n+1} is $L\wedge$. It is applied in the following way:

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

where $\Gamma, A \wedge B \Rightarrow \Delta$ is in segment m. We know by the set construction of definition 5.4 that the only change made to the Schütte set $\mathfrak{S}_n$ is to the Schütte sequent in the set that corresponds with the segment m, which now also contains the formulas A and B in its antecedent.

In B^{Ths}_{n+1} we know that the extra rule applied with respect to B^{Ths}_n is $L\wedge$. Which is done in the following way:

$$\frac{G[\Gamma, A, B \Rightarrow \Delta]}{G[\Gamma, A \wedge B \Rightarrow \Delta]}\ L\wedge$$

Where the focused sequent in G is the sequent with the mark m. We know by the set construction of definition 5.10 that the only change made to the Schütte set $\mathfrak{S}_n$ is to the Schütte sequent in the set with the mark m, which now also contains the formulas A and B in its antecedent. This means that the Schütte sets constructed from both the B^t_i and the B^{Ths}_i are the same.

The argumentation is analogous for the cases 2,3,4,5,6,7,8 and 11.

Case 9:
The last rule that was applied in the branch B^t_{n+1} is $K\Box$, it is done in the following way:

$$\frac{B_1, B_2, ..., B_n \Rightarrow A}{\Sigma, \Box B_1, \Box B_2, ..., \Box B_n \Rightarrow \Phi, \Box A}\ K\Box$$

Where the conclusion of the $K\Box$ rule is has the segment-mark m and the premise has the segment-mark m'. This means that by the set construction of definition 5.4 that for $\mathfrak{S}_{n+1}$, $\mathcal{S}_{n+1} = \mathcal{S}_n \cup \{S_{m'}\}$ where $S_{m'}$ is the set version[2] of the sequent $B_1, B_2, ..., B_l \Rightarrow A$. And $\mathcal{R}_{n+1} = \mathcal{R}_n \cup \{(S_m \mathcal{R} S_{m'})\}$.

In B^{Ths}_{n+1} we know that the extra rules applied with respect to B^{Ths}_n are $R\Box$ and l applications of $L\Box$. Which is done in the following way:

$$\cfrac{\cfrac{\cfrac{\cfrac{G[\overbrace{\Sigma, \Box B_1, \Box B_2, ..., \Box B_n \Rightarrow \Phi}^{m} / \overbrace{B_1, B_2, ..., B_l \Rightarrow A}^{m'}]}{\uparrow}\ L\Box_l}{G[\Sigma, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow \Phi / B_1, B_2 \Rightarrow A]}\ L\Box_{...}}{G[\Sigma, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow \Phi / B_1 \Rightarrow A]}\ L\Box_2}{\cfrac{G[\Sigma, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow \Phi / \Rightarrow A]}{G[\underbrace{\Sigma, \Box B_1, \Box B_2, ..., \Box B_l \Rightarrow \Phi, \Box A}_{m}]}\ R\Box}\ L\Box_1$$

We know by the set construction of definition 5.10 that the set $\mathfrak{S}_{n+1}$ of the $\mathbf{Ths}_*$ branch, $\mathcal{S}_{n+1} = \mathcal{S}_n \cup \{S_{m'}\}$ and $\mathcal{R}_{n+1} = \mathcal{R}_n \cup \{(S_m \mathcal{R} S_{m'})\}$ where $S_{m'}$ is the sequent constructed by combining the sequents in the tree-hypersequents in the branch that are marked with m'. This means that the sequent $S_{m'}$ for the constructed set of the $\mathbf{Ths}_*$ branch is also

[2] This is because in the construction of definition 5.4 the sequents are created from sets of formulas

exactly the set version of the sequent $B_1, B_2, ..., B_l \Rightarrow A$. This means that the Schütte sets constructed from both the B_i^t and the B_i^{Ths} are the same.

The argumentation is analogous for case 10.

Case 12:
Nothing changes in both branches and therefore the Schütte sets constructed from both the B_i^t and the B_i^{Ths} are the same. ■

Remark 6.5. If a branch of $\mathbf{G3m}_T$ is translated into $\mathbf{Ths}_T$ or a branch of $\mathbf{G3m}_{S4}$ is translated into $\mathbf{Ths}_{S4}$, the relation $\mathcal{R}$ is reflexive for both Schütte sets because of the constructions of definitions 5.4 and 5.10. This is also true for the transitivity of relation $\mathcal{R}$ in the translations from $\mathbf{G3m}_{K4}$ to $\mathbf{Ths}_{K4}$ and $\mathbf{G3m}_{S4}$ to $\mathbf{Ths}_{S4}$.

Corollary 6.6. *If B^{G3m} is a rule saturated branch in a derivation of* $\mathbf{G3m}_*$ *and produces the Schütte set $\mathfrak{S}$, then the translated branch B^{Ths} also produces the same the Schütte set $\mathfrak{S}$.*

To show the translated branch in the tree-hypersequent calculus can also be used to create a Schütte model, it is important to show that this branch is also saturated for the rules.

Lemma 6.7. *If a branch B^{G3m} of a derivation in* $\mathbf{G3m}_*$ *is rule saturated according to definition 5.1 then the translation B^{Ths} created by using function $\mathcal{F}$ is rule saturated for* $\mathbf{Ths}_*$ *according to definition 5.7.*

Proof. This needs to be proven for each different condition of definition 5.7 separately. This can be done using proof by contradiction. We only prove conditions 2 and 3 because these are the most interesting and the others are similar.

Condition 2:
Suppose B^{Ths} is not saturated for the $L\Box$ rule. This means that there is a formula $\Box A$ in an antecedent of a sequent marked m and a child sequent m' for which the rule $L\Box$ is not applied in the branch B^{Ths}.

Getting a child sequent m' for m in the tree hypersequent of B^{Ths} is only possible if in the translation there is a step of case 9 or 10 where the $K\Box$ (or $4\Box$) is applied in the following way:

$$\begin{array}{r} m': \\ m: \end{array} \frac{B_1, B_2, ..., B_n \Rightarrow C}{\Sigma, \Box B_1, \Box B_2, ..., \Box B_n \Rightarrow \Phi, \Box C} K\Box$$

But because the $L\Box$ rule is not applied with formula $\Box A$ as principal between the sequent m and m', none of the formulas $\Box B_1, \Box B_2, ..., \Box B_n$ can be of the from $\Box A$. Otherwise the rule $L\Box$ *was* applied in the translation. Therefore $\Box A \in \Sigma$ but this is a contradiction and therefore B^{Ths} has to be saturated for the rule $L\Box$ and meets condition 2 of definition 5.7.

Condition 3:
Suppose B^{Ths} is not saturated for the $R\Box$ rule. This means that there is a formula $\Box A$ in a succedent of a sequent marked m for which the rule $A\Box$ is not applied in the branch B^{Ths}.

This means that in the branch B^{G3m} there is a sequent with segment-mark m in which the formula $\Box A$ has not been principal for the rule $K\Box$ or $4\Box$. Otherwise, the rule $R\Box$ *was* applied in the translation with principal formula $\Box A$ in the sequent marked m.

This means that in the branch B^{G3m} there is a sequent with segment-mark m in which the formula $\Box A$ has not been principal in either the rule $K\Box$ or $4\Box$. This, however, contradicts the fact that the branch B^{G3m} is rule saturated according to definition 5.1. Therefore B^{Ths} has to be saturated for the $R\Box$ rule and meets condition 3 of definition 5.7. ■

Corollary 6.8. *If* $\mathfrak{M}$ *is a* **G3m**$_*$ *Schütte model for the sequent* $\Gamma \Rightarrow \Delta$*, then* $\mathfrak{M}$ *is a* **Ths**$_*$ *Schütte model for the equivalent tree hypersequent* $\Gamma \Rightarrow \Delta$*, where* $*$ *is either K, T, K4 or S4.*

6.1.2 Ths to Tls

The transformation of a branch of a derivation in **Ths**$_*$ to a branch in **Tls**$_*$ is done in the same manner as the translation from **G3m**$_*$ to **Ths**$_*$. For this translation, the sequent markers of the sequents in the tree-hypersequents are used to label the formulas in the labeled sequents. The translation is possible by matching the marks to the labels. It is actually possible to translate all tree-hypersequents to tree-labeled sequents and vice versa. Besides this, all the rules of **Ths**$_*$ can easily be translated to rules of **Tls**$_*$. This is shown in [12]. However, we will limit ourselves to translating branches of derivations of the two calculi.

Definition 6.9 (Translation Function **Ths**$_*$ $\rightarrow$ **Tls**$_*$)**.** Here we define the translation function $\mathcal{G}$ from **Ths**$_*$ to **Tls**$_*$. The function uses as input a saturated failed branch of a derivation tree of a tree-hypersequent $G_1 = \Gamma \Rightarrow \Delta$ in **Ths**$_*$ and it returns a derivation branch in **Tls**$_*$ for the tree-hypersequent $S_1 = x : \Gamma \Rightarrow x : \Delta$.

Let $B^{Ths} = G_1, G_2, ..., G_n$ or $B^{Ths} = G_1, G_2, ...$ be such a saturated branch in a derivation in **Ths**$_*$.

The function works in a step by step translation of the branch **Ths**$_*$ starting at G_1 and then translating applications of rules in the branch B^{Ths} in a bottom-up fashion. During the steps of the function, we keep track of the part of B^{Ths} that is translated as B^t, and the translation itself as B^{Tls}. The function stops if it reaches the top of the branch. The function $\mathcal{G}$ starts at step 0.

Step 0**:**
Tree-hypersequent $G_1 = \Gamma \Rightarrow \Delta$ at the root of B^{Ths} is marked with the marker x is translated into the tree-labeled sequent $S_1 = x : \Gamma \Rightarrow x : \Delta$. Create the translation branch $B_0^{Tls} = S_1$. Also create the already translated subbranch $B_0^t = G_1$. Go to step 1.

Step $i > 0$**:**
Suppose tree-hypersequent G_j is the last tree-hypersequent of B^t. What is done in this step depends on which rule is applied backward from tree-hypersequent G_j to G_{j+1} of B^{Ths}. We therefore distinguish different cases for each applied rule.

Case 1. $L\wedge$:
The rule $L\wedge$ rule is used between G_j and G_{j+1} in the following way:

$$\dfrac{G[\overbrace{\Gamma, A, B \Rightarrow \Delta}^{m}]}{G[\underbrace{\Gamma, A \wedge B \Rightarrow \Delta}_{m}]}\ L\wedge$$

This is translated in an application of the $L\wedge$ rule of $\mathbf{Tls}_*$ to the prefixed formula $m : A \wedge B$ in the last sequent of the branch B^{Tls} in the following way:[3]

$$\dfrac{\Phi, m : \Gamma, m : A, m : B \Rightarrow m : \Delta, \Sigma}{\Phi, m : \Gamma, m : A \wedge B \Rightarrow m : \Delta, \Sigma}\ L\wedge$$

We add this bottom-up application of the $L\wedge$ rule to the translation branch B_{i-1}^{Tls} to get the branch B_i^{Tls}. We add G_{j+1} to the already translated subbranch such that $B_i^t = B_{i-1}^t, G_{j+1}$. Then go to step i+1.

In cases 2. $R\wedge$, 3. $L\vee$, 4. $R\vee$, 5. $L\rightarrow$, 6. $R\rightarrow$, 7. $L\neg$ and 8. $R\neg$:
If one of the other propositional rules is used between G_j and G_{j+1}, the translation function works in a similar way as in the case of $L\wedge$.

Case 9. $R\Box$:
The $R\Box$ rule is used between G_j and G_{j+1} in the following way:

$$\dfrac{G[\overbrace{\Gamma \Rightarrow \Delta}^{\text{m}} / \overbrace{\Rightarrow A}^{m'}]}{G[\underbrace{\Gamma \Rightarrow \Delta, \Box A}_{m}]}\ R\Box$$

[3]In the tree-labeled sequents Φ and Σ stand for the formulas that are present in all the other sequents of the tree hypersequent which are not shown in the rule. The presentation of the rules of $\mathbf{Tls}_*$ uses the focus function '[]' to focus on one or two sequents in the tree-hypersequent. Because this is not the case for the tee-labeled sequents we add these formulas (with there proper prefix based on the sequent marks) to the context of the rules in $\mathbf{Tls}_*$ as the multisets Φ and Σ.

This is translated in an application of the $R\Box$ rule of $\mathbf{Tls}_*$ to the prefixed formula $m : \Box A$ in the succedent of the last sequent of the branch B^{Tls} in the following way:

$$\frac{\Phi, m : \Gamma, mRm' \Rightarrow \Sigma, m : \Delta, m' : A}{\Phi, m : \Gamma \Rightarrow \Sigma, m : \Delta, m : \Box A}\ R\Box$$

We add this bottom-up application of the $R\Box$ rule to the translation branch B^{Tls}_{i-1} to get the branch B^{Tls}_i. We add G_{j+1} to the already translated subbranch such that $B^t_i = B^t_{i-1}, G_{j+1}$. Then go to step i+1.

Case 10. $L\Box$:
The $L\Box$ rule is used between G_j and G_{j+1} in the following way:

$$\frac{G[\overbrace{\Gamma, \Box A \Rightarrow \Delta}^{m} / \overbrace{\Psi, A \Rightarrow \Pi}^{m'}]}{G[\underbrace{\Gamma, \Box A \Rightarrow \Delta}_{m} / \underbrace{\Psi \Rightarrow \Pi}_{m'}]}\ L\Box$$

This is translated in an application of the $L\Box$ rule of $\mathbf{Tls}_*$ to the prefixed formula $m : \Box A$ in the antecedent of the last sequent of the branch B^{Tls} in the following way:

$$\frac{\Phi, m : \Gamma, m' : \Psi, mRm', m : \Box A, m' : A \Rightarrow m : \Delta, m' : \Pi, \Sigma}{\Phi, m : \Gamma, m' : \Psi, mRm', m : \Box A \Rightarrow m : \Delta, m' : \Pi, \Sigma}\ L\Box$$

We add this bottom-up application of the $L\Box$ rule to the translation branch B^{Tls}_{i-1} to get the branch B^{Tls}_i. We add G_{j+1} to the already translated subbranch such that $B^t_i = B^t_{i-1}, G_{j+1}$. Then go to step i+1.

Case 11. *trans*:
If the *trans* rule is used between G_j and G_{j+1}, the translation function works in a similar way as in the case of $L\Box$.

Case 12. *ref*:
The *ref* rule is used between G_j and G_{j+1} in the following way:

$$\frac{G[\overbrace{\Gamma, \Box A, A \Rightarrow \Delta}^{m}]}{G[\underbrace{\Gamma, \Box A \Rightarrow \Delta}_{m}]}\ ref$$

This is translated in an application of the *ref* rule of $\mathbf{Tls}_*$ to the prefixed formula $m : \Box A$ in the antecedent of the last sequent of the branch B^{Tls} in the following way:

$$\frac{\Phi, m : \Gamma, m : \Box A, m : A \Rightarrow m : \Delta, \Sigma}{\Phi, m : \Gamma, m : \Box A \Rightarrow m : \Delta, \Sigma}\ ref$$

We add this bottom-up application of the *ref* rule to the translation branch B^{Tls}_{i-1} to get the branch B_i^{Tls}. We add G_{j+1} to the already translated subbranch such that $B_i^t = B_{i-1}^t, G_{j+1}$. Then go to step i+1.

The function stops when G_j is the last sequent in the branch B^{Ths} and the translation of B^{Ths} is B_{i-1}^{Tls}.

Lemma 6.10. *Suppose the translation function $\mathcal{G}$ is used on a branch B^{Ths}, then is it the case that after step i of function $\mathcal{G}$ that for each formula A in each sequent with mark m in the top tree-hypersequent of the branch B_i^t there is a prefixed formula $m : A$ in the top tree-labeled sequent of the branch B_i^{Tls}.*

Proof. This is easily checked with induction on i. ∎

This lemma secures that it is always possible to apply the translated rule at each step of the function. But it does not show yet that both branches give the same Schütte sets.

Lemma 6.11. *Suppose the translation function $\mathcal{G}$ is used on a branch B^{Ths}, then it is the case that after step i of function $\mathcal{G}$, the branch B_i^t constructs the same Schütte set $\mathfrak{S}_i =< \mathcal{S}, \mathcal{R} >$ using the construction of definition 5.10 as the translation branch B_i^{Tls} using the construction of definition 5.14.*

Proof. We prove this with induction on the step i of the function $\mathcal{G}$.

If $i = 0$:
Both branches in B_0^t and B_0^{Tls} only consist of the tree-hypersequent $\Gamma \Rightarrow \Delta$ marked with m and $m : \Gamma \Rightarrow m : \Delta$. Therefore it is trivial to check that the Schütte set of sequents that are created with the definitions 5.10 and 5.14 are the same.

Induction hypothesis:
If we are at step $i = n+1$, we know that at step $i = n$, that the top the branch B_i^t constructs the same schütte set $\mathfrak{S}_n =< \mathcal{S}, \mathcal{R} >$ as the translation branch B_n^{Tls}.

If $i = n + 1$
We discern cases by the different cases 1-12 of the step $i < 0$ of the translation function $\mathcal{G}$.

Case 1:
If the last rule that was applied in B_{n+1}^t is $L\wedge$. It is applied in the following way:

$$\frac{G[\overbrace{\Gamma, A, B \Rightarrow \Delta}^{m}]}{G[\underbrace{\Gamma, A \wedge B \Rightarrow \Delta}_{m}]} L\wedge$$

Where the sequent $\Gamma, A \wedge B \Rightarrow \Delta$ in G is marked with m. We know by the set construction of definition 5.10 that the only change made compared to the Schütte set $\mathfrak{S}_n$ is the Schütte sequent in the set that corresponds with the marker m, which now also contains the formulas A and B in its antecedent.

In B_{n+1}^{Tls} we know that the extra rule applied with respect to B_n^{Tls} is $L\wedge$. Which is done in the following way:

$$\frac{\Phi, m:\Gamma, m:A, m:B \Rightarrow m:\Delta, \Sigma}{\Phi, m:\Gamma, m:A \wedge B \Rightarrow m:\Delta, \Sigma}\ L\wedge$$

Where the prefix m corresponds to the sequent mark m in the B_{n+1}^t branch. We know by the set construction of definition 5.14 that the only change made compared to the Schütte set $\mathfrak{S}_n$ is in the Schütte sequent in the set that corresponds to sequent with the label m, which now also contains the formulas A and B in its antecedent. This means that the Schütte sets constructed from both the B_i^t and the B_i^{Tls} are the same.

The argumentation is analogous to case 1 for the cases 2,3,4,5,6,7,8 and 12.

Case 9:
If the last rule that was applied in B_{n+1}^t is $R\Box$, it is applied in the following way:

$$\frac{G[\overbrace{\Gamma \Rightarrow \Delta}^{m} / \overbrace{\Rightarrow A}^{m'}]}{G[\underbrace{\Gamma \Rightarrow \Delta, \Box A}_{m}]}\ R\Box$$

This means that by the set construction of definition 5.4 that for $\mathfrak{S}_{n+1}$, $\mathcal{S}_{n+1} = \mathcal{S}_n \cup \{S_{m'}\}$ where $S_{m'}$ is the sequent $\Rightarrow A$ and $\mathcal{R}_{n+1} = \mathcal{R}_n \cup \{(S_m \mathcal{R} S_{m'})\}$.

In B_{n+1}^{Tls} we know that the extra rule applied is $R\Box$, which is done in the following way:

$$\frac{\Phi, m:\Gamma, mRm' \Rightarrow \Sigma, m:\Delta, m':A}{\Phi, m:\Gamma \Rightarrow \Sigma, m:\Delta, m:\Box A}\ R\Box$$

We know by the set construction of definition 5.14 that for the the Schütte set $\mathfrak{S}_{n+1}$ of B_{n+1}^{Tls}, $\mathcal{S}_{n+1} = \mathcal{S}_n \cup \{S_{m'}\}$ where $S_{m'}$ is the sequent $\Rightarrow A$ and $\mathcal{R}_{n+1} = \mathcal{R}_n \cup \{(S_m \mathcal{R} Sm')\}$. This means that the Schütte sets constructed from both the B_i^t and the B_i^{Tls} are the same.

Case 10:
If the last rule that was applied in B_{n+1}^t is $L\Box$, it is applied in the following way:

$$\frac{G[\overbrace{\Gamma, \Box A \Rightarrow \Delta}^{m} / \overbrace{\Psi, A \Rightarrow \Pi}^{m'}]}{G[\underbrace{\Gamma, \Box A \Rightarrow \Delta}_{m} / \underbrace{\Psi \Rightarrow \Pi}_{m'}]}\ L\Box$$

We know by the set construction of definition 5.10 that the only change made compared to the Schütte set $\mathfrak{S}_n$ is the Schütte sequent in the set that corresponds with the marker m', which now also contains the formula A in its antecedent.

In B^{Tls}_{n+1} we know that the extra rule applied is $L\Box$. Which is done in the following way:

$$\frac{\Phi, m : \Gamma, m' : \Psi, mRm', m : \Box A, m' : A \Rightarrow m : \Delta, m' : \Pi, \Sigma}{\Phi, m : \Gamma, m' : \Psi, mRm', m : \Box A \Rightarrow m : \Delta, m' : \Pi, \Sigma} \; L\Box$$

Where the labels m and m' corresponds to the sequent labels m and m' in B^t_{n+1}. We know by the set construction of definition 5.14 that the only change made compared to the Schütte set $\mathfrak{S}_n$ is the Schütte sequent in the set that corresponds to sequent with the label m', which now also contains the formula A in its antecedent. This means that the Schütte sets constructed from both the B^t_i and the B^{Tls}_i are the same.

The argumentation for case 11 is analogous to case 10. ■

Remark 6.12. If a branch of $\mathbf{Ths}_T$ is translated into $\mathbf{Tls}_T$ or a branch of $\mathbf{Ths}_{S4}$ is translated into $\mathbf{Tls}_{S4}$, the relation $\mathcal{R}$ is reflexive for both Schütte sets because of the constructions of definitions 5.10 and 5.14. This is also true for the transitivity of relation $\mathcal{R}$ in the translations from $\mathbf{Ths}_{K4}$ to $\mathbf{Tls}_{K4}$ and $\mathbf{Ths}_{S4}$ to $\mathbf{Tls}_{S4}$.

Corollary 6.13. *If B^{Ths} is a rule saturated branch in a derivation of* $\mathbf{Ths}_*$ *and produces the Schütte set $\mathfrak{S}$, the translated branch B^{Tls} also produces the same the Schütte set $\mathfrak{S}$.*

Lemma 6.14. *If a branch B^{Ths} of a derivation in* $\mathbf{Ths}_*$ *is rule saturated according to definition 5.7 then the translation B^{Tls} created by using function $\mathcal{G}$ is rule saturated for* $\mathbf{Tls}_*$ *according to definition 5.13.*

Proof. This needs to be proven for each different condition of definition 5.13 separately. This can be done using proof by contradiction. We only prove condition 1, the others are similar.

Condition 1:
Suppose B^{Tls} is not saturated for the propositional rule $L\wedge$. This means that there is a prefixed formula $m : A \wedge B$ in an antecedent of a sequent for which the rule $L\wedge$ is not applied in the branch B^{Tls}.

This means that in the branch B^{Ths} there is a sequent in a tree-hypersequent with mark m in which the formula $A \wedge B$ has not been principal for the rule $L\wedge$. Otherwise the rule $L\wedge$ of $\mathbf{Tls}_*$ *was* applied in the translation B^{Tls} with principal prefixed formula $m : A \wedge B$.

This means that in the branch B^{Ths} there is a formula $A \wedge B$ in a sequent with mark m in a tree-hypersequent which is never principal in any $L\wedge$ rule. This, however, contradicts the fact that the branch B^{Ths} is rule saturated according to definition 5.7. Therefore B^{Ths} has to be saturated for the $R\Box$ rule and meets condition 1 of definition 5.13. ■

Corollary 6.15. *If* $\mathfrak{M}$ *is a* $\mathbf{Ths}_*$ *Schütte model for the tree-hypersequent* $\Gamma \Rightarrow \Delta$*, then* $\mathfrak{M}$ *is a* $\mathbf{Tls}_*$ *Schütte model for the equivalent tree-labeled sequent* $x : \Gamma \Rightarrow x : \Delta$*, where* $*$ *is either K, T, K4 or S4.*

6.1.3 Tls to G3m

Translating a branch of a derivation in $\mathbf{Tls}_*$ into a branch in $\mathbf{G3m}_*$ is not that straightforward. To carry out a proper translation we need some lemmas about the possibility of permuting the order of the application of rules in a derivation tree. We first have to reorder the branch in $\mathbf{Tls}_*$ appropriately so that we can create segments in the $\mathbf{G3m}_*$ translation which consist of rules that all have principal formulas with the same labels. This has to be done, because the $K\Box$, and $4\Box$ rules of $\mathbf{G3m}_*$ cannot permute over the propositional rules, while this is possible for the modal rules of $\mathbf{Tls}_*$ in some cases.

6.1.3.1 Permutation lemma's

Lemma 6.16. *In a branch of* $\mathbf{Tls}_*$*, instances of the* $L\Box$ *and trans rules permute down with respect to the rules* $L\neg$*,* $R\neg$*,* $L\wedge$ *,* $R\wedge$*,* $L\vee$*,* $R\vee$*,* $L\rightarrow$*,* $R\rightarrow$ *and ref if the first active prefix in the* $L\Box$ *or trans rule is not the same as the active prefix in the rule which is permuted up.*

Proof. There are two cases for the relation between the prefixes active in the rules. We will show that the lemma holds for the two cases.

In the case the prefixes active in the two rules are different the permutation is straightforward. For example $L\Box$ permuting with $L\wedge$:

$$\dfrac{\dfrac{x : \Box A, y : A, z : B, z : C, xRy, \Gamma \Rightarrow \Delta}{x : \Box A, z : B, z : C, xRy, \Gamma \Rightarrow \Delta}\,L\Box}{x : \Box A, z : B \wedge C, xRy, \Gamma \Rightarrow \Delta}\,L\wedge$$

$$\downarrow$$

$$\dfrac{\dfrac{x : \Box A, y : A, z : B, z : C, xRy, \Gamma \Rightarrow \Delta}{x : \Box A, y : A, z : B \wedge C, xRy, \Gamma \Rightarrow \Delta}\,L\wedge}{x : \Box A, z : B \wedge C, xRy, \Gamma \Rightarrow \Delta}\,L\Box$$

It is also possible in the case that the active prefix in the upwards permuted is the same as the second active prefix of the downwards permuted rule. For example of the active prefix of the upward permuted $L\wedge$ rule is y and xRy is the active relational atom in the permuted $L\Box$:

$$\cfrac{\cfrac{x:\Box A, y:A, y:B, y:C, xRy, \Gamma \Rightarrow \Delta}{x:\Box A, y:B, y:C, xRy, \Gamma \Rightarrow \Delta}\, L\Box}{x:\Box A, y:B\wedge C, xRy, \Gamma \Rightarrow \Delta}\, L\wedge$$

$$\downarrow$$

$$\cfrac{\cfrac{x:\Box A, y:A, y:B, y:C, xRy, \Gamma \Rightarrow \Delta}{x:\Box A, y:A, y:B\wedge C, xRy, \Gamma \Rightarrow \Delta}\, L\wedge}{x:\Box A, y:B\wedge C, xRy, \Gamma \Rightarrow \Delta}\, L\Box$$

■

Lemma 6.17. *In a branch of* $\mathbf{Tls}_*$ *the rules* $L\Box$ *and trans permute down with respect to rules trans and* $L\Box$ *if the first active prefix in the permuted down rule is not the second active prefix in the permuted up* $L\Box$ *or trans rule.*

Proof. There are three cases for the relation between the prefixes active in the rules. We will show that the lemma holds for all the three cases.

1. The two rules have different prefixes in their active relational atoms:
The permutation is trivial.

2. The two rules have the same first prefix in their active relational atoms:
We show as example the permutation between two $L\Box$ rules:

$$\cfrac{\cfrac{x:\Box A, x:\Box B, y:A, z:B, xRy, xRz, \Gamma \Rightarrow \Delta}{x:\Box A, x:\Box B, y:A, xRy, xRz, \Gamma \Rightarrow \Delta}\, L\Box}{x:\Box A, x:\Box B, xRy, xRz, \Gamma \Rightarrow \Delta}\, L\Box$$

$$\downarrow$$

$$\cfrac{\cfrac{x:\Box A, x:\Box B, y:A, z:B, xRy, xRz, \Gamma \Rightarrow \Delta}{x:\Box A, x:\Box B, z:B, xRy, xRz, \Gamma \Rightarrow \Delta}\, L\Box}{x:\Box A, x:\Box B, xRy, xRz, \Gamma \Rightarrow \Delta}\, L\Box$$

3. The permuted down rule has as second active prefix the the first active prefix of the the permuted up rule:

We show as example the permutation between two $L\Box$ rules:

$$\dfrac{\dfrac{x:\Box A, y:\Box B, y:A, z:B, xRy, yRz, \Gamma \Rightarrow \Delta}{x:\Box A, y:\Box B, z:B, xRy, yRz, \Gamma \Rightarrow \Delta}\ L\Box}{x:\Box A, y:\Box B, xRy, yRz, \Gamma \Rightarrow \Delta}\ L\Box$$

$$\downarrow$$

$$\dfrac{\dfrac{x:\Box A, y:\Box B, y:A, z:B, xRy, yRz, \Gamma \Rightarrow \Delta}{x:\Box A, y:\Box B, y:A, xRy, yRz, \Gamma \Rightarrow \Delta}\ L\Box}{x:\Box A, y:\Box B, xRy, yRz, \Gamma \Rightarrow \Delta}\ L\Box$$

■

Lemma 6.18. *In a branch of* **Tls**$_*$*, the rules* $L\Box$ *and trans permute down with respect to rule* $R\Box$ *when the relational atom created in the* $R\Box$ *is not active in these rules.*

Proof. There are two cases for the relation between the prefixes active in the rules. We will show that the lemma holds for the two cases.

1. The two rules have totally different labels in their active relational atoms:
The permutation is trivial.

2. The two rules have the same first active prefix in their active relational atoms:
We show as example a permutation with an $L\Box$ rule.

$$\dfrac{\dfrac{x:\Box A, y:A, xRy, xRz, \Gamma \Rightarrow \Delta, z:B}{x:\Box A, xRy, xRz, \Gamma \Rightarrow \Delta, z:B}\ L\Box}{x:\Box A, xRy, \Gamma \Rightarrow \Delta, x:\Box B}\ R\Box$$

$$\downarrow$$

$$\dfrac{\dfrac{x:\Box A, y:A, xRy, xRz, \Gamma \Rightarrow \Delta, z:B}{x:\Box A, y:A, xRy, \Gamma \Rightarrow \Delta, x:\Box B}\ R\Box}{x:\Box A, xRy, \Gamma \Rightarrow \Delta, x:\Box B}\ L\Box$$

■

Lemma 6.19. *In a branch of* **Tls**$_*$*, the rule* $R\Box$ *permutes down with respect to the rules* $L\neg$*,* $R\neg$*,* $L\wedge$ *,* $R\wedge$*,* $L\vee$*,* $R\vee$*,* $L\rightarrow$*,* $R\rightarrow$ *and ref if the first active prefix in the* $R\Box$ *rule is not the same as the active prefix in the upward permuted rule.*

Proof. Similar to the proof of lemma 6.16 ■

Lemma 6.20. *In a branch of* **Tls**$_*$*, the rule* $R\Box$ *permutes down with respect to the rules* $R\Box$*,* $L\Box$ *and trans if the first active prefix in the permuted down* $R\Box$ *rule is not the second active prefix in the upwards permuted* $R\Box$*,* $L\Box$ *or trans rule.*

Proof. The only cases for relations between the prefixed of the rule we need to check are when the prefixes are completely different or whenever the two active relational atoms in the rules have the same first label.

1. The prefixes are completely different:
Trivial.

2. The two active relational atoms in the rules have the same first label: This is the reverse permutation of the permutation seen in the proof of case 2 in lemma 6.18. ■

Lemma 6.21. *In derivations, Rules $L\neg$, $R\neg$, $L\wedge$, $R\wedge$, $L\vee$, $R\vee$, $L\rightarrow$, $R\rightarrow$ and ref permute down with respect to rules $L\neg$, $R\neg$, $L\wedge$, $R\wedge$, $L\vee$, $R\vee$, $L\rightarrow$, $R\rightarrow$ and ref in the case the active prefix in the rule is not the same as the active prefix in the rule permuted with.*

Proof. trivial ■

Lemma 6.22. *In derivations, Rules $L\neg$, $R\neg$, $L\wedge$, $R\wedge$, $L\vee$, $R\vee$, $L\rightarrow$, $R\rightarrow$ and ref permute down with respect to rules $R\Box$, $L\Box$ and trans whenever the active prefix in the rule is not the second active prefix in the permuted trans, $R\Box$ or $L\Box$ rule.*

Proof. The permutation is straightforward. For example the rule $L\wedge$ permuting with $L\Box$:

$$\dfrac{\dfrac{x:\Box A, y:A, z:B, z:C, xRy, \Gamma \Rightarrow \Delta}{x:\Box A, y:A, z:B\wedge C, xRy, \Gamma \Rightarrow \Delta}\,L\wedge}{x:\Box A, z:B\wedge C, xRy, \Gamma \Rightarrow \Delta}\,L\Box$$

$$\downarrow$$

$$\dfrac{\dfrac{x:\Box A, y:A, z:B, z:C, xRy, \Gamma \Rightarrow \Delta}{x:\Box A, z:B, z:C, xRy, \Gamma \Rightarrow \Delta}\,L\Box}{x:\Box A, z:B\wedge C, xRy, \Gamma \Rightarrow \Delta}\,L\wedge$$

■

6.1.3.2 Reordering of tree-labeled sequent branches and translation

With all these permutation lemmas it is possible to rearrange a derivation branch of $\mathbf{Tls}^*$ in such a way that the application of the rules follows a specific order. An order in which all the rules with principal formulas with the same prefixes are grouped together starting with modal rules. Table 6.1 gives a summary of all the permutation lemmas proven in the previous section.

Lemma 6.23. *Suppose we have a failed branch of a derivation in $\mathbf{Tls}_*$. It is possible to reorder the branch in such a way that all rule applications which have the same active*

Lemma	**Permuting (down)**	**Permuted (up)**	**Condition**
6.16	$L\Box$ or *Trans*	Propositional rules and *ref*	The first active prefix in the permuting rule is not the active prefix in the permuted rule.
6.17	$L\Box$ or *Trans*	$L\Box$ or *Trans*	First active prefix in the permuting rule is not the second active prefix of the permuted rule.
6.18	$L\Box$ or *Trans*	$R\Box$	Relational atom created in permuted is not active in the permuting rule.
6.19	$R\Box$	Propositional rules and *ref*	The first active prefix in the permuting rule is not the active prefix in the permuted rule.
6.20	$R\Box$	$R\Box$, $L\Box$ and *trans*	The first active prefix of the permuting rule is not the second active prefix of the permuted rule
6.21	Propositional rules and *ref*	Propositional rules and *ref*	The active prefixes of both rules are not the same
6.22	Propositional rules and *ref*	$R\Box$, $L\Box$ and *trans*	The active prefix in the permuting rule is not the second active prefix in the permuted rule.

TABLE 6.1: Summary of the permutation lemma's

prefix or active second prefix in the case of modal rules are grouped together in the following order seen from the root to the leaf: (1) R□ (2) trans (3) L□ (4) propositional rules and ref.

Proof. This is done with the induction on the height h of the tree described by the relational atoms in the union of antecedents in the branch.

If $h = 1$**:**
If $h = 1$ there is only one prefix in the branch, otherwise it would not be a tree. Therefore, all used rules can only be propositional rules and *ref* rules with the same label, and thus there is no reordering needed.

Induction hypothesis: A branch with height $h = n$ is reorderable.

If h = n+1:
Suppose the labels at height n+1 of the relation tree are the labels l_1,l_2,..,l_m with $l_i, l_j \in \{l_1, l_2, .., l_m\}$. All the propositional rules or the *ref* rule with active prefixes which are not l_1,l_2,..,l_m can be permuted down with all the rules which have one of the l_1,l_2,..,l_m prefixes as their active prefix or active second prefix. This is evident from lemma 6.21 and lemma 6.22.

Besides this the rules $R\Box$, $L\Box$ and *trans* which do not have any of the labels l_1,l_2,..,l_m as their second active prefix can be permuted down with all the rules which have one of the l_1,l_2,..,l_m prefixes as their active prefix or active second prefix. This is because of lemma's 6.19, 6.20, 6.16, 6.17 and 6.18 plus the fact that the permuted down rules cannot have any of the labels l_1,l_2,..,l_m as their first active prefix as these labels are the leafs of the relation tree.

This means that the branch can be split into a lower part with only labels which are of height $h \leq n$ and an upper part which only contains formulas with the labels l_1,l_2,..,l_m as their active or second active prefix. Because of the induction hypothesis, this lower part can be reordered according to the to be proven lemma. Which means we only still have to prove that the upper part can be ordered too.

Because all these labels l_1,l_2,..,l_m are at height n+1 of the relation tree they cannot be related to each other, otherwise, the sequents in the branch would not be tree-labeled sequents. Therefore we know that for none of the rules $L\Box$,$R\Box$ or *trans* a relational atom of the form l_iRl_j is active.

We also know that all the propositional or *ref* rules with a label l_i as their active prefix or $L\Box$ and *trans* rules with l_i as their second prefix have to be applied after the application of the $R\Box$ rule which introduced the label l_i.

For every label, l_i, the $L\Box$ and *trans* rules which have l_i as their second prefix can be permuted downward such that they are applied right after the $R\Box$ rule which introduced the label l_i (and has it as it secondary prefix). For this to happen the $L\Box$ and *trans* rules have to be permuted down with propositional or *ref* rules which have one of the labels l_1,l_2,..,l_m as their active prefix, or $L\Box$, $R\Box$ and *trans* rules which do not have l_i as their first prefix.

These permutations are possible because of lemma 6.16, lemma 6.17 and lemma 6.18.

For every label, l_i, the propositional and *ref* rules which have the label l_i as their active prefix can be permuted downward such that they are applied after the $R\Box$, $L\Box$ and *trans* rules which have l_i as their second prefix. For this to happen the propositional and *ref* rules have to be permuted down with propositional or *ref* rules which have one of the labels l_1,l_2,..,l_m but not l_i as their active prefix, or $L\Box$, $R\Box$ and *trans* rules which do not have l_i as either their first or second prefix.

These permutations are possible because of lemma 6.21 and lemma 6.22.

If these permutations are done for each label $l_i \in \{l_1, l_2, .., l_m\}$ the second part of the branch is ordered which concludes the proof. ■

Definition 6.24 (Translation Function $\mathbf{Tls}_* \rightarrow \mathbf{G3m}_*$)**.** Here we define the translation function $\mathcal{H}$ from $\mathbf{Tls}_*$ to $\mathbf{G3m}_*$. The function uses as input a saturated failed branch of a derivation tree of a tree-labeled sequent $S_1 = x : \Gamma \Rightarrow x : \Delta$ which is *reordered* according to lemma 6.23. It returns a derivation branch in $\mathbf{G3m}_*$ for the sequent $T_1 = \Gamma \Rightarrow \Delta$.

Let $B^{Tls} = S_1, S_2, ..., S_n$ or $B^{Tls} = S_1, S_2, ...$ be such a branch.

The function $\mathcal{H}$ works in a step by step translation of the branch $\mathbf{Tls}_*$ starting at S_1 and then translating applications of rules in the branch B^{Tls} in a bottom-up fashion. During the steps of the function, we keep track of the part of B^{Tls} that is translated as B^t, and the translation itself as B^{G3m}. The function stops if it reaches the top of the branch. The function $\mathcal{H}$ starts at step 0.

Step 0**:**
tree-labeled sequent $S_1 = x : \Gamma \Rightarrow x : \Delta$ at the root of B^{Tls} is translated into the sequent $T_1 = \Gamma \Rightarrow \Delta$. Create the translation branch $B_0^{G3m} = T_1$. Also create the already translated subbranch $B_0^t = S_1$. Go to step 1.

Step $i > 0$**:**
Suppose tree-labeled sequent S_j is the last tree-labeled sequent of B^t. What is done in this step depends on which rule is applied backward from S_j to S_{j+1} in B^{Tls}. We therefore distinguish different cases based on the applied rule(s).

Case 1. $L\wedge$:
The rule $L\wedge$ rule is used between S_j and S_{j+1} in the following way:

$$\frac{\Phi, x : \Gamma, x : A, x : B \Rightarrow x : \Delta, \Sigma}{\Phi, x : \Gamma, x : A \wedge B \Rightarrow x : \Delta, \Sigma}\ L\wedge$$

This is translated in an application of the $L\wedge$ rule of $\mathbf{G3m}_*$ to the last sequent of the branch B^{G3m}, which has the segmentmark x because of the reordering, in the following way:

$$\begin{array}{ll} x: & \dfrac{\Gamma, A, B \Rightarrow \Delta}{\Gamma, A \wedge B \Rightarrow \Delta}\ L\wedge \\ x: & \end{array}$$

We add this application of the $L\wedge$ rule to the translation branch B_{i-1}^{G3m} to get the branch B_i^{G3m}. We add S_{j+1} to the already translated subbranch such that $B_i^t = B_{i-1}^t, S_{j+1}$. Then go to step i+1.

In cases 2. $R\wedge$, 3. $L\vee$, 4. $R\vee$, 5. $L\rightarrow$, 6. $R\rightarrow$, 7. $L\neg$ and 8. $R\neg$:
If one of the other propositional rules is used between S_j and S_{j+1}, the translation function works in a similar way as in the case of $L\wedge$.

Case 9. $R\Box$ and B^{Tls} is a branch in a derivation of $\mathbf{Tls}_K$ or $\mathbf{Tls}_T$:
Because the branch B^{Tls} is ordered we know that if the $R\Box$ is used between S_j and S_{j+1} in B^{Tls}. We know that the labeled sequents $S_{j+1}, ..., S_{j+l+1}$ are seperated by l applications of the $L\Box$ rule with the same active relational atom xRy that is active in the $R\Box$ rule. Besides that we know that the branch B^{Tls} is saturated. This means

according to condition 2 of definition 5.13 that for every different formula $x : \Box B$ the rule $L\Box$ is used with relational atom xRy. Therefore we can assume the following applications of rules between the tree-labeled sequents $S_j, S_{j+1}, ..., S_{j+l+1}$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Psi, x : \Sigma, xRy, x : \Box\Gamma, y : B_1, y : B_2, ..., y : B_l \Rightarrow \Pi, x : \Phi, y : A}{\uparrow}L\Box_l}{\Psi, x : \Sigma, xRy, x : \Box\Gamma, y : B_1, y : B_2 \Rightarrow \Pi, x : \Phi, y : A}L\Box_{...}}{\Psi, x : \Sigma, xRy, x : \Box\Gamma, y : B_1 \Rightarrow \Pi, x : \Phi, y : A}L\Box_2}{\Psi, x : \Sigma, xRy, x : \Box\Gamma \Rightarrow \Pi, x : \Phi, y : A}L\Box_1}{\Psi, x : \Sigma, x : \Box\Gamma \Rightarrow \Pi, x : \Phi, x : \Box A}R\Box$$

Where $x : \Box\Gamma$ is the multiset of all labeled formulas of the form $x : \Box B$ in the antecedent of these labeled sequents. Because of the saturation we know that $x : \Box B \in x : \Box\Gamma$ if and only if $y : B \in \{y : B_1, y : B_2, ..., y : B_l\}$. The two multisets only differ in duplicates. This is translated in an application of backtracking to the sequent $\Sigma, \Box B_1, \Box B_2, ..., \Box B_m \Rightarrow \Phi, \Box A$, which is at the end of segment $x$and the $K\Box$ rule of $\mathbf{G3m}_*$ to the last sequent S' of the branch B^{G3m} in the following way:

$$\begin{array}{l} y: \\ x: \\ z: \end{array} \quad \dfrac{\dfrac{B_1, B_2, ..., B_m \Rightarrow A}{\Sigma, \Box B_1, \Box B_2, ..., \Box B_m \Rightarrow \Phi, \Box A}K\Box}{S'}\mathit{backtrack}$$

It is possible that the rule $L\Box$ is used more than once for one prefixed formula $x : \Box B$, or that $x : \Box B$ occurs two times in the antecedents of $S_{j+1}, ..., S_{j+l+1}$ and the rule $L\Box$ is only used once. This means that the $K\Box$ rule sometimes needs to be helped by extra height preserving contractions or height preserving weakening after the application of $K\Box$. Two examples of this are the following two translations:

$$\dfrac{\dfrac{\dfrac{x : \Sigma, xRy, x : \Box B, y : B, y : B \Rightarrow x : \Phi, y : A}{x : \Sigma, xRy, x : \Box B, y : B \Rightarrow x : \Phi, y : A}L\Box}{x : \Sigma, xRy, x : \Box B \Rightarrow x : \Phi, y : A}L\Box}{x : \Sigma, x : \Box B \Rightarrow x : \Phi, x : \Box A}R\Box$$

$$\rightsquigarrow$$

$$\dfrac{\dfrac{B, B \Rightarrow A}{B \Rightarrow A}\mathit{contraction}}{\Sigma, \Box B \Rightarrow \Phi, \Box A}K\Box$$

$$\dfrac{\dfrac{x : \Sigma, xRy, x : \Box B, x : \Box B, y : B \Rightarrow x : \Phi, y : A}{x : \Sigma, xRy, x : \Box B, x : \Box B \Rightarrow x : \Phi, y : A}L\Box}{x : \Sigma, x : \Box B, x : \Box B \Rightarrow x : \Phi, x : \Box A}R\Box$$

$$\rightsquigarrow$$

$$\dfrac{\dfrac{B \Rightarrow A}{B, B \Rightarrow A}\mathit{weakening}}{\Sigma, \Box B, \Box B \Rightarrow \Phi, \Box A}K\Box$$

We add these applications of backtracking the $K\Box$ rule and possible extra contractions and weakenings to the translation branch B_{i-1}^{G3m} to get the branch B_i^{G3m}. We add *all* the labeled sequents $S_{j+1}, ..., S_{j+l+1}$ to the already translated subbranch such that $B_i^t = B_{i-1}^t, S_{j+1}, ..., S_{j+l+1}$. Then go to step i+1.

Case 10. $R\Box$ and B^{Tls} is a branch in a derivation of $\mathbf{Tls}_{K4}$ or $\mathbf{Tls}_{S4}$:
These can be translated into backtracking, the $4\Box$ rule and extra weakenings and contractions in a similar way as in case 9 with the addition that we treat the extra *trans* rules the same way as the $L\Box$ rules.

Case 11. *ref*:
The rule *ref* rule is used between S_j and S_{j+1} in the following way:

$$\frac{\Phi, x : \Gamma, x : \Box A, x : A \Rightarrow \Sigma, x : \Delta}{\Phi, x : \Gamma, x : \Box A \Rightarrow \Sigma, x : \Delta}\ ref$$

This is translated in an application of the $T\Box$ rule of $\mathbf{G3m}_*$ to the last sequent of the branch B^{G3m}, which has the segmentmark x because of the reordering, in the following way:

$$\begin{array}{l} x\text{:} \\ x\text{:} \end{array} \quad \frac{\Gamma, \Box A, A \Rightarrow \Sigma, \Delta}{\Gamma, \Box A \Rightarrow \Delta}\ T\Box$$

We add this application of the $T\Box$ rule to the translation branch B_{i-1}^{G3m} to get the branch B_i^{G3m}. We add S_{j+1} to the already translated subbranch such that $B_i^t = B_{i-1}^t, S_{j+1}$. Then go to step i+1.

Lemma 6.25. *Suppose the translation function $\mathcal{H}$ is used on a saturated reordered branch B^{Tls}, then it is the case that after step i of function $\mathcal{H}$ that for each formula $x : A$ in the top tree-labeled sequent of the branch B_i^t where x is the active or second active prefix of the last rule in B_i^t, there is a formula A in the top tree-labeled sequent of the branch B_i^{G3m}.*

Proof. This is checked with induction on i. ■

This lemma secures that after each step, the next step of the translation can also be executed if we are translating any propositional or the *ref* rules. This is because these are sorted on the active prefix. Besides that, the $R\Box$ rule and the $L\Box$ and *trans* rules can also be translated into the $K\Box$ or $4\Box$ rule, because we can use backtracking to go to the last sequent with the appropriate label segment label. Next, we want to show that both branches give the same Schütte sets.

Lemma 6.26. *Suppose the translation function $\mathcal{H}$ is used on a branch B^{Tls}, then it is the case that after step i of function $\mathcal{H}$, the branch B_i^t constructs the same Schütte set $\mathfrak{S}_i =< \mathcal{S}, \mathcal{R} >$ using the construction of definition 5.14 as the translation branch B_i^{G3m} using the construction of definition 5.4.*

Proof. We prove this with induction on the step i of the function $\mathcal{H}$.

If $i = 0$**:**
Both branches in B_0^t and B_0^{Tls} only consist of the tree-labeled sequent $x : \Gamma \Rightarrow x : \Delta$ and the sequent $\Gamma \Rightarrow \Delta$ marked with x. Therefore it is trivial to check that the Schütte set of sequents that are created with the definitions 5.14 and 5.4 are the same.

Induction hypothesis:
If we are at step $i = n + 1$, we know that at step $i = n$, that the top the branch B_i^t constructs the same schütte set $\mathfrak{S}_n =< \mathcal{S}, \mathcal{R} >$ as the translation branch B_n^{G3m}.

If $i = n + 1$
We discern cases by the different cases 1-11 of the step $i < 0$ of the translation function $\mathcal{H}$.

Case 1:
If the last rule that was applied in B_{n+1}^t is $L\wedge$, it is applied in the following way:

$$\frac{\Phi, x : \Gamma, x : A, x : B \Rightarrow \Sigma, x : \Delta}{\Phi, x : \Gamma, x : A \wedge B \Rightarrow \Sigma, x : \Delta}\ L\wedge$$

We know by the set construction of definition 5.10 that the only change made compared to the set $\mathfrak{S}_n =< \mathcal{S}, \mathcal{R} >_n$ is the sequent in the set that corresponds with the label x, which now also contains the formulas A and B in its antecedent.

In B_{n+1}^{G3m} we know that the extra rule applied is $L\wedge$. Which is done in the following way in the segment with mark x:

$$\begin{array}{ll} x: & \dfrac{\Gamma, A, B \Rightarrow \Delta}{\Gamma, A \wedge B \Rightarrow \Delta}\ L\wedge \\ x: & \end{array}$$

We know by the set construction of definition 5.14 that the only change made compared to the Schütte set $\mathfrak{S}_n =< \mathcal{S}, \mathcal{R} >_n$ is the sequent in the set that corresponds to sequent with the label m, which now also contains the formulas A and B in its antecedent. This means that the Schütte sets constructed from both the B_i^t and the B_i^{G3m} are the same.

The argumentation is analogous to case 1 for the cases 2,3,4,5,6,7,8 and 11.

Case 9:
If an $R\Box$ rule is applied in the in step n+1 in B_{n+1}^t with the relational atom xRy, it also includes all the $L\Box$ rules that occur in the branch with the same active relational atom xRy. This is evident from the ordering of lemma 6.23. This means that the application

of the $R\Box$ and the m $L\Box$ rules in the **Tls** branch looks like this:

$$\cfrac{\cfrac{\cfrac{\cfrac{\Sigma, xRy, x:\Box\Gamma, y:B_1, y:B_2, ..., y:B_m \Rightarrow \Phi, y:A}{\uparrow}\,L\Box_m}{\Sigma, xRy, x:\Box\Gamma, y:B_1, y:B_2 \Rightarrow \Phi, y:A}\,L\Box_{...}}{\Sigma, xRy, x:\Box\Gamma, y:B_1 \Rightarrow \Phi, y:A}\,L\Box_2}{\cfrac{\Sigma, xRy, x:\Box\Gamma \Rightarrow \Phi, y:A}{\Sigma, x:\Box\Gamma \Rightarrow \Phi, x:\Box A}\,R\Box}\,L\Box_1$$

where $x:\Box\Gamma$ is the multiset of all formulas of the form $x:\Box B$ in the antecedent of the sequents. We know that $x:\Box B \in x:\Box\Gamma$ if and only if $y:B \in \{y:B_1, y:B_2, ..., y:B_m\}$ because the total branch B^{Tls} is saturated.

This means that by the set construction of definition 5.14 that for $\mathfrak{S}_{n+1}$, $\mathcal{S}_{n+1} = \mathcal{S}_n \cup \{S_y\}$ where S_y is the set version of the sequent $B_1, B_2, ..., B_m \Rightarrow A$. And $\mathcal{R}_{n+1} = \mathcal{R}_n \cup \{S_x \mathcal{R} S_y\}$.

In B^{G3m}_{n+1} we know that the extra rule applied in step n+1 are backtracking and the $K\Box$ rule. This is done in the following way:

$$\begin{array}{ll} y: & \cfrac{\cfrac{B_1, B_2, ..., B_o \Rightarrow A}{\Sigma, \Box B_1, \Box B_2, ..., \Box B_o \Rightarrow \Phi, \Box A}\,K\Box}{S'}\,backtrack \\ x: & \\ z: & \end{array}$$

Were $x:\Box\Gamma = \{x:\Box B_1, x:\Box B_2, ..., x:\Box B_o\}$ and the conclusion of the $K\Box$ rule is the end of the segment x, and the premise is the first sequent in the segment y. Besides this it is possible that after the application of the $K\Box$ rule some implicit contraction is used for double applications of the $L\Box$ rule for the same formula $x:\Box B$ in B^t_{n+1}, or implicit weakening for formula of the from $x:\Box C$ for which the rule $L\Box$ is not used. However, because we know the branch is saturated, there is no formula $x:\Box C \in \{x:\Box B_1, x:\Box B_2, ..., x:\Box B_o\}$ which is different from all formulas $x:\Box B_1, x:\Box B_2, ..., x:\Box B_m$ for which the $L\Box$ rule is used. This means that the set version of $B_1, B_2, ..., B_o \Rightarrow A$ is the same as the set version of $B_1, B_2, ..., B_m \Rightarrow A$.

This means that by the set construction of definition 5.4 that for $\mathfrak{S}_{n+1}$, $\mathcal{S}_{n+1} = \mathcal{S}_n \cup \{S_y\}$ where S_y is the set version of the sequent $B_1, B_2, ..., B_n \Rightarrow A$ and $\mathcal{R}_{n+1} = \mathcal{R}_n \cup \{S_x \mathcal{R} S_y\}$. This means that the Schütte sets constructed from both the B^t_i and the B^{G3m}_i are the same.

Case 10 is proven analogous to case 9 where we deal with the translations of *trans* rules in the same was as with the $L\Box$ rules. ■

Remark 6.27. If a branch of $\mathbf{Tls}_T$ is translated into $\mathbf{G3m}_T$ or a branch of $\mathbf{Tls}_{S4}$ is translated into $\mathbf{G3m}_{S4}$, the relation $\mathcal{R}$ is reflexive for both Schütte sets because of the constructions of definitions 5.14 and 5.4. This is also true for the transitivity of relation $\mathcal{R}$ in the translations from $\mathbf{Tls}_{K4}$ to $\mathbf{G3m}_{K4}$ and $\mathbf{Tls}_{S4}$ to $\mathbf{G3m}_{S4}$.

Corollary 6.28. *If B^{Tls} is a reordered rule saturated branch in a derivation of* $\mathbf{Ths}_*$ *and produces the Schütte set $\mathfrak{S}$, the translated branch B^{G3m} also produces the same the Schütte set $\mathfrak{S}$.*

Lemma 6.29. *If a branch B^{Tls} of a derivation in* $\mathbf{Tls}_*$ *is rule saturated according to definition 5.13 then the translation B^{G3m} created by using function $\mathcal{H}$ is rule saturated for* $\mathbf{G3m}_*$ *according to definition 5.1.*

Proof. This needs to be proven for each different condition of definition 5.1 separately. This can be done using proof by contradiction. We only prove condition 3, the most interesting one, the others are similar.

Condition 3:
Suppose B^{G3m} is a branch of either $\mathbf{Tls}_K$ or $\mathbf{Tls}_T$ and is not saturated for the $K\Box$ rule. This means that there is a formula $\Box A$ in a succedent of a sequent with segment-mark x for which the rule $R\Box$ is not applied in the branch B^{G3m}.

This means that in the branch B^{Tls} there is a formula $x : \Box A$ in the succedent of a sequent for which the $R\Box$ has not been used with $x : \Box A$ as principal. Otherwise, the rule $K\Box$ *was* applied in the translation function $\mathcal{H}$ with principal formula $\Box A$ in the sequent marked x.

This means that in the branch B^{Tls} there is a sequent in which the formula $x : \Box A$ is in the antecedent which has not been principal in any the $R\Box$ rule in the branch. This, however, contradicts the fact that the branch B^{Tls} is rule saturated according to definition 5.13. Therefore B^{G3m} has to be saturated for the $K\Box$ rule.

The same argument can be made for when B^{G3m} is a branch of either $\mathbf{Tls}_{K4}$ or $\mathbf{Tls}_{S4}$ and is not saturated for the $4\Box$ rule. This means that in either case, B^{G3m} meets condition 3 of definition 5.1. ∎

Corollary 6.30. *If $\mathfrak{M}$ is a* $\mathbf{Tls}_*$ *Schütte model for the sequent $x : \Gamma \Rightarrow x : \Delta$, then $\mathfrak{M}$ is a* $\mathbf{G3m}_*$ *Schütte model for the equivalent sequent $\Gamma \Rightarrow \Delta$, where $*$ is either K, T, K4 or S4.*

6.2 Equality of Schütte countermodels

It is now straightforward to prove that these three calculi exactly produce the same Schütte countermodels.

Theorem 6.31 (Schütte Model Equivalence)**.** *$\mathfrak{M}$ is a* $\mathbf{G3m}_*$ *Schütte model for the sequent S, if and only if $\mathfrak{M}$ is a* $\mathbf{Ths}_*$ *Schütte model for the equivalent tree hypersequent*

S' if and only if $\mathfrak{M}$ is a $\mathbf{Tls}_*$ *Schütte model for the equivalent labeled sequent S'', where $*$ is either K, T, K4 or S4.*

Proof. Evident from Corollary 6.8, 6.15 and 6.30 ■

6.3 Limitation of the Schütte model equivalence theorem

A limitation of the theorem of Schütte model equivalence is that we only look at countermodels for sequents which are directly translatable into sequents of $G3m$, while both **Tls** and **Ths** are able to express more than just these single label tree-labeled sequent, or single sequents tree-hypersequents. In **Ths** we can for example also try to find countermodels for tree-hypersequents which contain more than one sequent such as the following sequent:

$$p \Rightarrow q/p \Rightarrow q; p \Rightarrow q$$

If this is a tree-hypersequent of $\mathbf{Ths}_K$, the branch existing of this one sequent is already saturated and via the Schütte set construction of definition 5.10 and the countermodel construction of 4.2, we come to the countermodel of figure 6.2.

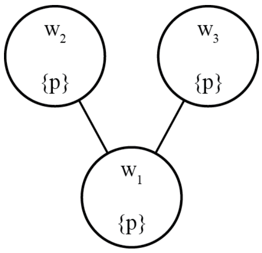


FIGURE 6.2: $\mathbf{Ths}_K$ Schütte countermodel of the tree-hypersequent $p \Rightarrow q/p \Rightarrow q; p \Rightarrow q$.

There is, however, no way of translating this tree-hypersequent directly into $\mathbf{G3m}_K$. The fact that is consists of three sequents already implies that the $K\Box$ rule of $\mathbf{G3m}_K$ is used already twice. From the perspective of the Schütte countermodel construction it seems as if the tree-hypersequent should be translated in a partial derivation in $G3m$ and not in a single sequent.

One could decide to just use the interpretation of $p \Rightarrow q/p \Rightarrow q; p \Rightarrow q$ and use the sequent $\Rightarrow I(p \Rightarrow q/p \Rightarrow q; p \Rightarrow q)$ as a translation in $\mathbf{G3m}_K$. This, however, can produce the Schütte countermodel in figure 6.3, while this is not a Schütte countermodel of the tree-hypersequent $p \Rightarrow q/p \Rightarrow q; p \Rightarrow q$. Does this mean that $\mathbf{G3m}_K$ can construct more Schütte countermodels for translations of complex tree-hypersequents than $\mathbf{Ths}_K$ can? This is not the case, because the sequent $\Rightarrow I(p \Rightarrow q/p \Rightarrow q; p \Rightarrow q)$ is also a tree-hypersequent, and because of the Schütte model equivalence theorem this produces the same Schütte modal in $\mathbf{G3m}_K$ as in $\mathbf{Ths}_K$.

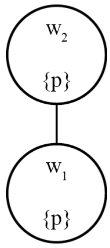


FIGURE 6.3: $\mathbf{G3m}_K$ Schütte countermodel of the sequent $\Rightarrow I(p \Rightarrow q/p \Rightarrow q; p \Rightarrow q)$.

Because of this it is not a problem that the Schütte model equivalence theorem only considers simple sequents. If we consider more complex sequents like Tree-hypersequents with multiple sequents or a labeled sequent with multiple prefixes, we will use the interpretation of these sequents and use that to construct a simple sequent from it. In this sense, the theorem considers all cases.

6.4 Evaluating the three sequent systems based on Schütte countermodel generation

With the proof of Schütte model equivalence it is shown that the three sequent systems construct the same set of Schütte countermodels for the modal logics K, T, K4 or S4. What does this mean for the overall evaluation of these three calculi. How do they compare?

At first glance the most striking result is that the simple **G3m** systems are as powerful as the extended sequent calculi systems of **Ths** and **Tls**. The extra syntax on top of the normal sequent calculus that these two systems bring is not necessary when aiming to make Schütte models, they can only construct Schütte models that are also possible to construct in **G3m**. From this point of view, the **G3m** systems seem the best. On the other hand, **G3m** needs backtracking on the $K\Box$ and $4\Box$ rule to construct the Schütte models, which is not needed in the other systems. This might be seen as the main problem **Ths** and **Tls** overcome by introducing the extra syntax from the Schütte countermodel construction point of view.

It is also possible to look at some good and bad aspects of the other two systems based on the way Schütte model construction is done. **Ths** shows the relation between sequents and the worlds in the countermodel the best. The Schütte sequents in the Schütte set are basically the different sequents in the tree hypersequents along the branch, and because in these sequents along the branch the atomic propositions are upwards cumulative it is even possible to directly construct the countermodel from the leaf sequent in the branch (if we use a finite proof procedure). In this sense one might see **Ths** as a system that represents the relation between Kripke semantics and sequents that is visible in the Schütte method for modal logics most elegant.

Tls makes for easy countermodel construction because the labels and relational atoms are all already in the sequents. But this also makes this system less elegant. All formulas for all worlds are gathered together in one sequent, while in **Ths** and **G3m** there are separate sequents for separate worlds. One could therefore argue that **Ths** and **G3m** show the close relationship between the sequents and worlds in the Schütte countermodel more clearly than the labeled approach.

6.5 Comments on alternative geometric rules for Tls

For the labeled logic we did not use the standard reflexivity and transitivity rule used by Negri in [19] which are shown below. However, 'translations' of the tree-hypersequent reflexivity and transitivity rules into the labeled system were used. This is possible because these kind of calculi are so similar, and a rigorous explanation of how one can translate rules between the calculi is explained in [12]. The difference between these two pairs of rules if that the original ones of Negri directly encode transitivity and reflexivity in the relational atoms, and then use the $L\Box$ rule to unfold modal formulas according to the transitive or reflexive atoms. The translated rules use the modal formulas directly in the transitivity and reflexivity rules, as is the case for these rules in **Ths** and **G3m**.

$$\frac{\Gamma, xRx \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\, ref' \qquad \frac{\Gamma, xRy, yRz, xRz \Rightarrow \Delta}{\Gamma, xRy, yRz \Rightarrow \Delta}\, trans'$$

There is a specific reason why the standard instances of these rules are not used here, and it has to do with the fact that when we use these rules the tree-like structure of the tree-labeled sequents is not preserved, one instead has to include (partially) transitive or reflective trees into the admissible structures the relational atoms can form in a labeled sequent. This itself is not something that makes these rules problematic when comparing the calculus to **Ths** and **G3m**, but it has an impact on the way the Schütte method is used on the **Tls** calculi. It means that in the set of sequent construction of definition 5.14 it is not necessary anymore to explicitly define the transitivity and the reflexivity of the relation function. With the *ref'* and *trans'* rules the reflexivity and transitivity is of the relational function is already built into relational atoms occurring in the antecedent of the **Tls** branch.[4] This makes it different from **G3m** and **Ths**, which need to have the transitivity or reflexivity of the relation function explicitly stated. There is however another reason the alternative rules for the **Tls** calculus were used.

[4]These rules are also problematic when we only limiting ourselves to *tree*-labeled sequents.

This has to do with the transitivity rule *trans'* and how that affects formulas in the sequents of the constructed Schütte set of sequents. Because the *trans'* works totally with the relational atoms, the use of this rule does not duplicate the principal $\Box A$ formula itself to the related world. The difference can be seen in example 6.1, where there is no labeled formula $y : \Box A$, but a relational atom xRz instead. This makes showing that the Schütte set of sequents are the same for translations of the three calculi a bit of a problem. It would lead to some $\Box A$ formulas being in some sequents of the Schütte sets constructed from the $\mathbf{Ths}_{K4/S4}$ and $\mathbf{G3m}_{K4/S4}$ branches while they do not appear in the corresponding sequents of the Schütte set constructed by the translated branch in $\mathbf{Tls}_{K4/S4}$. This difference is, however, only limited to only a set of $\Box A$ formulas. This means that the Schütte models created from these Schütte sets still are equal because the model constructions of definition 5.4, 5.10 and 5.14 only take into account the atomic propositions of each sequent.[5]

Example 6.1. *Example of the trans rule in either* $\mathbf{Tls}_{K4}$ *or* $\mathbf{Tls}_{S4}$*:*

$$\frac{x : \Box A, y : \Box A, xRy, yRz \Rightarrow}{x : \Box A, xRy, yRz \Rightarrow}\ trans$$

Example of the alternative trans' rule in either $\mathbf{Tls'}_{K4}$ *or* $\mathbf{Tls'}_{S4}$ *as in [19]:*

$$\frac{x : \Box A, xRy, yRz, xRz \Rightarrow}{x : \Box A, xRy, yRz \Rightarrow}\ trans'$$

This concludes the reason for using alternative transitive and reflexive rules. It changes some occurrences of modal formulas in some Schütte sequents in the Schütte sets, but because it does not change the atoms that are in these sequents or the relational atoms, it does not change the Schütte models.

[5]And the saturation of these sequents lead to the assurance that these models are countermodels. But the saturation is not in danger by using these alternative rules.

Chapter 7

Conclusion

In this thesis, three different sequent systems for modal logic were analyzed with respect to their role in the creation of Schütte countermodels. These sequent systems were a basic modal sequent calculus, a labeled sequent calculus, and a hypersequent calculus.

To compare these proof systems, first, the Schütte style completeness proof was explored in the context of the modal logic, Kripke semantics, and sequent systems. This lead to an overarching structure which is apparent in all the Schütte completeness proofs and consists of a proof procedure, rule-saturation and countermodel construction based on a failed branch of the derivation tree. As an extra step in the Schütte countermodel construction, the concept of a Schütte set of sequents was defined. This extra step shows the closeness between sequents and countermodels, where all formulas in the antecedent of a sequent are true in the Schütte model, and all formulas in the consequent are refuted in the countermodel. Besides this, the Schütte set also shows that in the Schütte countermodel, the modal relations between worlds coincide with relations between sequents in the Schütte set which can directly be found in the branch.

As the main result this thesis proved that the Schütte models that could be created with the sequent systems **G3m**, **Ths** and **Tls** were exactly the same for the logics K, T, $\mathsf{K4}$ or $\mathsf{S4}$. This shows that these systems are equivalent with respect to Schütte model construction. As a consequence of this result, it is now evident that none of these calculi is in any sense better in creating Schütte countermodels. This also means that specific measures of the Schütte countermodels, like the smallest depth of the model, or the smallest amount of worlds is the same for these three calculi. This must be, because for each Schütte model that can be created from a branch on one of these calculi, an identical Schütte model can be created from a translated branch in the other calculus, so none of these sequent calculi is able to make smaller (in depth or size) models than the other.

For the translations of the branches of the different sequent calculi themselves, it is clear that the translation from **Tls** to **G3m** is more difficult than the other two translations. While the translation of **Ths** to **Tls** and **G3m** to **Ths** can be done easily, translating per rule in the original branch, this is not the case of the translation from **Tls** to **G3m**. For this translation, the original branch in **Tls** needs to be reordered. This is needed because of the inflexibility of the $K\Box$ or $4\Box$ rules of the **G3m** calculus, and because of the segmentation in the **G3m** branches. The same problems would arise if we would have translated branches of **Ths** to **G3m** directly. However, the fact that the unordered branches of **Tls** cannot be translated into **G3m** has no impact on which Schütte models can be created with these calculi. The reason being that reordering a branch in **Tls** does not change the created Schütte set created from that branch. In that sense, the fact that the translation is a bit more difficult from **Tls** to **G3m** is not important. One could even say that, with respect to Schütte models, the only thing that is gained by the additions of labels in **Tls**[1] compared to **G3m** is more ways to achieve the same.

Conceptually, the most interesting result in this thesis is how derivations in these three different kinds of sequent calculi compare with respect to the Kripke semantics. Where we have segments of the derivation representing Schütte sequents and worlds in a Kripke model in the basic sequent systems, we have sequents in the tree-hypersequent system and prefixes in the labeled system. This shows that these systems differ on what level of syntax they encode the worlds and relations of the Kripke semantics in the sequents: on the derivation level, on the sequents level or on the level of the formulas.

7.1 Limits and future research

Even though this thesis makes a first step at formalising the Schütte method, what exactly counts as a Schütte countermodel is still up for debate. These countermodels are created from failed derivations of the analytic calculus, but what extra conditions there are for the construction and the derivation can be chosen differently than is done in this thesis. Here we have defined a Schütte countermodel being a model which is constructed from the saturated application of the rules of a calculus in a branch on a derivation, and the whole branch was used in the creation of the countermodel.

Choosing different constraints for a model to be called a Schütte countermodel can also change the effectiveness of analytic calculi to make them. A possible other condition that could be put on the construction of the Schütte countermodels is that they should be created only from a failed *leaf* of a derivation, instead of the whole failed branch. This would, for example, limit the possibility of the $\mathbf{G3m}_*$ calculi to make countermodels because the different sequents of the Schütte set, and the different worlds of the Schütte

[1] Or the tree structure in **Tls** for that matter.

countermodels are found along the derivation branch[2]. This is not the case for $\mathbf{Tls}_*$ or $\mathbf{Ths}_*$ which have the sequents of the Schütte set and the worlds of the Schütte model occurring in each node of the branch. But even though it is possible to alter the way these Schütte models are constructed from the proof systems, the specific constructions chosen in this thesis are sensible and intuitive ones.

A way in which this research can be continued is extending it to other logics and analyzing the Schütte method for other calculi. In this thesis we only looked at the modal logics K, T, $\mathsf{K4}$ or $\mathsf{S4}$. It might be possible to do something similar for other logics like GL, D or $\mathsf{S5}$. Besides this, the research could also be expanded to include other calculi which are inspired by sequent calculi like a display calculus.

In this thesis, we compared analytic calculi based on the Schütte completeness proof, that is, we used the Schütte method to construct countermodels from a proof system as a tool to compare sequent calculi. This Schütte completeness proof lends itself very well for comparison because it created countermodels as a part of the proof, which then can be used as a measure, by for example looking at the depth or size of these models. However, the same approach might be used for comparison based on other meta-proofs about proof systems. Can we compare analytic calculi based on other standard proofs of properties common to those calculi?

[2]This shows immediately that this choice for limiting the Schütte countermodel construction is a bad one, because the failed branch in a derivation of $\mathbf{G3m}_*$ does give enough information to make a countermodel

Chapter 8

Schütte Completeness for Modal Dyckhoff Calculus $\mathbf{DY_K}$

In this extra chapter we will construct a Schütte completeness proof for the intuitionistic modal calculus $\mathbf{DY_K}$[1] presented in [14] which is an extension of the intuitionistic propositional calculus presented by Dyckhoff in [7]. This is done as an exploration into how Schütte counter-models could be constructed for non-classical modal logics. The Schütte completeness proof shown here is inspired by the proof in [26] for the propositional Dyckhoff calculus. $\mathbf{DY_K}$ is especially interesting because it is terminating, and differs from the classical modal calculi treated in the rest of this thesis by the fact that it only allows for single formula succedent sequents. We will see that this changes the way extra nodes in the Schütte model are created. But before we can go into the calculus and the proof, we first have to look at the specific Kripke models which can be used as a semantic for intuitionistic modal logic.

8.1 Intuitionistic modal Kripke models

Intuitionistic modal Kripke frames differ a bit from normal Kripke models. This is because we have two kinds of relations over the worlds, the modal relation, and the intuitionistic relation.

Definition 8.1. An intuitionistic Kripke Frame is a tuple $\mathfrak{F} =< W, \leq, R >$ such that:

1. W is a non empty set (of possible worlds).
2. $\leq$ is a partial order on W.

[1] We will only consider the $\Diamond$ free part of intuitionistic modal logic

3. $R \subseteq (W \times W)$ is a binary relation on W. If wRw' then we say world w' is accessible from w.

With the use of this definition, we can then define what a full intuitionistic Kripke model is.

Definition 8.2. An intuitionistic Kripke Model is a tuple $\mathfrak{M} =< W, \leq, R, V >$ such that:

1. $< W, \leq, R >$ is an intuitionistic Kripke frame
2. V is a function assigning a truth value to each atomic formula p for each world $w \in W$. $V(w,p) \in \{0,1\}$. We require V to be monotone with respect to $\leq$. Such that if $V(w,p) = 1$ and $w \leq w'$, then $V(w',p) = 1$.

We can now give an interpretation of the connectives for intuitionistic modal Kripke models.

Definition 8.3. let $\mathfrak{M} =< W, \leq, R, V >$ be an intuitionistic Kripke model with $x, y, z \in W$ and A, B and C modal formula and p an atom. Then the truth of a formula is inductively defined relative to model $\mathfrak{M}$ and world x in the following way.

$\mathfrak{M}, x \nvDash \bot$

$\mathfrak{M}, x \vDash p \iff V(x,p) = 1$

$\mathfrak{M}, x \vDash A \wedge B \iff \mathfrak{M}, x \vDash A$ and $\mathfrak{M}, n \vDash B$

$\mathfrak{M}, x \vDash A \vee B \iff \mathfrak{M}, x \vDash A$ or $\mathfrak{M}, x \vDash B$

$\mathfrak{M}, x \vDash A \rightarrow B \iff$ For all y for which $x \leq y$: If $\mathfrak{M}, y \vDash A$ then $\mathfrak{M}, y \vDash B$

$\mathfrak{M}, x \vDash \Box A \iff$ For all y for which $x \leq y$: For all z: If yRz, then $\mathfrak{M}, z \vDash A$

Lemma 8.4 (monotonicity)**.** *For each formula A, model $\mathfrak{M}$ and world w in $\mathfrak{M}$, if $\mathfrak{M}, w \vDash A$ then for each world w' such that $w \leq w'$ it is the case that $\mathfrak{M}, w' \vDash A$.*

8.2 $\mathbf{DY_K}$ and some preliminary theorems

Definition 8.5. $\mathbf{DY_K}$ consist of the following rules:

Axioms:

$$\frac{}{\Gamma, p \Rightarrow p}\ Ax \qquad \frac{}{\Gamma, \bot \Rightarrow D}\ L\bot$$

Rules:

$$\frac{\Gamma, A, B \Rightarrow D}{\Gamma, A \wedge B \Rightarrow D}\ L\wedge \qquad \frac{\Gamma \Rightarrow A \quad \Gamma \Rightarrow B}{\Gamma \Rightarrow A \wedge B}\ R\wedge$$

$$\frac{\Gamma, A \Rightarrow D \quad \Gamma, B \Rightarrow D}{\Gamma, A \vee B \Rightarrow D}\ L\vee \qquad \frac{\Gamma \Rightarrow A_{1,2}}{\Gamma \Rightarrow A_1 \vee A_2}\ R\vee_{1,2}$$

$$\frac{\Gamma, A \Rightarrow D}{\Gamma, p, p \to A \Rightarrow D}\ Lp\to \qquad \frac{\Gamma, A \Rightarrow B}{\Gamma \Rightarrow A \to B}\ R\to$$

$$\frac{\Gamma, A \to (B \to C) \Rightarrow D}{\Gamma, A \wedge B \to C \Rightarrow D}\ L\wedge\to \qquad \frac{\Gamma, A \to C, B \to C \Rightarrow D}{\Gamma, A \vee B \to C \Rightarrow D}\ L\vee\to$$

$$\frac{\Gamma \Rightarrow A \quad \Pi, \Box\Gamma, B \Rightarrow D}{\Pi, \Box\Gamma, \Box A \to B \Rightarrow D}\ L\Box\to \qquad \frac{\Gamma, B \to C \Rightarrow A \to B \quad \Gamma, C \Rightarrow D}{\Gamma, (A \to B) \to C \Rightarrow D}\ L\to\to$$

$$\frac{\Gamma \Rightarrow A}{\Pi, \Box\Gamma \Rightarrow \Box A}\ K\Box$$

Lemma 8.6. *All rules of* $\mathbf{DY_K}$ *except* $K\Box$, $L\Box$, $R\vee$ *and* $R\to\to$ *are invertible.*

We will also define the size of a sequent analogous to how this is done in [7].

Definition 8.7. To define the size of a sequent we first need to define the weight of a formula. The weight of a formula is inductively defined as follows:

1. $wt(p) = 1$
2. $wt(\Box A) = wt(A) + 1$
3. $wt(A \vee B) = wt(A \to B) = wt(A) + wt(B) + 1$
4. $wt(A \wedge B) = wt(A) + wt(B) + 2$

If we look at a sequent as the multiset $\{S^a, S^s\}$ as consisting of the multisets of the antecedent and the succedent. We can now define an ordering over sequents. We say that for two sequents S and T that sequent $S >> T$ if $\{T^a, T^s\}$ is the result of replacing one or more formula of $\{S^a, S^s\}$ for zero or more formula of lower weight. We say that T is of lower size than S.

Definition 8.8. A sequent $\Phi, \Box\Gamma \Rightarrow A$ is critical if:

1. A is of the form $B \vee C$, $\Box B$ or p.
2. For all formulas $B \in \Gamma$ they are of the form p,$\Box B$, $(B \to C) \to D$, $p \to C$ or $\Box B \to C$.
3. there is no formula $p \to B \in \Gamma$ for which also $p \in \Gamma$.
4. If A is the atom p, it cannot be the case that $p \in \Gamma$

Notice that none of the rules $Ax, L\perp, L\wedge, R\wedge, L\vee, Lp \to, R \to, L\wedge \to$, and $L\vee \to$ can be applied to a critical sequent.

Theorem 8.9. *A critical sequent $\Phi, \Box\Gamma \Rightarrow A$ is an intuitionistic tautology if and only if at least one of the following is true:*

1. *A is of the form $B \vee C$ and and at least one of the following sequents is an intuitionistic tautology $< \Phi, \Box\Gamma \Rightarrow B >, < \Phi, \Box\Gamma \Rightarrow C >$*
2. *A is of the form $\Box B$ and $\Gamma \Rightarrow B$ is an intuitionistic tautology.*
3. *There is an implication $(B \to C) \to D \in \Phi$ such that both $\Phi - \{(B \to C) \to D\}, \Box\Gamma, B, C \to D \Rightarrow C$ and $\Phi - \{(B \to C) \to D\}, \Box\Gamma, D \Rightarrow A$ are intuitionistic tautologies.*
4. *There is an implication $\Box E \to F \in \Phi$ such that both $\Gamma \Rightarrow E$ and $\Phi - \{\Box E \to F\}, \Box\Gamma, F \Rightarrow A$ are intuitionistic tautologies.*

Proof. The proof is from contradiction via countermodels: If we have countermodels as a result of the fact that statements 1-4 are false, it is possible to also construct a countermodel to the critical sequent.

Let us look at the case that A is of the form $\Box G$, because the cases that it is an atom or $\perp$ are easier, and the case that A is $B \vee C$ is explained in [26] and is also quite straightforward. Let $(B_1 \to C_1) \to D_1, ..., (B_n \to C_n) \to D_n$ be a list of all the formulas of that form in Φ. Let $\Box E_1 \to F_1, ..., \Box E_m \to F_m$ be a list of all the formulas of that form in Φ.

1. Statements 1 is immediately false because formula A is not of that form.

2. Because statement 2 is not true there must be a countermodel for the sequent $\Gamma \Rightarrow G$. Let X be these countermodels with the root x_i which is the world in which the sequent is countered.

3. Because statement 3 is not true for each formula $(B_i \to C_i) \to D_i \in \Phi$ there must be a countermodel for one of the sequents $\Phi - \{(B_i \to C_i) \to D_i\}, B_i, \Box\Gamma, C_i \to D_i \Rightarrow C_i$ or $\Phi - \{(B_i \to C_i) \to D_i\}, \Box\Gamma, D_i \Rightarrow A$. If we have a countermodel to the sequent $\Phi - \{(B_i \to C_i) \to D_i\}, \Box\Gamma, D_i \Rightarrow A$ we also have immediately a countermodel to the sequent $\Phi, \Box\Gamma \Rightarrow A$ and the proof is easily shown. Because of this, we assume the more interesting case, that for every i we only have a countermodel for $\Phi - \{(B_i \to C_i) \to D_i\}, \Box\Gamma, B_i, C_i \to D_i \Rightarrow C_i$. Let K_i be these countermodels with the root k_i which is the world in which the sequent is countered.

4. Because statement 4 is not true for each formula $\Box E_j \to F_j \in \Phi$ there must be a countermodel for one of the sequents $\Gamma \Rightarrow E_j$ or $\Phi - \{\Box E_j \to F_j\}, \Box\Gamma, F_j \Rightarrow A$. If we have a countermodel to the sequent $\Phi - \{\Box E_j \to F_j\}, \Box\Gamma, F_j \Rightarrow A$ we also have immediately a countermodel to the sequent $\Phi, \Box\Gamma \Rightarrow A$ and the proof is easily shown. Therefore, we assume that for every j we only have a countermodel for $\Gamma \Rightarrow E_j$. Let L_j be these countermodels with the root l_j which is the world in which the sequent is countered.

With the use of all these countermodels, it is possible to construct a bigger countermodel which refutes all. We do this by adding creating the model $\mathfrak{M} =< \mathfrak{M}^W, \mathfrak{M}^{\leq}, \mathfrak{M}^R, \mathfrak{M}^V >$[2]. All the countermodels $X, K_1, ..., K_n, L_1, ..., L_M$ together plus another world w which is the root of the model. $w \leq k_i$ for $1 \leq i \leq n$, wRl_j for $1 \leq j \leq M$ and wRx.

$$
\begin{aligned}
\mathfrak{M}^W &= X^W \cup K_1^W \cup ... \cup K_n^W \cup L_1^W \cup ... \cup L_m^W \cup \{w\} \\
\mathfrak{M}^{\leq} &= X^{\leq} \cup K_1^{\leq} \cup ... \cup K_n^{\leq} \cup L_1^{\leq} \cup ... \cup L_m^{\leq} \cup \{w \leq k_i | 1 \leq i \leq n\} \\
\mathfrak{M}^R &= X^R \cup K_1^R \cup ... \cup K_n^R \cup L_1^R \cup ... \cup L_m^R \cup \{wRl_j | 1 \leq j \leq M\} \cup \{wRx\} \\
\mathfrak{M}^V &= X^V \cup K_1^V \cup ... \cup K_n^V \cup L_1^V \cup ... \cup L_m^V \cup \{V(p, w) = 1 | p \in \Phi\}
\end{aligned}
$$

Note that forcing all atoms p in w for which $p \in \Phi$ does not violate monotonicity. This is because in all atoms in Φ are also forced in the root nodes of all sub-models $K_1, ..., K_n$ (and monotonicity holds in these models!). And the roots of these sub-models are the only worlds for which $w \leq w'$. The total model is also visualised in figure 8.1.

To show that $\mathfrak{M}$ is a countermodel to the sequent $\Phi, \Box\Gamma \Rightarrow A$ we need to show that for each formula $\phi \in \Phi$ and each formula $\phi \in \Box\Gamma$ that $\mathfrak{M}, w \vDash \phi$ and that $\mathfrak{M}, w \nvDash A$.

[2] In the definition below, we assume the transitive closure of $\mathfrak{M}^{\leq}$

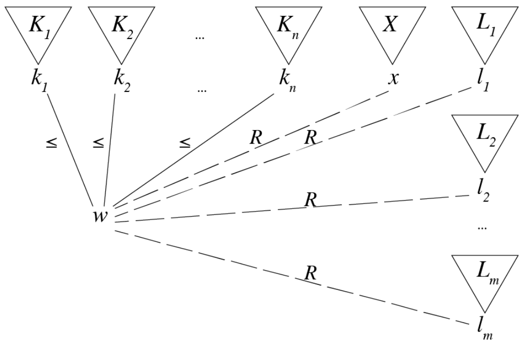


FIGURE 8.1: Countermodel $\mathfrak{M}$ with root world w.

Because A is of the form $\Box G$ and there is the root x of sub-model X for which wRx and $\mathfrak{M}, x \nVdash G$ it is true that $\mathfrak{M}, w \nVdash \Box G$.

For each world w' for which wRw' we know that $\mathfrak{M}, w' \vDash B$ for each formula $\Box B \in \Box\Gamma$. We know this because these are precisely the root-worlds $x, l_1, ..., l_m$ of the sub-models $X, L_1, ..., L_m$. And in all of these worlds all formulas in $b \in \Gamma$ are forced.

Because the sequent $\Phi, \Box\Gamma \Rightarrow A$ is critical we know that the sort of formula occurring in Φ is limited to the forms described in definition 8.8, we therefore only have to look at formulas of these forms.

For each atom $p \in \Phi$ it is forced in w by definition.

For each formula $\phi \in \Phi, \Box\Gamma$ where ϕ is of the form $p \to B$, $\Box B$ or $\Box E \to F$ we know that in every world w' which is not w for which $w \leq w'$ it is the case that $\mathfrak{M}, w' \vDash \phi$. We know this because these w' are precisely the root-worlds $k_1, ..., k_n$ of the sub-models $K_1, ..., K_m$ plus for each of these sub-models the other worlds in these sub-models which are *following* from the root node. We know that for each formula ϕ and for each k_i where $1 \leq i \leq n$ that $\mathfrak{M}, k_i \vDash \phi$ because the model K_i is a countermodel to the sequent $\Phi - \{(B_i \to C_i) \to D_i\}, B_i, \Box\Gamma, C_i \to D_i \Rightarrow C_i$. And it is also true for each world *following* k_i in model K_i because of monoticity in these sub-models.

For each formula $p \to C \in \Phi$ we know that it cannot be that $p \in \Phi$ by definition 8.8 therefore $\mathfrak{M}, w \nVdash p$. Besides that we know that in every world w' which is not w for which $w \leq w'$ it is the case that $\mathfrak{M}, w' \vDash p \to C$. Therefore $\mathfrak{M}, w \Vdash p \to C$.

For each formula $\Box B \in \Box\Gamma$ we know that all worlds w' for which wRw' it is the case that $\mathfrak{M}, w' \vDash B$. These are the worlds $l_1, ..., l_m, x$ and we know that they force each formula B because these worlds are countermodels to the sequents $< \Gamma \Rightarrow E_1 >, ..., < \Gamma \Rightarrow E_m >$ $, < \Gamma \Rightarrow G >$. Therefore, $\mathfrak{M}, w \vDash B$.

For each formula $\Box E_j \to F_j \in \Phi$ we know that $\mathfrak{M}, l_j \nvDash E_j$. Therefore $\mathfrak{M}, w \nVdash \Box E_j$. Besides that we know that in every world w' which is not w for which $w \leq w'$ it is the case that $\mathfrak{M}, w' \vDash \Box E_j \to F_j$. Therefore $\mathfrak{M}, w \Vdash \Box E_j \to F_j$.

For each formula $(B_i \to C_i) \to D_i \in \Phi$ we know that $\mathfrak{M}, k_i \nvDash C_i$ and $\mathfrak{M}, k_i \vDash B_i$ and thus $\mathfrak{M}, k_i \nvDash B_i \to C_i$. Because of this and $w \leq k_i$ we know that $\mathfrak{M}, w \nvDash (B_i \to C_i)$. We know that $\mathfrak{M}, k_i \vDash (B_i \to C_i) \to D_i$ because $k_i \nvDash B_i \to C_i$ and for each world for w' which for which $k_i \leq w'$ we also know the following: $\mathfrak{M}, w' \vDash B_i$ and $\mathfrak{M}, w' \vDash C_i \to D_i$ because of monoticity and thus in each world w' $\mathfrak{M}w' \vDash (B_i \to C_i) \to D_i$ regardless of whether C_i becomes forced in world w'. We also know that $(B_i \to C_i) \to D_i$ is also forced in all other root worlds k_k of the sub-models K_k where $k \neq i$ and all worlds in these sub-models *following* from the root node because $(B_i \to C_i) \to D_i \in \Phi$ and monoticity of the sub-models. This means that for all different worlds v for which $w \leq v$ $\mathfrak{M}, v \vDash (B_i \to C_i) \to D_i$ and $\mathfrak{M}, w \nvDash B_i \to C_i)$ resulting in the fact that $\mathfrak{M}, w \vDash (B_i \to C_i) \to D_i$.

This means that in $\mathfrak{M}$ there is the world w which forces all the formulas in the antecedent of the critical sequent $\Phi, \Box\Gamma \Rightarrow A$ but does not force the formula in the succedent and therefore $\mathfrak{M}$ is a countermodel to the critical sequent.

This means that if none of the statements 1-4 is true, then the sequent $\Phi, \Box\Gamma \Rightarrow A$ is not an intuitionistic tautology. ■

8.2.1 Proof procedure and Schütte proof

With the result of theorem 5.1, it is possible to create a proof procedure which shows the Schütte completeness of the calculus $\mathbf{DY_K}$ for intuitionistic Kripke models. We first use all the invertible rules exhaustively. If none of these rules are applicable anymore, we use the rules $L\Box \to$, $L \to\to$, $R\vee_{1,2}$ and $K\Box$ together on that sequent.

Definition 8.10 (Proof Procedure)**.** The proof procedure starts with writing the to be proven sequent S at the root of the reduction tree, and then use the following proof function on each leaf of the tree until no rule can be applied anymore. the function also returns either true or false which will indicate whether we found a proof for the sequent.

Suppose the leaf is of the form $\Phi, \Box\Gamma \Rightarrow D$, then try to apply the rules of $\mathbf{DY_K}$ in the following way in the following order to the leaf. There are 9 cases:

1. if D is p and $p \in \Phi$ or $\bot \in \Phi$, end the application of rules to the leaf, the leaf is an axiom and return true.
2. If D is $A \wedge B$ apply the rule $R\wedge$ and write the two sequents $\Phi, \Box\Gamma \Rightarrow A$ and $\Phi, \Box\Gamma \Rightarrow B$ above the leaf in the reduction tree. Return true if both of the new leaf sequents return true.

3. If D is $A \to B$ apply the rule $R\to$ and write the sequent $\Phi, \Box\Gamma, A \Rightarrow B$ above the leaf. Return true if the new leaf sequent returns true.

4. If Φ contains a formula of the form $p \to A$ and the atom p, apply the rule $Lp \to$ to the leaf. Write the sequent $\Phi - p \to A, p, A, \Box\Gamma \Rightarrow D$ above it. Return true if the new leaf sequent returns true.

5. If Φ contains a formula of the form $A \wedge B \to C$, the rule $L\wedge \to$ is applied to the leaf. Write the sequent $\Phi - A \wedge B \to C, A \to (B \to C), \Box\Gamma \Rightarrow D$ above it. Return true if the new leaf sequent returns true.

6. If Φ contains a formula of the form $A \vee B \to C$, the rule $L\vee \to$ is applied to the leaf. Write the sequent $\Phi - A \vee B \to C, A \to C, B \to C, \Box\Gamma \Rightarrow D$ above it. Return true if the new leaf sequent returns true.

7. If Φ contains a formula of the form $A \wedge B$, the rule $L\wedge$ is applied to the leaf. Write the sequent $\Phi - A \wedge B, A, B, \Box\Gamma \Rightarrow D$ above it. Return true if the new leaf sequent returns true.

8. If Φ contains a formula of the form $A \vee B$, the rule $L\vee$ is applied to the leaf. Write the sequents $\Phi - A \vee B, A, \Box\Gamma \Rightarrow D$ and $\Phi - A \vee B, B, \Box\Gamma \Rightarrow D$ above it. Return true if both new leaf sequents return true.

9. If this step is reached in the function, we know that the leaf $\Gamma \Rightarrow D$ is critical. We will call the rules $L \to\to$, $L\Box \to$, $R\vee$ and $K\Box$ on the leaf multiple times if possible as shown beneath. We will return true if at least one of the following returns True:

   $K\Box$:
   If D is of the form $\Box A$ the rule $L\vee$ is applied to the leaf. We will write the sequent $\Gamma \Rightarrow A$ above the leaf and return true the new leaf sequents returns true.

   $R\vee$:
   If D is $A \vee B$ apply the rule $R\vee$ two times to the leaf and write the sequents $\Phi, \Box\Gamma \Rightarrow A$ and $\Phi, \Box\Gamma \Rightarrow B$ above the leaf. Return true if *at least one* of the new leafs return true.

   $L\Box \to$:
   Let $\Box E_1 \to F_1, ..., \Box E_n \to F_n$ be a list of all the formulas of that form in Φ. We will apply the rule $L\Box \to$ using each of these formulas as principal. This means that for each $1 \leq i \leq n$ we will write the sequents $\Gamma \Rightarrow E_i$ and $\Phi - \Box E_i \to F_i, F_i, \Box\Gamma \Rightarrow D$ above it. Return true if there is at least one i for which both new leaf sequents return true.

   $L \to\to$:
   Let $(A_1 \to B_1) \to C_1, ..., (A_m \to B_m) \to C_m$ be a list of all the formulas of that form in Φ. We will apply the rule $L\Box \to$ (and immediately $R \to$ to the left premise) using each of these formulas as principal. This means that for each

$1 \leq j \leq m$ we will write the sequents $\Phi - (A_j \to B_j) \to C_j, A_j, B_j \to C_j, \Box\Gamma \Rightarrow B_j$ and $\Phi - (A_j \to B_j) \to C_j, C_j, \Box\Gamma \Rightarrow D$ above it. Return true if there is at least one j for which both new leaf sequents return true.

If none of 1-9 are applicable to a leaf sequent, stop the function for that leaf. If a sequent does not return true, it is false.

Theorem 8.11. *The procedure is finite*

Proof. Each application of the cases 1-9 of the proof function writes zero or more sequents above the leaf of lower size than the original leaf sequent according to the weight function described in definition 8.7. Because the weight of any formula cannot be smaller than 1, and it is impossible to have infinite formulas in a sequent, the procedure terminates. ■

Theorem 8.12. *If the procedure fails (the root sequent returns false) we can create a Schütte countermodel that refutes the root sequent based on a sub-tree of the reduction tree.*

Proof. We can create a sub-tree of the reduction tree the following way:

Start at the root sequent and go up the reduction tree bottom-up, choosing what parts of the tree will stay based on the application of the function at that node of the reduction tree.

We know case 1 is not applied to a node, otherwise, the of that node would not be false.

If the function applied cases 2-8 to a node, we choose one next node which has returned false. We know there is one because the previous sequent was false.

If the function applied case 9 to a node, we choose one next node for *each* application of a rule $L \to\to$, $L\Box \to$, $R\vee$ and $K\Box$ which has returned false. We know there is one for each application because the previous sequent was false.

we now can create a countermodel for the root sequent of this sub-tree. This is done with induction on the depth of the sub-tree. The depth is straightforwardly defined as the longest row of applied cases 1-9 of the proof procedures function.

If $\mathfrak{D} = 0$:
We know the only sequent in the sub-tree must be of the form $\Phi, \Box\Gamma \Rightarrow D$ where all formulas in Φ are either atoms or implications of the form $p \to A$ where $p \notin \Phi$ and D is an atom q. We know all this because otherwise one of the parts 2-9 of the proof function could be applied. We also know that the atom $q \notin \Phi$, otherwise $\Phi, \Box\Gamma \Rightarrow D$ would be an axiom and not false.

Because of this we can create a countermodel $\mathfrak{M} =< W, \leq, R, V >$ such that:

1. $W = \{w\}$
2. The partial order $\leq$ on W is only consisting of $w \leq w$.
3. The binary relation R is empty
4. $V(w, p) = 1 \leftrightarrow p \in \Phi$

This model $\mathfrak{M}$ forces all formulas in $\Phi, \Box\Gamma$ in world w while $\mathfrak{M}, w \not\Vdash q$. This is trivial.

Induction Hypothesis:
If $\mathfrak{D} = n$ we can create a Schütte model that refutes the root sequent.

If $\mathfrak{D} = n + 1$ and the last call to the function applied the rules of 1-8:
Because the rules applied in 1-8 are all invertible a countermodel to the premise of that rule is also a countermodel to the conclusion of that rule.

If $\mathfrak{D} = n + 1$ and the last call to the function applied the rules of 9:
As shown in theorem 8.9 it is possible to use the countermodels of all the premises of the applied rules and construct a countermodel to the sequent to which 9 was applied to. ■

Example 8.1. *We can for example apply the function to the sequent $\Rightarrow \neg\neg\Box p \vee \neg\Box p$, which gives us the following tree.*

$$\dfrac{\dfrac{\dfrac{}{\Box p \Rightarrow \bot}}{\Rightarrow \Box p \to \bot}\,3 \qquad \dfrac{\dfrac{\Rightarrow p \quad \bot \Rightarrow \bot}{\Box p \to \bot \Rightarrow \bot}\,9}{\Rightarrow (\Box p \to \bot) \to \bot}\,3}{\Rightarrow (\Box p \to \bot) \vee ((\Box p \to \bot) \to \bot)}\,9$$

This gives us the countermodel $\mathfrak{M} =< W, \leq, R, V >$ where:

$$W = \{w_1, w_2, w_3, w_4\}$$
$$\leq = \{w_1 \leq w_2, w_1 \leq w_3\}$$
$$R = \{w_3 R w_4\}$$
$$V = \emptyset$$

In figure 8.2, we can see the countermodel $\mathfrak{M}$. It is a countermodel for $\neg\neg\Box p \vee \neg\Box p$ for the following reasons. $w_4 \nvDash p$ and thus $w_3 \nvDash \Box p$, because there is no other world w' such that $w_3 \leq w'$ $w_3 \vDash \neg\Box p$ and also $w_3 \nvDash \neg\neg\Box p$. Besides this $w_2 \vDash \Box p$, and $w_2 \nvDash \neg\Box p$. Therefore $w_1 \nvDash \neg\neg\Box p$ and $w_1 \nvDash \neg\Box p$ which results in $w_1 \nvDash \neg\neg\Box p \vee \neg\Box p$.

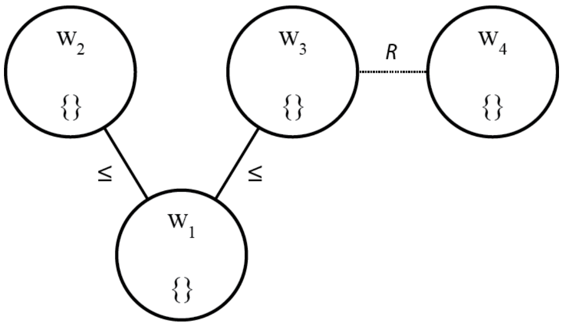


FIGURE 8.2: $\mathbf{DY}_K$ Schütte countermodel of the sequent $\Rightarrow \neg\neg\Box p \vee \neg\Box p$.

Bibliography

[1] A. Avron. On modal systems having arithmetical interpretations. *J. Symbolic Logic*, 49(3):935–942, 09 1984.

[2] A. Avron. The method of hypersequents in the proof theory of propositional non-classical logics. In W. Hodges, M. Hyland, C. Steinhorn, and J. Truss, editors, *Logic: from foundations to applications: European logic colloquium*, pages 1–32. Clarendon Press, 1996.

[3] E.W. Beth. *Semantic Entailment and Formal Derivability*. Noord-Hollandsche Uitgevers Maatschappij, 1955.

[4] M. Bílková. *Interpolation in modal logics*. PhD thesis, Univerzita Karlova, Filozofická fakulta, 2006.

[5] K. Brünnler. Deep sequent systems for modal logic. *Advances in modal logic*, 6:107–119, 2006.

[6] J.M. Dunn and N.D. Belnap. The substitution interpretation of the quantifiers. *Noûs*, 2(2):177–185, 1968.

[7] R. Dyckhoff. Contraction-free sequent calculi for intuitionistic logic. *J. Symb. Log.*, 57:795–807, 1992.

[8] M. Fitting. Tableau methods of proof for modal logics. *Notre Dame J. Formal Logic*, 13(2):237–247, 04 1972.

[9] M. Fitting. *Proof methods for modal and intuitionistic logics*, volume 169. Springer Science & Business Media, 1983.

[10] D. Garg, V. Genovese, and S. Negri. Countermodels from sequent calculi in multi-modal logics. In *Proceedings of the 2012 27th Annual IEEE/ACM Symposium on Logic in Computer Science*, pages 315–324. IEEE Computer Society, 2012.

[11] G. Gentzen. Collected works, 1969.

[12] R. Goré and R. Ramanayake. Labelled tree sequents, tree hypersequents and nested (deep) sequents. *Advances in modal logic*, 9:279–299, 2012.

[13] J Hintikka. *Two Papers on Symbolic Logic Form and Content in Quantification Theory and Reductions in the Theory of Types.* Helsinki, Societas Philosophica, 1955.

[14] R. Iemhoff. Terminating sequent calculi for two intuitionistic modal logics. *Journal of Logic and Computation*, 28(7):1701–1712, 10 2018.

[15] S. Kanger. *Provability in Logic.* Stockholm, Almqvist & Wiksell, 1957.

[16] P. Minari. Labeled sequent calculi for modal logics and implicit contractions. *Archive for Mathematical Logic*, 52, 11 2013.

[17] G. Mints. *A Short Introduction to Intuitionistic Logic.* Springer US, 2000.

[18] S Negri. Proof analysis in modal logic. *Journal of Philosophical Logic*, 34(5-6):507–544, 2005.

[19] S. Negri. Kripke completeness revisited. *Acts of Knowledge-History, Philosophy and Logic*, pages 247–282, 2009.

[20] S. Negri. Proof theory for modal logic. *Philosophy Compass*, 6(8):523–538, 2011.

[21] S. Negri. On the duality of proofs and countermodels in labelled sequent calculi. In *International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, pages 5–9. Springer, 2013.

[22] F. Poggiolesi. Method of tree-hypersequents for modal propositional logic. In D. Makinson, J. Malinowski, and H. Wansing, editors, *Towards Mathematical Philosophy*, pages 31–51. Berlin: Springer-Verlag, 2009.

[23] F. Poggiolesi. *Gentzen calculi for modal propositional logic*, volume 32. Springer Science & Business Media, 2010.

[24] K. Schütte. Ein system des verknüpfenden schliessens. *Archiv für mathematische Logik und Grundlagenforschung*, 2(2):55–67, Mar 1956.

[25] R.M. Smullyan. *First-Order Logic.* Berlin, Springer-Verlag, 1968.

[26] V. Švejdar. On sequent calculi for intuitionistic propositional logic. *Comment. Math. Univ. Carolin*, 47(1):159–173, 2006.

[27] A.S. Troelstra and H. Schwichtenberg. *Basic proof theory*, volume 43. Cambridge University Press, 2000.