UNIVERSITY OF OSLO
Department of Informatics

Free-Variable Calculi for the Modal Logics K45 and S5

Extended to the Logic of Only Knowing

Master's thesis

Martin G. Skjæveland

18th August 2006

Abstract

This thesis presents a free-variable sequent calculi for the modal logics K45, S5 and the logic of Only Knowing. Labels act as placeholders for points in models, using label variables to postpone the choice of point until more knowledge of a putative satisfying model is gathered, allowing a least commitment search. The relation of contextually equivalents is used to obtain variable-sharing derivations baring tight connections to matrix systems and the goal directed Connection calculus. A system of indexed formulae is employed to enforce reuse of label parameters, establishing an upper bound for the search space. The calculus of the logic of Only Knowing is defined by combining the calculi established for K45 and S5, and utilizing an auxiliary derivation to test models for maximality.

Acknowledgements

First and foremost I thank my supervisor Arild Waaler for providing me with an interesting and demanding problem, and for being helpful and encouraging throughout my work.

Sincere thanks to Roger Antonsen, Espen H. Lian and Morten R. Strand for hints, corrections and for being agile discussion partners.

Lastly I am grateful to all the logicians at the University of Oslo and University of Bergen, where I first was introduced to the field of theoretical logic, for the stimulation learning environment made available to me.

Contents

Chapter 1

Introduction

1.1 The Modal Logics K45 and S5

Propositional modal logic [Blackburn et al. 2005, Hughes and Cresswell 1968, Mints 1992] is syntactically obtained by extending classical propositional logic, adding two unary modal operators $\Box$ (necessarily) and $\Diamond$ (possibly) to the language. Formulae of the form $\Box P$ and $\Diamond P$ are read respectively "necessarily P" and "possibly P". For this reason modal logic is often referred to as the logic of necessity and possibility, a characterization which does not reveal its true capabilities. A more well serving and wide ranging description is the one found in Blackburn et al. [2005]: *"Modal languages are simple yet expressive languages for talking about relational structures."* Relational structures are found almost everywhere: time, knowledge, transitions, networks are all but a few areas which can be modelled using relational structures. Together with the intuitive semantical interpretation accredited to Kripke, they explain the big crowd of supporters modal logic enjoy.

Modal logic forms the basis of a large family of logics. Temporal logic, epistemic logic and doxastic logic are just a few which arise from modal logic by adjusting the interpretation of the two modal operators and the relation on the worlds. Instead of reading $\Box P$ as "necessarily P", other interpretations are "P is *provable*" or "the agent knows/believes that P holds" making modal logics also interesting from a philosophical viewpoint. The expressiveness, together with its low complexity, explains the popularity modal logics have obtained in the computer science community. By using simple modal logics one is capable of expressing rather complex notions as e.g., deadlock, livelock, liveness, fairness, termination and invariance, features exceedingly relevant for any computer program in a time where correctness of computer systems is becoming more and more important. Moreover the

SAT-problem [Garey and Johnsen 1979] of normal modal logics is a member of the set of PSPACE-hard problems [Ladner 1977], a highly favourable membership compared to the undecidable nature of first-order logic.

Of the normal modal logics, this thesis only concerns the modal logics K45 and S5 and calculi for these logics. They form in turn the basis for the epistemic and doxastic logic of Only Knowing ($\mathcal{ONL}$) [Levesque 1990], a logic which is presented in the last chapters of this thesis together with a calculus established by extending the calculus for K45 and S5 to suit $\mathcal{ONL}$. The modal logics K45 and S5 represent some of the simpler modal logics, made apparent by their NP-complete SAT-problem [Garey and Johnsen 1979], but also by how simple some of concepts we will make use of in this thesis are, in contrast to the more elaborate calculi as the ones found in [Beckert and Goré 1997, Massacci 2000] designed to apply to the whole family of normal modal logics.

1.2 The Logic of Only Knowing

The logic of Only Knowing ($\mathcal{ONL}$) [Levesque 1990, Rosati 2001] is an autoepistemic logic, making agents capable of reasoning about their own knowledge and beliefs. For this reason the logic is of special interest to the artificial intelligence community. The logic includes two unary modalities B (belief) and C (co-belief), where $\mathsf{B}P$ is read "the agent believes at least P", while the *co-belief* operator, is easiest expressed by $\mathsf{C}\neg P$ which reads "the agent believes at most P". Combining these operators we obtain the operator "all I know" O-operator originating from Levesque [1990]. $\mathsf{O}P$ is read "the agent believes exactly P" and can be expressed by means of B and C, $(\mathsf{B}P \wedge \mathsf{C}\neg P)$ "the agent believes at least P and at most P". Additionally, the logic has the distinct feature that every possible world is comprehensive by the agent, either by belief or co-belief. Technically this forces a model in $\mathcal{ONL}$ to be *maximal*, meaning that every valuation of propositional letters is represented in the model.

1.3 The LC-calculi

Most logical calculi are instruments testing whether a given set of formulae is valid. The calculi presented in this thesis, the Labelled Calculus—LC for short—inspired by the sequent calculus LK of Gentzen [1934-35], is no different. The method we use is a systematical search comprised by repeated analysis of the input formulae, refining the information for the construction of a putative countermodel, a model bearing witness of the non-validity of the input formulae. This search may end in the discovery that no such model can exist, from which we can conclude that we have a proof of the validity of the input formulae. The expositions emphasis is not on creating a mere calculi for K45 and S5, but constructing an *efficient* calculi ready for implementation in an automated theorem prover for the logics K45, S5 and $\mathcal{ONL}$. To aid such achievements three properties of LC stand out as important in describing the mechanics separating the calculi in this thesis from others.

The first is the use of *labels*.[1] A label is always associated with a formula and names a possible world where the formula is satisfied. The set of labels consists of both *ground labels* and *label variables*. Ground labels pin down a specific point in the set of possible worlds and are used by *existentially quantified* formulae as a witness of their satisfiability. Label variables[2] are used by *universally quantified* formula bearing information that the formula is satisfied in every, if any, possible world. The use of variables in proof search dates back to Prawitz and Kanger [1983] who used "dummies" as placeholders for arbitrary terms. By using label variables the choice of world is postponed until more knowledge of the putative countermodel is gathered through the analysis of the initial input formulae. The use of label variables allows us to be *least committing*, a feature important for the efficiency of a calculus: We do not choose until we have to, lowering the possibility of erroneous choices by reducing both the search space and the nondeterminism destructive for automated reasoning. Specifically, the choice is made by a *unification* of labels. The use of label variables and the resulting efficiency is in stark contrast to the early pioneering destructive systems for modal logics, the labelled system of Fitting [1983], and the ground system by Massacci [2000], where a least committing search strategy is impossible.

Additionally, we introduce the notion of *contextually equivalent* formulae, originally established in [Waaler 2001]. This notion relates formulae in a derivation having been subject to an implicit replication by the rule application to a different formula. Two contextually equivalent formulae are really the same formula occurring in different places in a derivation and are required to abide by a set of conditions making sure that two such related formulae are syntactically equal. These conditions make derivations variable-sharing and the calculi *invariant under rule application* [ibid] or, equivalently, allows for full permutation of inferences in derivations. This quality is crucial for the transformation of LC into the *matrix system* of Wallen [1990]. The matrix system is superior in the compact representation of derivations, and is the reason why we have to keep strict control over the implicitly copied formulae in LC. Matrix systems are in turn applicable

[1] Sometimes called *prefixes* in literature, e.g. by Massacci [2000], Rosati [2001].

[2] Often called *free-variables*.

to the connection calculus [Bibel 1987], a connection driven calculi exploiting the permutability of inferences aiming in its proof search directly at the constructs in a derivation bearing evidence of the proof of the validity of the input formulae. Such search method is called *goal-directed*, the search is to focus on actively "constructing" *connections* (which we will call axioms) and not passively finding them. The connection calculus is thought to be a worthy opponent [Holen 2005, Kreitz and Otten 1999, Otten 1997] to the well-established position that resolution [Robinson 1965] has in the field of automated theorem proving. The problem with resolution and modal logic is that resolution require the input formulae on some normal form, which does not exist for modal logics.

Finally a clever indexing system of formulae is employed inspired by [Wallen 1990]. This system gives us full control of the introduction of labels in derivations easing the enforcement of the conditions concerning contextual equivalence and allows for establishment of upper bounds on the search space granted by an efficient reuse of label parameters. This method can be understood as a variant of the *finite diamond-rule* found in [Beckert and Goré 1997] where the *gödelization* of a formula is introduced as label, resulting in equal formulae introducing equal labels, hence resulting in a recycling of labels. The use of indices in LC makes the calculi readily available for application to the splitting calculus [Antonsen and Waaler 2006], an application whose result would allow for a significant reduction in size of derivation by setting the upper limit of permissible formulae replications of branch-wise and not by considering the whole derivation as we are forced to in LC.

Creating a free-variable calculus invariant under order of rule application for the modal logic K45 carries with it a problem which does not arise in first-order logic, but is obvious to the logicians working with free-logic [Bencivenga 1986], a logic were an empty domain is allowable. The problem is that a universally quantified formula in K45 may quantify over an empty set of points. This means that deducting from "P is satisfied in all points" to "P is satisfied in the point U, where U is a dummy for an arbitrary point" is not in general sound. Our solution is to make the function assigning points to label variables (dummies) defined also when the set of points is empty, allowing us to discover those cases where there are no points for U.

Chapter guide

Chapter 2 establishes the basics of syntax and semantics for the modal logics K45 and S5. Note however that of the normal modal logic we only care for

K45 and S5 and make use of a simplified model suited for these logics.

Chapter 3 provides groundwork for all the calculi in the thesis establishing the syntax and semantics for the language of labelled formulae employed in all LC-calculi, defining the instruments and mechanics of LC and providing the important notions of soundness and completeness.

Chapter 4 is the exposition if the simplest calculus in this thesis, the LC^{g}-calculus. This is a straightforward ground calculus, with one exception, it implements the *contextual equivalence* relation, the relation which all LC-calculi abide by. Soundness and completeness are established using standard methods.

Chapter 5 provides the $\mathsf{LC}^{\mathsf{gt}}$-calculus, a slightly more complicated calculus compared to LC^{g}. The calculus $\mathsf{LC}^{\mathsf{gt}}$ introduces the use of indexed formulae which are used to control the introduction of label parameters by existentially quantified formula. The calculus is defined as an intermediate step towards the realization of the free-variable calculus $\mathsf{LC}^{\mathsf{gt}}$. $\mathsf{LC}^{\mathsf{gt}}$ represents the sound and complete *ground* version of the $\mathsf{LC}^{\mathsf{fv}}$-calculus.

Chapter 6 establishes the free-variable calculus $\mathsf{LC}^{\mathsf{fv}}$ for the modal logics K45 and S5. Throughout this chapter a comparison between the $\mathsf{LC}^{\mathsf{fv}}$- and $\mathsf{LC}^{\mathsf{gt}}$-calculi is made, making the strengths of the free-variable calculus especially apparent. This comparison has it's finale in the syntactical proof sketch of soundness for $\mathsf{LC}^{\mathsf{fv}}$, showing that every $\mathsf{LC}^{\mathsf{fv}}$-proof is transferable to an $\mathsf{LC}^{\mathsf{gt}}$-proof. Semantical proofs of soundness and completeness are also given.

Chapter 7 provides the basics for the logic of Only Knowing.

Chapter 8 represents the *work in progress* of this thesis. the chapter defines $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$, the free-variable calculus for the logic of Only Knowing. It is established by arguing that a compound K45/S5-calculus, with an additional test of the maximality condition of models, are sufficient means for providing an $\mathcal{ONL}$-calculus. Luckily, K45/S5-calculi are all we have available in this thesis, so merging the correct mechanics for a free-variable $\mathcal{ONL}$-calculus is easily established by relying on the results in earlier chapters. However, the time limit has restricted me to only give a sketch of the system making the exposition suffer from a lack of examples and intuitions compared to the chapters on the calculi for K45 and S5. The test for maximality of models is established through a conjecture, by which the proof sketches of both soundness and completeness of $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$ rely.

Reader's guide

The thesis is written such that examples and remarks may be dropped when reading. This is also the reason why many remarks and examples contain a lot of text. The reader comfortable with skipping examples and remarks may do so without missing vital notions.

The reader familiar with modal logic, the logic of Only Knowing and the automated theorem proving notions introduced in the introduction, should be able to grasp the chapters 6 and 8 directly. However important definitions are

- Definition 2.7 on page 10,
- Definition 3.9 on page 18,
- Definition 3.10 on page 19 and
- Definition 3.11 on page 19.

These definitions concern the conceptions of models and the interpretation of labels. I recommend going through these definitions, referring to succeeding examples and remarks if intuitions are desirable, and then reading through the examples 4.18 on page 41, 5.15 on page 57 and 6.24 on page 79 before embarking on the chapters 6 and 8. These three examples are derivations over the same input formula in respectively LC^{g}, $\mathsf{LC}^{\mathsf{gt}}$ and $\mathsf{LC}^{\mathsf{fv}}$. These should provide the reader with insights of the calculi. They also display how the systems handle the case where a universally quantified formula quantifies over a possible empty domain.

The following layout of different textual environments are used:

Definition 1.1
This is a definition. ■

Theorem 1.2
This is a theorem. ■

Lemma 1.3 This is a lemma. ■

Corollary 1.4 This is a corollary. ■

Proof. This is a proof containing a claim and the proof of the claim.

Claim. The claim. ○

Proof. Proof of claim. □

This concludes the proof of Corollary 1.4. □

Example 1.5 This is an example. ◦

Remark. This is a remark. ◦

Scientific acknowledgements

The $\mathsf{LC}^{\mathsf{fv}}_{K45/S5}$-calculi is derived mainly from the systems discussed in [Waaler 2001]. This thesis provides however a clear and instructive exposition on the construction of calculi, by working through the process of a simple ground calculus and iteratively improving and refining to obtain an efficient free-variable calculus. The exposition contains several examples providing the reader with valuable insights on how the different calculi relate, and how the different features of the calculi are obtained.

More specifically the contributions made by this thesis are:

- We establish termination bounds for $\mathsf{LC}^{\mathsf{fv}}$ by employing greater reuse of label parameters than in the systems of [Waaler 2001].
- The semantical proof given of soundness of $\mathsf{LC}^{\mathsf{fv}}$ is not found elsewhere in literature.
- The semantical proof relies partly on the "syntactical point" $\odot$ added to the codomain of the label variable assignment function ρ. This symbol makes an otherwise partial function total, and allows universally quantifies formulae to quantify over an empty domain. I later discovered that Beckert and Goré [1997] uses a similar approach.
- The main idea providing the $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-calculus, namely explicitly testing models for maximality, is found in [Rosati 2001] where a ground system for $\mathcal{ONL}$ is given. I provide a solution to interpret the modalities B and C as alternately K45- and S5-modalities, by requiring axioms to be closed by two different label substitutions. Furthermore, an attempt to apply the AUX-tableau by Rosati [2001] to the $\mathsf{LC}^{\mathsf{fv}}$-calculi is given. This thesis provides a sketch of the first free-variable calculus of the logic of Only Knowing to the author's knowledge.

Chapter 2

Preliminaries on the Modal Logics K45 and S5

2.1 Syntax

Syntax is a set of symbols, the language, and a set of rules that govern how symbols are combined to form new symbolic structures. The *formula* is the fundamental syntactical structure.

Formulae of the modal logics K45 and S5 are defined in the usual way from a countable infinite set of propositional letters **P**, the classical connectives $\neg$ (not), $\rightarrow$ (implies), $\vee$ (or) and $\wedge$ (and), the modal operators $\Box$ (necessarily) and $\Diamond$ (possibly), and the punctuation symbols '(' and ')'. We call this language the *core language* and the formulae in this language *core formulae*.

Definition 2.1 (Core formula)
The set of formulae in the core language is the smallest set Σ such that

1. $\mathbf{P} \subset \Sigma$, where **P** is the set of propositional letters,
2. if $X \in \Sigma$, then $\neg X \in \Sigma$, $\Box X \in \Sigma$ and $\Diamond X \in \Sigma$, and
3. if $X, Y \in \Sigma$, then $(X \rightarrow Y) \in \Sigma$, $(X \vee Y) \in \Sigma$ and $(X \wedge Y) \in \Sigma$.

If $F \in \mathbf{P}$, then F is called *atomic*. F is called *non-atomic* if $F \in \Sigma \setminus \mathbf{P}$. ∎

Example 2.2 (Core formula) The following are examples of core formulae:

- $(P \wedge (Q \vee R))$
- R
- $\Box Q$

- $\Diamond(P \to \Box P)$ ∘

Definition 2.3 (Immediate subformulae)

Let F be a core formula. If F is of the form $\oplus X$, where X is a core formula and $\oplus \in \{\neg, \Box, \Diamond\}$, then X is the *immediate subformula* of F. If F is of the form $(X \oplus Y)$, where X, Y are core formulae and $\oplus \in \{\wedge, \vee, \to\}$, then X and Y are the immediate subformulae of F. Atomic formulae have no immediate subformulae. ■

Example 2.4 (Immediate subformulae) The immediate subformulae of $(P \wedge (Q \vee R))$ are P and $(Q \vee R)$. The immediate subformula of $\Box Q$ is Q and the immediate subformula of $\Diamond(P \to \Box P)$ is $(P \to \Box P)$. ∘

Definition 2.5 (Subformula relations)

Let $<_1$ be a binary relation on formulae defined such that $X <_1 Y$ if and only if Y is an *immediate subformula* of X. Let $<$ be the transitive closure of $<_1$ and call it the *proper subformulae* relation. Let $\leq$, named the *subformulae* relation, denote the reflexive and transitive closure of $<_1$. ■

The *subformulae* of a formula X is the set of formulae Ψ such that $Y \in \Psi$ if and only if $X \leq Y$. The *proper subformulae* of X is $\Psi \setminus \{X\}$.

Example 2.6 (Subformulae) The subformulae of $\Box Q$ are $\{\Box Q, Q\}$ whereas the proper subformulae of $\Box Q$ is the set $\{Q\}$. The subformulae of $\Diamond(P \to \Box P)$ is the set $\{\Diamond(P \to \Box P), (P \to \Box P), P, \Box P\}$. The proper subformulae of $\Diamond(P \to \Box P)$ is the set $\{(P \to \Box P), P, \Box P\}$. ∘

Notice that the definition of different types of subformulae and subformula relations are defined for the general notion of a formula. We will later use these definitions on other kinds of formulae.

2.2 Semantics

Semantics is the mechanism used to give meaning to syntax. How and when can we say that a formula true and when is it false? A *model* is the context in which formulae obtain meaning.

Definition 2.7 (Model)

A K45-model for the core language is a triplet $\mathcal{M} = (W, W^+, V)$ where W is a non-empty set of elements we call *points* or *worlds*, $W^+ \subseteq W$ and $V : \mathbf{P} \to \mathcal{P}(W)$.

An S5-model for the core language is a K45-model $\mathcal{M} = (W, W^+, V)$, where $W^+ = W$. ∎

The set W is called the *domain* of a model and the function V is called a *valuation*. Intuitively $V(P)$ is the set of points in the domain of a model where the propositional letter P is *satisfied*. This function is extended to arbitrary formulae by the *satisfaction relation*.

Definition 2.8 (Satisfaction)
Each model $\mathcal{M} = (W, W^+, V)$ defines a satisfaction relation $\models$ ($\not\models$ denotes its complement) as the weakest relation closed under the following clauses:

$$
\begin{array}{lll}
\mathcal{M}, w \models P & \text{iff} & w \in V(P),\ P \in \mathbf{P}, \\
\mathcal{M}, w \models (X \wedge Y) & \text{iff} & \mathcal{M}, w \models X \text{ and } \mathcal{M}, w \models Y, \\
\mathcal{M}, w \models (X \vee Y) & \text{iff} & \mathcal{M}, w \models X \text{ or } \mathcal{M}, w \models Y, \\
\mathcal{M}, w \models \neg X & \text{iff} & \mathcal{M}, w \not\models X, \\
\mathcal{M}, w \models (X \rightarrow Y) & \text{iff} & \mathcal{M}, w \not\models X \text{ or } \mathcal{M}, w \models Y, \\
\mathcal{M}, w \models \Box X & \text{iff} & \mathcal{M}, w' \models X \text{ for all } w' \in W^+, \\
\mathcal{M}, w \models \Diamond X & \text{iff} & \mathcal{M}, w' \models X \text{ for at least one } w' \in W^+,
\end{array}
$$

where $w \in W$. ∎

Intuitively—and to establish the correct vocabulary—$\mathcal{M}, w \models F$ asserts that the core formula F is *satisfied by* the model $\mathcal{M}$, *in* the point w, which is a member of the domain of $\mathcal{M}$. We say that $\mathcal{M}$ *satisfies* F in w, by which it follows that F is *satisfiable* in w.

Example 2.9 (Satisfaction) Let $\{P, Q, R\}$ be a set of propositional letters, $\mathcal{M} = (W, W^+, V)$ be a K45-model where $W = \{w_1, w_2, w_3, w_4\}$, $W^+ = \{w_2, w_3, w_4\}$, and V be such that $V(P) = \{w_2, w_3\}$, $V(Q) = \{w_1, w_3, w_4\}$ and $V(R) = \{w_1, w_2\}$. See Figure 2.1 for an illustration of this model. We will now test whether the following holds: (1) $\mathcal{M}, w_2 \models R$, (2) $\mathcal{M}, w_3 \models \Box Q$, and (3) $\mathcal{M}, w_1 \models \Diamond(P \rightarrow \Box P)$.

(1) $\mathcal{M}, w_2 \models R$ holds since $w_2 \in V(R)$.

(2) $\mathcal{M}, w_3 \models \Box Q$ does not hold, since $\mathcal{M}, w_3 \models \Box Q$ iff $\mathcal{M}, y \models Q$ for all $y \in W^+$, but $\mathcal{M}, w_2 \not\models Q$.

(3) $\mathcal{M}, w_1 \models \Diamond(P \rightarrow \Box P)$ holds, since
 $\mathcal{M}, w_1 \models \Diamond(P \rightarrow \Box P)$
 iff $\mathcal{M}, x \models (P \rightarrow \Box P)$ for one $x \in W^+$
 iff $\mathcal{M}, x \not\models P$ or $\mathcal{M}, x \models \Box P$, for one $x \in W^+$
 iff $\mathcal{M}, x \not\models P$ for one $x \in W^+$, or; if $W^+ \neq \emptyset$, $\mathcal{M}, y \models P$ for all $y \in W^+$
 The last assertion holds given that $\mathcal{M}, w_2 \not\models P$. The reader should also observe that $\Diamond(P \rightarrow \Box P)$ is satisfiable in every point of the model as

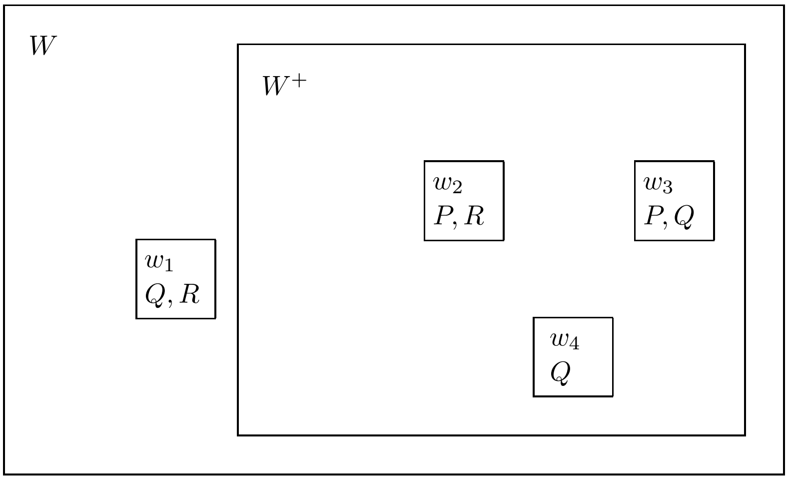


Figure 2.1: Graphical display of the K45-model (W, W^+, V) used in Example 2.9. $W = \{w_1, w_2, w_3, w_4\}$, $W^+ = \{w_2, w_3, w_4\}$ and V is such that $V(P) = \{w_2, w_3\}$, $V(Q) = \{w_1, w_3, w_4\}$ and $V(R) = \{w_1, w_2\}$.

$\mathcal{M}, x \not\models P$ for one $x \in W^+$ holds no matter what point we test for satisfaction in, and that this is sufficient to satisfy the formula. ◦

Remark (to Example 2.9). In above example the statement

$$\mathcal{M}, x \not\models P \text{ or } \mathcal{M}, x \models \Box P, \text{ for one } x \in W^+$$

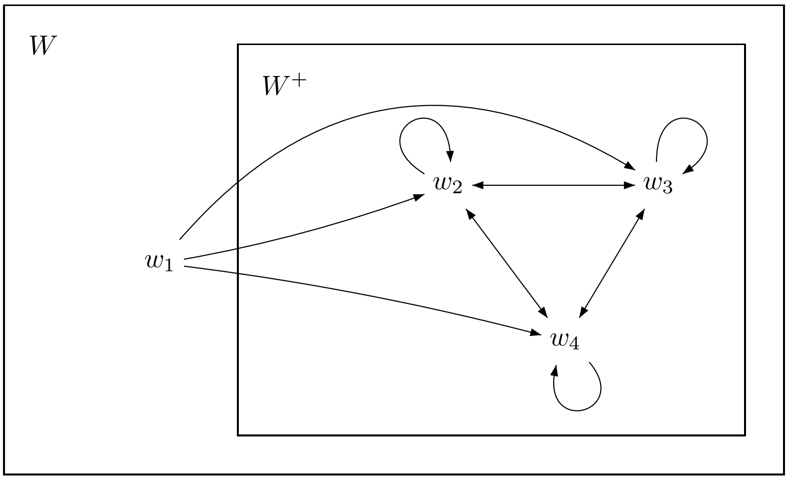
occurs in the dissection of the formula marked (3). This statement may seem as a claim which is always true: "Either there is a point in W^+ which does not satisfy P, or all points in W^+ satisfy P." But this translation misses one important subtlety, "all points in W^+ satisfy P" must be asserted under the assumption that W^+ is non-empty. If we let $W^+ = \emptyset$ then the formula $\Diamond(P \to \Box P)$ is trivially not satisfiable in any model as there are no points in W^+ to satisfy $(P \to \Box P)$. The case $W^+ = \emptyset$ is a special case concerning K45-models and requires special attention throughout the thesis. ◦

The structure of a model by Definition 2.7 may be unfamiliar even to the reader familiar with modal logic. The reason for this relatively simple model is that we only care for the logics K45 and S5, in contrast to the more common definition of a model in model logic which is designed to apply to a wider range of modal logics. To give insights to how our definition relates to the common definition of a model, a discussion follows.

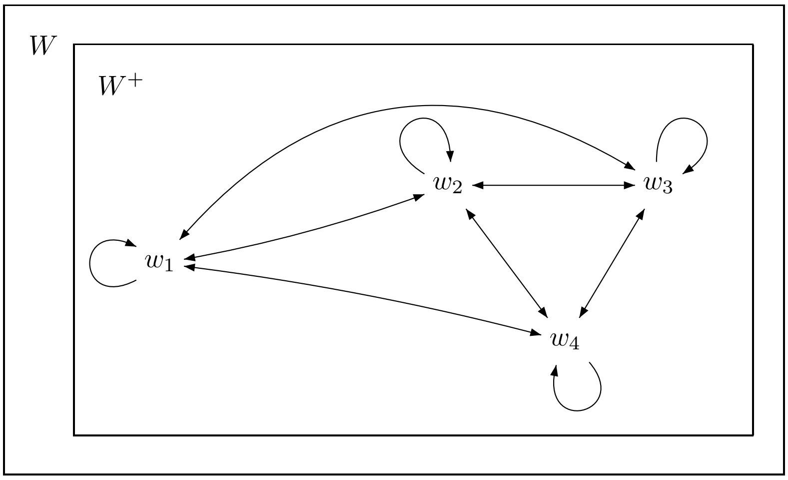
Remark. The common way of defining a model for modal logics is by requiring that a model is $\mathcal{M} = (W, R, V)$, where W and V are as in Definition 2.7; W is a non-empty set and V is a valuation, and R is a binary relation on W [Blackburn et al. 2005]. The relation R is often called the *accessibility* relation and plays a major role in the definition of formulae of the form $\Diamond F$ and $\Box F$. A formula of the form $\Box F$ is satisfied by a model in a point x if every point accessible from x satisfies F, and $\Diamond F$ is satisfiable in x if there is at least one point accessible from x satisfying F. A point y is accessible from x if $(x, y) \in R$. By varying R we obtain different modal logics, e.g., if $(x, x) \in R$ for all $x \in W$, the relation R is reflexive, meaning that every point is related to itself. This modal logic is referred to as the logic **T**. In K45, R is transitive: $\{(x, y), (y, z)\} \subseteq R$ implies $(x, z) \in R$ for all $x, y, z \in W$, and Euclidean: $\{(x, y), (x, z)\} \subseteq R$ implies $(y, z) \in R$ for all x, y, z. In S5 R is reflexive, transitive and Euclidean, which is the same as saying that R is an equivalence relation.

In our definition of model and satisfaction we see that by explicitly giving the set of accessible points we let the set W^+ play the role the relation R has in the above model. In contrast to the accessibility relation where the set of accessible points from two point may differ, the set of accessible points from any point in our model is always the set W^+. The difference between a K45-model and an S5-model by our definition is that in S5 $W = W^+$, while $W^+ \subseteq W$ in K45. What does this mean? We should be able to show that the points in W are related in the same way as they would have been if we had defined our model in the more the common way. Let us adapt the notion of accessibility on the relation the satisfaction relation defines in the cases of formulae of the form $\Box F$ or $\Diamond F$, i.e., all points in W^+ is accessible from every point in W. Let us first look at S5. As $W = W^+$, every point in W is related to every point in W. This is amounts to an equivalence relation. In K45, $W^+ \subseteq W$ implies that the points in $W \setminus W^+$ are not related by the accessibility relation, meaning that we in this case loose reflexivity, but that the relation is still transitive and euclidean.

Consult Figure 2.2. The arrows between points in the models in the figure display the relation we for reasons of instruction temporarily have named the accessibility relation. This relation is not a part of the model but is encoded by the set W^+ and the satisfaction relation on formulae of the form $\Box F$ or $\Diamond F$. In both models every point in W is related to all points in W^+ and no other points. This means that a point in W^+ is related to every point in W^+, including itself. A point in $W \setminus W^+$ is also related to all points in W^+, but not to any point in $W \setminus W^+$. $\circ$



(a) K45-model. $W = \{w_1, w_2, w_3, w_4\}$ and $W^+ = \{w_2, w_3, w_4\}$.



(b) S5-model. $W = W^+ = \{w_1, w_2, w_3, w_4\}$.

Figure 2.2: Both figures display two models over the same domain $W = \{w_1, w_2, w_3, w_4\}$. (a) illustrates a K45-model as $W \subset W^+$ and, (b) shows an S5-model. The outermost rectangle illustrates the set W. W^+ is denoted by the rectangle contained in W. The lowercase letters characterize the elements in the sets and the arrows represent the accessibility relation discussed in the remark on page 13. Although present in every model, these figures do not consider the function V.

Chapter 3

Groundwork on the LC-calculi

The calculi established in this thesis will be called LC-*calculi*. Every calculus will be denoted LC_Y^X where X is used to identify a distinct member of the LC-family and Y characterizes the logic it complies to. The LC-calculi are mainly inspired by the sequent calculus LK (Logische Kalküle) pioneered by Gentzen Gentzen [1934-35] from which also the name LC is inspired. LC is an abbreviation for the *Labelled Calculus*. A *label* is coupled with a formula and is used to identify points in a model satisfying the companionating formula. From this viewpoint the LC-calculi can be seen as a mapping of modal logic into a first-order logic of only unary predicates. This relation to first-order logic lets us adapt already established techniques and mechanics of automated reasoning in first-order logic to the field of modal logics. The idea of labels is by no means new, although perhaps occurring under a different name, see e.g. [Beckert and Goré 1997, Gabbay 1996, Massacci 2000].

The LC-calculi will in this thesis be established for three logics: K45, S5 and the logic of Only Knowing ($\mathcal{ONL}$) [Levesque 1990, Rosati 2001]. The systems for these logics will be very similar, so L will be used as a fusion symbol for 'K45', 'S5' and '$\mathcal{ONL}$'. In the same spirit LC_L is used to prefix general notions meant for application to all LC_L-calculi.

3.1 Syntax

The *labelled language* extends the core language by including the symbols 'ϵ', a countable infinite set of variables denoted Var, a countable infinite set of parameters denoted Par and the punctuation symbols '[' and ']'.

Definition 3.1 (Label)
If $s \in \mathsf{Var} \cup \mathsf{Par} \cup \{\epsilon\}$, then s is called a *label*. If $s \in \mathsf{Par} \cup \{\epsilon\}$, then the label s is called *ground*. ▪

We call Var the set of *label variables*, Par the set of *label parameters* and ϵ *the empty label*. Uppercase letters $U, V, W, \ldots$ are used to denote label variables and label parameters are denoted by lowercase letters $a, b, c, \ldots$. The letters s and t are used to denote arbitrary labels.

Definition 3.2 (Signed formula)
If F is a formula, then $F_\bot$ and $F_\top$ are signed formulae. $\top$ (top) and $\bot$ (bottom) are called *polarities*. The polarity of $F_\bot$ is $\bot$ and the polarity of $F_\top$ is $\top$. A formula with no polarity is called *unsigned*. ▪

Definition 3.3 (Labelled formula)
A *labelled formula* is a signed core formula with a label and where every subformulae of the core formula is assigned a distinct *formula number*. If F is a core formula, s is a label and i is a formula number, then $\underset{i}{F}[s]_\top$ and $\underset{i}{F}[s]_\bot$ are labelled formulae, assuming also the proper subformulae of F are given formula numbers. A labelled formula is called atomic (non-atomic) if its underlying core formula is atomic (non-atomic). ▪

Example 3.4 (Labelled formula) The following are examples of labelled formulae:

- $\underset{1}{\Box}\underset{2}{Q}[U]_\top$
- $\underset{1}{\Diamond}(\underset{3}{P} \underset{5}{\rightarrow} \underset{2}{\Box}\underset{7}{P})[c]_\top$
- $(\underset{4}{P} \underset{2}{\rightarrow} (\underset{3}{Q} \underset{1}{\vee} (\underset{7}{P} \underset{2}{\wedge} \underset{10}{R})))[s]_\top$ ◦

Notation. We often do not display the formula numbers of labelled formulae. The formula numbers exist because of two reasons, preparation for the more complicated calculi in this thesis and because of distinctness and identification of formulae. We want two labelled formulae with equal core formulae, labelling and polarity to coexist in a *set* of formulae as two different elements and we want to be able to uniquely identify them. This is done by assigning different formula numbers to two otherwise syntactically equal formulae. ◦

Every formula in the LC-calculi is a formula with a label and a polarity. (Such a formula is not necessarily a labelled formula. Later we will extend the notion of labelled formulae to indexed formulae.) This means that if not explicitly stated otherwise a formula occurring without a label or a polarity is in fact labelled and does in fact have a polarity, but what these values are not important in the current context, so we may say that P is a labelled formula even though no label or polarity is indicated.

α	α_1	α_2
$(X \wedge Y)[s]_\top$	$X[s]_\top$	$Y[s]_\top$
$(X \vee Y)[s]_\bot$	$X[s]_\bot$	$Y[s]_\bot$
$(X \to Y)[s]_\bot$	$X[s]_\top$	$Y[s]_\bot$
$\neg X[s]_\top$	$X[s]_\bot$	
$\neg X[s]_\bot$	$X[s]_\top$	

β	β_1	β_2
$(X \vee Y)[s]_\top$	$X[s]_\top$	$Y[s]_\top$
$(X \wedge Y)[s]_\bot$	$X[s]_\bot$	$Y[s]_\bot$
$(X \to Y)[s]_\top$	$X[s]_\bot$	$Y[s]_\top$

ν	$\nu_0(t)$
$\Box X[s]_\top$	$X[t]_\top$
$\Diamond X[s]_\bot$	$X[t]_\bot$

π	$\pi_0(t)$
$\Diamond X[s]_\top$	$X[t]_\top$
$\Box X[s]_\bot$	$X[t]_\bot$

Figure 3.1: Labelled formula types and their components. X and Y are arbitrary core formulae with formula numbers.

Definition 3.5 (Type)

Every non-atomic formula is of either type α, β, ν or π, depending on its polarity and outermost connective/modal operator, as defined in Figure 3.1. ∎

Example 3.6 (Type) The following are examples of the type of different formulae. The main connective/modal operator and polarity are underlined to indicate that these features determine the type of the formulae.

- $\underline{\Box}Q[U]_{\underline{\top}}$ is a ν-formula,
- $\underline{\Diamond}(P \to \Box P)[c]_{\underline{\top}}$ is a π-formula, and
- $(P \underline{\to} (Q \vee (P \wedge R)))[s]_{\underline{\top}}$ is of type β. ∘

Remark. Types originate from Smullyan who established the types $\alpha, \beta, \gamma, \delta$ for first-order logic [Smullyan 1968]. They were later extended to modal logic by Fitting [1983], adding the types π and ν. ∘

For each type we define either one or two *components*. If X is an α-formula its components are α_1 and α_2, except the case where X is of the form $\neg Y[s]_\bot$ or $\neg Y[s]_\top$, where Y is some core formula. In this case the α-formula X has α_1 as its single component. The components of a β-formula is β_1 and β_2, while the component of a ν-formula and a π-formula is respectively $\nu_0(t)$ and $\pi_0(t)$, where t is some label. The components of the different types of formulae are defined in Figure 3.1.

Example 3.7 (Components) We use the same formulae as in Example 3.6 above. The component of $\Diamond(P \rightarrow \Box P)[c]_\top$ is $\pi_0 = (P \rightarrow \Box P)[t]_\top$, and $Q[t]_\top$ is the component of $\Box Q[U]_\top$. The components of $(P \rightarrow (Q \vee (P \wedge R)))[s]_\top$ is $\beta_1 = P[s]_\bot$ and $\beta_2 = (Q \vee (P \wedge R))[s]_\top$. ◦

Definition 3.8 (Immediate subformulae)

Let F be a labelled formula. The immediate subformulae of F are the components of F as defined in Figure 3.1. ▪

By using Definition 3.8, the subformula relations defined in Definition 2.5 are easily lifted to labelled formulae, defining the notions of subformulae and proper subformulae also for labelled formulae.

Throughout this thesis we will use the different types as shorthand notation for an arbitrary labelled formulae of the given type. If α is said to be a formula, then α denotes an arbitrary formula of type α.

3.2 Semantics

A label is a placeholder for a possible point or points in the set of worlds where the formula connected to the label is satisfied. This causes the need for some kind of translation of these labels into the set of worlds. The solution is to define a function, a *label interpretation*, mapping labels into the set of points W. This function is composed of two separate functions ρ and ϕ, where ρ, called a *label variable assignment*, interprets label variables, while ϕ, called a *ground label interpretation*, assigns ground labels to points in W.

Definition 3.9 (Label interpretation)

Let $\mathcal{M} = (W, W^+, V)$ be a model for the core language. A label variable assignment ρ is a total function $\rho : \mathsf{Var} \rightarrow W^+ \cup \{\odot\}$, such that $\rho(U) = \odot$ only if $W^+ = \emptyset$. A ground label interpretation ϕ is defined as a total function $\phi : \mathsf{Par} \cup \{\epsilon\} \rightarrow W$ such that $\phi(a) \in W^+$ if $a \in \mathsf{Par}$. ϕ_ρ is then a *label interpretation* $\phi_\rho = \phi \cup \rho$, i.e., $\phi_\rho(a) = \phi(a)$ for every $a \in \mathsf{Par} \cup \{\epsilon\}$ and $\phi_\rho(U) = \rho(U)$ for every $U \in \mathsf{Var}$. ▪

Remark. The definition of a label interpretation is compact and not very transparent, so we make some instructive observations. Notice that every label parameter is mapped to a point in W^+, making ϵ is the only label which can be mapped to a point in $W \setminus W^+$, a set which can be non-empty only if the model in question is a K45-model. In S5 ϵ is necessarily mapped to W^+ by ϕ, since $W = W^+$ in every S5-model.

The reason for adding the symbol $\odot$ to the codomain of the variable assignment ρ is not obvious. The role of $\odot$ in the definition to ensure that ρ is always defined for all legal input. The need for this is because of the special case of W^+ being empty in a model $\mathcal{M} = (W, W^+, V)$. If the label variable assignment was defined as $\rho : \mathsf{Var} \rightarrow W^+$ and $W^+ = \emptyset$, then $\rho(U)$ would in this case be undefined for any label variable U. The incorporation of $\odot$ is a way of making an otherwise partial function total, and hence a way of being able to address the case of W^+ being empty. ◦

Definition 3.10 (*L*-Model)

An L-model for the labelled language is a pair $(\mathcal{M}, \phi)$, where $\mathcal{M} = (W, W^+, V)$ is an L-model for the core language and ϕ is a ground label interpretation defined relative to $\mathcal{M}$. ∎

Truth and *satisfaction* of labelled formulae are defined relative to a label variable assignment. This allows us to pass judgement on all labelled formulae, including formula labelled with label variables.

Definition 3.11 (Truth)

Let $\vDash$ be a *satisfaction relation* defined on labelled formulae relative to an L-model for the labelled language. The symbol $\nvDash$ denotes its complement. Let $(\mathcal{M}, \phi)$ be an L-model for the labelled language and ρ some label variable assignment.

Let $F[s]$ be an arbitrary unsigned labelled formula. $\underset{i}{F}[s]_\top$ is *satisfied* in $(\mathcal{M}, \phi)$ under ρ, written $(\mathcal{M}, \phi) \vDash_\rho \underset{i}{F}[s]_\top$, if and only if $\mathcal{M}, \phi_\rho(s) \vDash F$ or $\phi_\rho(s) = \odot$. $\underset{i}{F}[s]_\bot$ is satisfied in $(\mathcal{M}, \phi)$ under ρ, written $(\mathcal{M}, \phi) \vDash_\rho \underset{i}{F}[s]_\bot$, if and only if $\mathcal{M}, \phi_\rho(s) \nvDash F$ or $\phi_\rho(s) = \odot$. We say that a formula is *satisfiable* if it is satisfied in some L-model under some label variable assignment.

A set of formulae Γ is satisfied in $(\mathcal{M}, \phi)$ under ρ, denoted $(\mathcal{M}, \phi) \vDash_\rho \Gamma$, if every formula in the set is satisfied in $(\mathcal{M}, \phi)$ under ρ. We say that Γ is satisfiable if there is an L-model and a label variable assignment that satisfies it.

Let F now be an arbitrary labelled formula. We write $(\mathcal{M}, \phi) \vDash F$ and say that F is *true* in $(\mathcal{M}, \phi)$, provided $(\mathcal{M}, \phi) \vDash_\rho F$ for every ρ.

A set of formulae Γ is true in $(\mathcal{M}, \phi)$, denoted $(\mathcal{M}, \phi) \vDash \Gamma$, if every formula in the set is true in $(\mathcal{M}, \phi)$. ∎

Remark. Note that we overload the symbol $\vDash$. When we say that $(\mathcal{M}, \phi) \vDash \underset{i}{F}[s]_\top$ (1) if and only if $\mathcal{M}, \phi(s) \vDash F$ (2), the relation in (1) is a relation on labelled formulae and L-models, and is defined in Definition 3.11, while (2) contains the relation defined in Definition 2.8.

Observe also that formula numbers of labelled formulae are not taken under consideration when discussing truth:

$$(\mathcal{M}, \phi) \models \underset{i}{F}[s]_{\top} \quad \Leftrightarrow \quad \mathcal{M}, \phi(s) \models F \quad \Leftrightarrow \quad (\mathcal{M}, \phi) \models \underset{j}{F}[s]_{\top}$$

for all i, j. ◦

Example 3.12 (L-Model) Let $(\mathcal{M}, \phi)$ be an L-model let $\mathcal{M} = (W, W^+, V)$ be the model in Example 2.9. Now we test if $(\mathcal{M}, \phi)$ satisfies $\Diamond(P \to \Box P)[c]_{\top}$. Even though we have not specified ϕ, we can conclude that the formula is satisfied in $(\mathcal{M}, \phi)$. Since we know from Example 2.9 that $\mathcal{M}, x \models \Diamond(P \to \Box P)$ for any $x \in W$, then no matter what point $\phi(c)$ denotes the labelled formula is satisfied in the model $(\mathcal{M}, \phi)$.

Next test if $(P \wedge \neg P)[U]_{\top}$ is satisfiable. Immediately the formula looks as though it is not satisfiable, $P \wedge \neg P$ is perhaps the simplest contradiction available in classical propositional logic, but let us not jump to any conclusions yet. Let $(\mathcal{M}, \phi)$ be some L-model, $\mathcal{M} = (W, W^+, V)$ and ρ some label variable assignment. First assume $\rho(U) = w \in W^+$, then $(\mathcal{M}, \phi) \models_{\rho} (P \wedge \neg P)[U]_{\top}$ does not hold, since $\mathcal{M}, w \models P \wedge \neg P$ obviously cannot be true as both $\mathcal{M}, w \models P$ and $\mathcal{M}, w \models \neg P$ cannot hold. But now assume $W^+ = \emptyset$—again the special case of K45—then $\rho(U) = \odot$ by Definition 3.9, and by Definition 3.11 the formula is satisfiable. ◦

In some calculi we restrict the set of labels in a language to only Par and ϵ, i.e., the set of ground labels, thus interpretation of label variables is not necessary. Observe that in such restricted languages a labelled formula is either 'true' in an L-model or 'not satisfied'.

The reason for classifying formulae into types and identifying their components is because different formula occurrences of the same type adhere to the same behaviour—as Lemma 3.15 will show. Before we commence on this lemma and its proof we need to establish the notion of an *extension* of a function and a result concerning the extension of ground label interpretations in relation to the same L-model in the core language, as we need this result in Lemma 3.15.

Definition 3.13 (Extension)

A function f' is an extension of the function f if $f'(x) = f(x)$ for every x in the domain of f. f' is an *extension by c* if only possibly c is added to the domain of f. ∎

Notice that by Definition 3.13 every function is its own extension.

Lemma 3.14 If ϕ' is an extension of a ground label interpretation ϕ, then $(\mathcal{M}, \phi) \models_\rho \Gamma$ only if $(\mathcal{M}, \phi') \models_\rho \Gamma$, where $(\mathcal{M}, \phi)$ is an L-model, ρ a label variable assignment and Γ a set of formulae. ■

Proof. Assume $(\mathcal{M}, \phi) \models_\rho \Gamma$, then $(\mathcal{M}, \phi) \models_\rho F[s]$ for all $F[s] \in \Gamma$ and by definition of satisfaction $\mathcal{M}, \phi_\rho(s) \models F$ for all $F[s]_\top \in \Gamma$ and $\mathcal{M}, \phi_\rho(s) \not\models F$ for all $F[s]_\perp \in \Gamma$. Since ϕ' is an extension of ϕ, $\mathcal{M}, \phi'_\rho(s) \models F$ for all $F[s]_\top \in \Gamma$ and $\mathcal{M}, \phi'_\rho(s) \not\models F$ for all $F[s]_\perp \in \Gamma$ by which it follows that $(\mathcal{M}, \phi') \models_\rho F[s]$ for all $F[s] \in \Gamma$ and $(\mathcal{M}, \phi') \models_\rho \Gamma$. □

As we have seen in the definition of a label interpretation and the observations given following it, the symbol ϵ plays different roles in the two logics K45 and S5. It is therefore convenient to define a set of labels relative to the logic L. Let Par_L denote a set of ground labels such that $\mathsf{Par}_{K45} = \mathsf{Par}$ and $\mathsf{Par}_{S5} = \mathsf{Par} \cup \{\epsilon\}$.

Lemma 3.15 (Satisfaction of components) Let Γ be a set of labelled formulae, $(\mathcal{M}, \phi)$ an L-model where $\mathcal{M} = (W, W^+, V)$, and ρ some variable assignment. For all α-, β-, ν- and π-formulae, and their respective components:

1. if $\alpha \in \Gamma$, then $\Gamma \cup \{\alpha_1, \alpha_2\}$ is satisfiable in $(\mathcal{M}, \phi)$ under ρ, iff Γ is satisfiable in $(\mathcal{M}, \phi)$ under ρ,
2. if $\beta \in \Gamma$, then $\Gamma \cup \{\beta_1\}$ is satisfiable in $(\mathcal{M}, \phi)$ under ρ or $\Gamma \cup \{\beta_2\}$ is satisfiable in $(\mathcal{M}, \phi)$ under ρ, iff Γ is satisfiable in $(\mathcal{M}, \phi)$ under ρ,

3.1. if $\nu \in \Gamma$, $W^+ \neq \emptyset$ and Γ is satisfiable in $(\mathcal{M}, \phi)$ under ρ, then $\Gamma \cup \{\nu_0(s)\}$ is satisfiable in $(\mathcal{M}, \phi)$ under ρ for every label $s \in \mathsf{Par}_L$,

3.2. if $\nu \in \Gamma$ and Γ is satisfiable in $(\mathcal{M}, \phi)$ under ρ, then $\Gamma \cup \{\nu_0(U)\}$ is satisfiable in $(\mathcal{M}, \phi)$ under ρ for every $U \in \mathsf{Var}$, and

4. if $\pi \in \Gamma$, then $\Gamma \cup \{\pi_0(c)\}$, where c is an arbitrary label parameter not occurring in Γ, is satisfiable in $(\mathcal{M}, \phi')$ under ρ for some extension ϕ' by c of ϕ, iff Γ is satisfiable in $(\mathcal{M}, \phi)$ under ρ. ■

Proof. There is one case for each type and one subcase for each of the different forms of formulae for the respective types. Only one subcase for each type is proven here, the remaining cases are easily obtained by following the same pattern as the provided subcase, and are left for the reader to establish.

Let the L-model $(\mathcal{M}, \phi)$, where $\mathcal{M} = (W, W^+, V)$, satisfy Γ under the label variable assignment ρ.

1. Assume that $\alpha \in \Gamma$ and that α is of the form $(X \wedge Y)[s]_\top$.

$$
\begin{aligned}
(\mathcal{M},\phi) \vDash_\rho \alpha \quad &\Leftrightarrow \quad (\mathcal{M},\phi) \vDash_\rho (X \wedge Y)[s]_\top \\
&\Leftrightarrow \quad \mathcal{M},\phi_\rho(s) \vDash (X \wedge Y) && \text{(Def. 3.11)} \\
&\Leftrightarrow \quad \mathcal{M},\phi_\rho(s) \vDash X \text{ and } \mathcal{M},\phi_\rho(s) \vDash Y, && \text{(Def. 2.8)} \\
&\Leftrightarrow \quad (\mathcal{M},\phi) \vDash_\rho X[s]_\top \text{ and } (\mathcal{M},\phi) \vDash_\rho Y[s]_\top && \text{(Def. 3.11)} \\
&\Leftrightarrow \quad (\mathcal{M},\phi) \vDash_\rho \alpha_1 \text{ and } (\mathcal{M},\phi) \vDash_\rho \alpha_2
\end{aligned}
$$

Since $(\mathcal{M},\phi) \vDash_\rho \alpha$ if and only if $(\mathcal{M},\phi) \vDash_\rho \alpha_1$ and $(\mathcal{M},\phi) \vDash_\rho \alpha_2$, the set $\Gamma \cup \{\alpha_1, \alpha_2\}$ is satisfiable in $(\mathcal{M},\phi)$ under ρ if and only if Γ is satisfiable in $(\mathcal{M},\phi)$ under ρ.

2. Assume that $\beta \in \Gamma$ and that β is of the form $(X \vee Y)[s]_\top$.

$$
\begin{aligned}
(\mathcal{M},\phi) \vDash_\rho \beta \quad &\Leftrightarrow \quad (\mathcal{M},\phi) \vDash_\rho (X \vee Y)[s]_\top \\
&\Leftrightarrow \quad \mathcal{M},\phi_\rho(s) \vDash (X \vee Y) && \text{(Def. 3.11)} \\
&\Leftrightarrow \quad \mathcal{M},\phi_\rho(s) \vDash X \text{ or } \mathcal{M},\phi_\rho(s) \vDash Y && \text{(Def. 2.8)} \\
&\Leftrightarrow \quad (\mathcal{M},\phi) \vDash_\rho X[s]_\top \text{ or } (\mathcal{M},\phi) \vDash_\rho Y[s]_\top && \text{(Def. 3.11)} \\
&\Leftrightarrow \quad (\mathcal{M},\phi) \vDash_\rho \beta_1 \text{ or } (\mathcal{M},\phi) \vDash_\rho \beta_2
\end{aligned}
$$

Since $(\mathcal{M},\phi) \vDash_\rho \beta$ if and only if $(\mathcal{M},\phi) \vDash_\rho \beta_1$ or $(\mathcal{M},\phi) \vDash_\rho \beta_2$, then $\Gamma \cup \{\beta_1\}$ or $\Gamma \cup \{\beta_2\}$ must be satisfiable in $(\mathcal{M},\phi)$ under ρ if and only if Γ is satisfiable in $(\mathcal{M},\phi)$ under ρ.

3. Assume that $\nu \in \Gamma$ and that ν is of the form $\Box X[t]_\top$

First we let the set W^+ be non-empty, and prove the claims 3.1 and 3.2 under this assumption. Secondly, to complete the proof, we assume that $W^+ = \emptyset$ and prove the claim 3.2 under this assumption.

First, let $W^+ \neq \emptyset$.

$$
\begin{aligned}
(\mathcal{M},\phi) \vDash_\rho \nu \quad &\Leftrightarrow \quad (\mathcal{M},\phi) \vDash_\rho \Box X[t]_\top \\
&\Leftrightarrow \quad \mathcal{M},\phi_\rho(t) \vDash \Box X && \text{(Def. 3.11)} \\
&\Rightarrow \quad \mathcal{M},x \vDash X \text{ for all } x \in W^+ && \text{(Def. 2.8)}
\end{aligned}
$$

Since $\phi'_{\rho'}(s) \in W^+$ for all $s \in \mathsf{Var} \cup \mathsf{Par}_L$ and every label interpretation $\phi'_{\rho'}$, $\mathcal{M},\phi_\rho(s) \vDash X$ must hold for arbitrary $s \in \mathsf{Var} \cup \mathsf{Par}_L$. By definition of truth, $(\mathcal{M},\phi) \vDash_\rho X[s]_\top$ for all $s \in \mathsf{Par}_L$, and $(\mathcal{M},\phi) \vDash_\rho \nu_0(s)$ for all $s \in \mathsf{Par}_L$.

Given that $(\mathcal{M},\phi) \vDash_\rho \nu \Rightarrow (\mathcal{M},\phi) \vDash_\rho \nu_0(s)$ for arbitrary label variable $s \in \mathsf{Var} \cup \mathsf{Par}_L$ when $W^+ \neq \emptyset$, and $(\mathcal{M},\phi) \vDash_\rho \Gamma$ by assumption, then $\Gamma \cup \{\nu_0(s)\}$ is satisfiable in the L-model $(\mathcal{M},\phi)$ under ρ for arbitrary $s \in \mathsf{Var} \cup \mathsf{Par}_L$, given a non-empty W^+.

Remark. The claim $\mathcal{M},\phi_\rho(s) \vDash X$ does not hold when W^+ is empty for any label s; there are no points in W^+ to satisfy the formula X, and moreover in the case of $s \in \mathsf{Var}$ in which $\rho(s) = \odot$, $\odot$ is no point in W and hence $\mathcal{M},\odot$ is deliberately not defined for the satisfaction relation on core formula. On the other hand $\mathcal{M}, \phi_\rho(s) \models \Box X$ holds, since $\mathcal{M}, x \models X$ for all $x \in W^+$ holds trivially if $W^+ = \emptyset$. $\circ$

Now assume that $W^+ = \emptyset$ and direct attention to case 3.2. This is a special case given custom care by the definition of the satisfaction relation (Definition 3.11). $\nu_0(U)$ is satisfied in $(\mathcal{M}, \phi)$ under ρ for arbitrary U by the fact that $\phi_\rho(U) = \odot$ when $W^+ = \emptyset$. Thus, by Definition 3.11, $\nu_0(U)$ is satisfied in $(\mathcal{M}, \phi)$ under ρ for any $U \in \mathsf{Var}$.

4. Assume that $\pi \in \Gamma$ and that π is of the form $\Diamond X[s]_\top$.

$$\begin{array}{rcll} (\mathcal{M}, \phi) \models_\rho \pi & \Leftrightarrow & (\mathcal{M}, \phi) \models_\rho \Diamond X[s]_\top & \\ & \Leftrightarrow & \mathcal{M}, \phi_\rho(s) \models \Diamond X & \text{(Def. 3.11)} \\ & \Leftrightarrow & \mathcal{M}, y \models X \text{ for some } y \in W^+ & \text{(Def. 2.8)} \end{array}$$

Now we extend the language with a *fresh* label parameter c, i.e., we add c which does not occur in Γ to the language, and construct a new L-model $(\mathcal{M}, \phi')$ by requiring that $\phi'(c) = y$ and $\phi'(s) = \phi(s)$ if $s \neq c$. The point y is the *witness* to the satisfaction of the π-formula. By assumption, such a witness must exist since the π-formula is satisfied. Now by construction of ϕ' in the newly created L-model $(\mathcal{M}, \phi')$, the π_0-formula is also satisfied, since $\phi'(c)$ captures the existence of the point $y \in W^+$. Given that $(\mathcal{M}, \phi)$ agrees with $(\mathcal{M}, \phi')$ on all the formulae in Γ, $(\mathcal{M}, \phi')$ satisfies the set $\Gamma \cup \{X[c]_\top\}$. We conclude that $\Gamma \cup \{\pi_0(c)\}$ is satisfied in $(\mathcal{M}, \phi')$ under ρ for a label variable c fresh to Γ, if and only if Γ is satisfiable in $(\mathcal{M}, \phi)$ under ρ. $\Box$

We say that α-formulae act *conjunctively* and β-formulae behave *disjunctively*, as Lemma 3.15 shows. ν-formulae act *universally* and π-formulae are said to behave *existentially*. For the reader familiar with first-order logic, and not modal logic, it can be fruitful to think of ν-formulae as γ-formulae and π-formulae as δ-formulae [Smullyan 1968].

The elaborate claim for the case of ν-formulae in the previous Lemma displays one of greatest difficulties the LC-calculi have to overcome. $(\mathcal{M}, \phi) \models \nu$ holds trivially if W^+ is empty, while $(\mathcal{M}, \phi) \models \nu(s)$ for any ground label s will not hold exactly because there are no points $\phi(s)$ in W^+ to satisfy the formula. In free variable calculus for first-order logic, see e.g., [Fitting 1996], this problem does not arise since the definition of a model in first-order logic requires a non-empty domain. But as in many free variable calculi for first-order logic [Fitting 1996] and modal logic [Beckert and Goré 1997] we want to be able to make use of variables in universally quantified formulae as the ν-formulae are. This explains the existence of the symbol $\odot$ which only comes into play when W^+ is empty: We want ν to be satisfiable if and only if $\nu_0(U)$ is satisfiable, for any label variable U, and as we see of the proof of claim 3.2 of the Lemma, this is made possible by the definition of the interpretation function ρ and the satisfaction relation on labelled formulae when W^+ is empty.

3.3 Calculus

A calculus is a system of syntactic manipulation of formulae and sequents. Through syntax we are able to obtain semantic knowledge of the analysed formulae. The link between syntax and semantics is established by the notions of *soundness* and *completeness* of the calculus, defined at the end of this chapter.

The exposition in this section is an introduction to the core elements of the LC-calculi and also serves as a general roadmap for the expositions of the calculi to come.

The smallest building block in the calculi is the sequent.

Definition 3.16 (Sequent)
A sequent is a set of formulae. ∎

We will frequently use Γ as notation for a sequent. When writing sequents the curly brackets will be dropped. Γ, F, where F is a formula, will be used as shorthand for $\Gamma \cup \{F\}$.

The syntactic manipulation or analysis of sequents is defined through the notion of an *inference*.

Definition 3.17 (Inference)
A set of inferences is a set $\mathcal{I}$ such that $\mathcal{I} \subseteq (\mathcal{S} \times \mathcal{S}) \cup (\mathcal{S} \times (\mathcal{S} \times \mathcal{S}))$ where $\mathcal{S}$ is the set of sequents. ∎

Typically we only want a few inferences to be admissible in a given calculus. Allowing every inference to be applicable at any time would clearly produce an incorrect calculus. The set of permissible inferences is specified through a set of *inference schemata*.

Definition 3.18 (Inference schema)
An inference schema specifies a set of inferences $\mathcal{R} \subseteq \mathcal{I}$, where $\mathcal{I}$ is the set of inferences, by indicating on what form the inferences $\mathcal{R}$ must adhere to. An inference schema is on one of the following forms

$$\langle \Gamma \cup F, \Gamma \cup \Psi \rangle \quad \text{or} \quad \langle \Gamma \cup F, \langle \Gamma \cup \Psi, \Gamma \cup \Psi' \rangle \rangle,$$

where Γ is an arbitrary set of formulae, Ψ and Ψ' are non-empty sets of formulae and F is a formula. Only the form, i.e., type, labelling and numbering, of the formulae in Ψ, Ψ' and F are specified, making Γ a placeholder for any set of formulae, while the formulae in Ψ, Ψ' and F have to abide by the form specification given by the schema. ∎

A set of inference schemata will also be called the *rules of a calculus*—or just *rules*, as the schemata will be the constructs governing the syntactic manipulation in a calculus. In the same spirit an inference may also be called a *rule application*.

Notation. The inference schemata $\langle \Gamma \cup F, \Gamma \cup \Psi \rangle$ and $\langle \Gamma \cup F, \langle \Gamma \cup \Psi, \Gamma \cup \Psi' \rangle \rangle$ will be written

$$\frac{\Gamma, \Psi}{\Gamma, F} \qquad \text{and} \qquad \frac{\Gamma, \Psi \quad \Gamma, \Psi'}{\Gamma, F}$$

respectively. ∘

The two parts of an inference schema divided by the horizontal line are the *numerator*, placed above the line, and the *denominator* situated below. The formula F occurring in the denominator is called the *principal formula*. Ψ and Ψ' are called the set of *active formulae*. All other formulae in the inference schemata, denoted by Γ, are called *extra formulae*. The notions of principal, active and extra formulae, and numerator and denominator will be used in the same way concerning inferences as when addressing inference schemata.

We say that an inference is *applied* to a denominator and a principal formula *resulting* respectfully in a numerator and active formulae. If an inference results in the *introduction* of a label in the numerator, the introduced label will be introduced by an active formula and is said to *belong* to its inference. This is emphasised by marking the horizontal line to which the inference corresponds with the introduced label. There will be exactly one calculus rule per type of formula in the calculi in this thesis, so we name a rule by the type of its principal formula and call it e.g. the α-rule. Also, an inference is said to be of the same type as its principal formula, and will e.g. be called an α-inference. When discussing the relationship between formulae in the denominator and numerator it is natural to view the occurrence of extra formulae in an inference as an *implicit copy* of formulae from the denominator to the numerator as the extra formulae are left unchanged by the inference. If the same formula occurs as both principal and an active formula in an inference, we say that the formula is *explicitly copied* by this inference.

A *skeleton* is a tree of sequents bearing close relations to the LK-derivation, the proof object of the calculus LK, but similar also to the proof object of tableaux systems [Beckert and Goré 1997, Fitting 1983]. The construction of a skeleton is regulated by the inferences applied to the sequents in the skeleton.

Definition 3.19 (Skeleton)
A skeleton is a finitely branching, but possibly infinite, tree where all nodes are sequents regulated by the rules of a calculus. ▪

A skeleton has its name because it may not carry any logical force. It is just a tree of sequents. We will later define structures making skeletons conform to certain rules and conditions.

3.4 Soundness and Completeness

Soundness and completeness are two core notions in the development of every calculus. To establish these notions we need first to build a greater vocabulary than what is currently available.

Definition 3.20 (*L*-countermodel)
Let $(\mathcal{M}, \phi)$ be an L-model.

- $(\mathcal{M}, \phi)$ is an L-countermodel for a sequent Γ if $(\mathcal{M}, \phi) \models \Gamma$.
- $(\mathcal{M}, \phi)$ is an L-countermodel for the branch θ in skeleton if $(\mathcal{M}, \phi)$ is an L-countermodel for every sequent occurring on θ.

  A branch θ is *satisfied* in (M, ϕ) under a label variable assignment ρ if $(\mathcal{M}, \phi) \models_\rho \Gamma$ holds for all sequents in θ.
- $(\mathcal{M}, \phi)$ is an L-countermodel for a skeleton δ, if $(\mathcal{M}, \phi)$ satisfies a branch in δ for every label variable assignment ρ. ▪

Remark. The reader familiar with some logic may find this definition strange; An L-model is an L-countermodel for a sequent if every formulae in the sequent is true?! The reason for this peculiarity is the use of polarity and the definition of satisfiability of formulae with polarity $\bot$. If we write a sequent using "standard" notation $\Gamma \vdash \Delta$, then Γ would denote the formula in the sequent having polarity $\top$ and Δ the formulae with polarity $\bot$. A countermodel for $\Gamma \vdash \Delta$ is a model satisfying every formulae in Γ and falsifying every formulae in Δ. By how we have formulated satisfaction of labelled formulae, this amounts exactly to the definition above. ◦

Definition 3.21 (*L*-Valid)
A sequent is L-valid if it has no L-countermodel. ▪

Through application of inferences defined by the rules of the calculus we construct a skeleton. A skeleton is a merely a tree of sequents. In order to apply some structure and guarantee that a skeleton conforms to the rules of the calculus, and also to be able to provide extra conditions on a skeleton in a particular calculus, we define the notion of a LC_L*-derivation*.

Definition 3.22 (LC_L-derivation)
A skeleton is an LC_L-derivation if it conforms to the conditions set by the calculus LC_L. ■

Apart from defining a set of rules, a calculus also needs to define the requirements for a *proof* in the calculus—what conditions needs to be met in order to call a derivation a proof. This requirement is called the *closure condition* of the calculus and is in all calculi in this thesis defined through the notion of an *axiom*.

Definition 3.23 (Axiom, LC_L)
An axiom in the calculus LC_L is a sequent of a specified form. ■

Definition 3.24 (LC_L-proof)
An LC_L-derivation δ is a LC_L*-proof* if every branch of δ contains an axiom. ■

If a branch θ in an LC_L-derivation contains a sequent categorized as an axiom in the LC_L-calculus, we say that θ is *closed*. If a branch is not closed, it is *open*.

A sequent Γ is said to beLC_L*-provable* if there exists a LC_L-proof δ where Γ is the root sequent of δ. We say that δ is a *proof* of Γ.

Now we are ready to define soundness and completeness of a calculus.

Definition 3.25 (Soundness)
A calculus LC_L is *sound* if every LC_L-provable sequent is L-valid. ■

Definition 3.26 (Completeness)
A calculus LC_L is *complete* if every L-valid sequent is LC_L-provable. ■

To give more meaning to these important notions before commencing on the first calculus in this thesis, a remark containing an informal discussion of soundness and completeness follows.

Remark. A calculus is sound if it does not do anything wrong. A complete calculus is one that inhibits the strength to do everything which is right. As extreme examples a calculus which can prove nothing is sound and a calculus which proves every sequent is complete. What we realize is that a sound *and* complete calculus is a system with the correct balance, it does nothing wrong and it can do everything which is right. An instructive "real world" example can be found in [Walicki 2006].

When designing calculi one has to be careful to add enough restrictions to the system to make it sound. On the other hand if the system becomes too restrictive, loss of completeness is at stake. From an automated reasoning viewpoint, from which this thesis should be read, concern is put into relaxing the restrictions as much as possible, arranging for quick and efficient reasoning without destructing soundness. One is also preoccupied establishing upper bounds on the size of search space, as this allows for early termination of the search. As restriction of search space may lead to the inability of finding countermodels, any carelessness here may result in lack of completeness. ◦

Chapter 4

The Calculus LC^{g}

The LC^{g}-calculus is the first calculus presented in this thesis and is also the simplest. The simplicity is obtained by the fact that the calculus is *ground*—hence the superscript g in LC^{g}—the calculus utilizes only ground labels in contrast to a free-variable calculus where also label variables are employed. The groundedness allows for a more direct connection of syntax and semantics making the arguments necessary to establish soundness and completeness of the calculus easier than if label variables were present. Also, by including this exposition of a completely independent ground calculus, the mechanics of the other LC-calculi in this thesis should appear more transparent, as many of the constructs introduced in this chapter are later reused in a more complex form.

Remark. The LC^{g}-calculi is essentially a ground version of the system established for K45 and S5 in [Waaler 2001], so many of the concepts used in this chapter can be found there. The main focus of this chapter is to explore a simple calculus implementing the *contextually equivalent* relation. The exposition is rich in examples illustrating this condition, the lack of invariance of rule application order in a ground system, and several derivations and proofs in LC^{g}. Thorough proofs of soundness and completeness are established. Results on which the chapters to come rely. ◦

We start by defining a sequent of labelled formulae, the *labelled sequent*.

Definition 4.1 (Labelled sequent)

A labelled sequent is a set of labelled formulae. A labelled sequent is called *ground* if every label in the sequent is ground, and it is called *empty labelled* if every formula in the sequent is labelled with ϵ and every formula number occurring in the sequent is distinct. If a formula in a labelled sequent appears unlabelled, the formula is by convention labelled with ϵ. ▪

Example 4.2 (Sequent) The following is a labelled sequent:

$$P[s]_\top, (Q \rightarrow P)[U]_\top, R[t]_\bot, P[s]_\bot, R[V]_\bot, S_\bot.$$

The apparently unlabelled formula $S_\bot$ is labelled with ϵ, giving $S[\epsilon]_\bot$. The sequent is not ground, as it contains the label variables U and V, and it is not empty labelled since ϵ is not the label of every formulae in the sequent. ◦

Notation. To reduce notation and increase readability we will in the remaining examples use the standard notation, $\Gamma \vdash \Delta$, when displaying sequents. $\Gamma \vdash \Delta$ denotes the sequent $\Gamma \cup \Delta$, where Γ is the set of all formulae in the sequent having polarity $\top$ and Δ is the set of formulae having polarity $\bot$. Because of this we do not display the polarity of formulae in sequents. Also, the "unnecessary" outermost parentheses are dropped whenever possible. Using this notation the sequent in above example 4.2 would be

$$P[s], Q \rightarrow P[U] \vdash R[t], P[s], R[V], S. \qquad \circ$$

The rules of LC^g are listed in Figure 4.1. α, β, ν, π and their respective components (see Figure 3.1) are all labelled formulae of the respective types. These rules apply to both the K45 and S5, but as we will see, different side conditions apply to the two logics. Notice that no rule manipulates formula numbers.

$$\frac{\Gamma, \alpha_1, \alpha_2}{\Gamma, \alpha} \qquad \frac{\Gamma, \beta_1 \quad \Gamma, \beta_2}{\Gamma, \beta} \qquad \frac{\Gamma, \underset{i}{\nu}, \nu_0(s)}{\Gamma, \underset{i}{\nu}}\ s \qquad \frac{\Gamma, \pi_0(c)}{\Gamma, \pi}\ c$$

Figure 4.1: Rules of LC^g. The rules left to right: α-, β-, ν- and π-rule. Γ is a set of labelled formulae. The formula numbers of the principal and active ν-formulae are displayed to indicate that the ν-formula is explicitly copied such that also the formula numbers occurring in the active ν-formula are the same and assigned in the same way as in the principal ν-formula. s is a label and c is a label parameter.

The following definition originally defined in [Waaler 2001] is the relation bridging the world of sequent calculi like LC and matrix systems [Wallen 1990].

Definition 4.3 (Contextually equivalent)
The relation *contextually equivalent* is defined on formulae in a skeleton as the least equivalence relation such that:

1. an extra formula occurrence in the denominator of an inference r is contextually equivalent to its copy occurring in every sequent in the

numerator of r (also occurring as extra formula, i.e., the formula is implicitly copied in an inference on a different formula),

2. let r and r' be inferences on two contextually equivalent principal formula occurrences. Then an active formula of r is contextually equivalent to exactly one active formula of r' assuming the formulae are syntactically equal.

Two inferences are contextually equivalent if their principal formulae are contextually equivalent. ■

One should intuitively think of a set of contextually equivalent formulae as a way of expressing that the formulae are the same formula just occurring in different sequents in a skeleton. By this observation, contextually equivalent formulae should share some features. These features will be proved and exploited later.

Example 4.4 (Contextually equivalent formulae) The formulae in the following skeleton are superscripted only for ease of identification and are not part of the formulae. Formulae marked with the same boldface superscript are contextually equivalent. The decimal number (not bold) in the superscript uniquely identifies a formula in the skeleton.

$$\dfrac{\dfrac{\underset{1}{P}^{\mathbf{1}.10}, \underset{2}{P}^{\mathbf{4}.11} \vdash \underset{5}{\Diamond}\underset{6}{P}^{\mathbf{6}.12}, \underset{6}{P}[a]^{\mathbf{7}.13}}{\underset{1}{P}^{\mathbf{1}.4}, \underset{2}{P}^{\mathbf{4}.5} \vdash \underset{5}{\Diamond}\underset{6}{P}^{\mathbf{3}.6}}\,a \qquad \dfrac{\dfrac{\underset{1}{P}^{\mathbf{1}.18}, \underset{4}{Q}^{\mathbf{5}.19} \vdash \underset{5}{\Diamond}\underset{6}{P}^{\mathbf{8}.20}, \underset{6}{P}[b]^{\mathbf{9}.21}, \underset{6}{P}[a]^{\mathbf{7}.22}}{\underset{1}{P}^{\mathbf{1}.14}, \underset{4}{Q}^{\mathbf{5}.15} \vdash \underset{5}{\Diamond}\underset{6}{P}^{\mathbf{6}.16}, \underset{6}{P}[a]^{\mathbf{7}.17}}\,b}{\underset{1}{P}^{\mathbf{1}.7}, \underset{4}{Q}^{\mathbf{5}.8} \vdash \underset{5}{\Diamond}\underset{6}{P}^{\mathbf{3}.9}}\,a}{\underset{1}{P}^{\mathbf{1}.1}, \underset{2}{P} \underset{3}{\vee} \underset{4}{Q}^{\mathbf{2}.2} \vdash \underset{5}{\Diamond}\underset{6}{P}^{\mathbf{3}.3}}$$

The formulae in $\{\Diamond P^{\mathbf{3}.3}, \Diamond P^{\mathbf{3}.6}, \Diamond P^{\mathbf{3}.9}\}$[1] of formula are contextually equivalent since $\Diamond P^{\mathbf{3}.6}$ and $\Diamond P^{\mathbf{3}.9}$ are extra formulae in a β-inference on $P \vee QP^{\mathbf{3}.2}$ in the root sequent. $\{P^{\mathbf{1}.1}, P^{\mathbf{1}.4}, P^{\mathbf{1}.7}, P^{\mathbf{1}.10}, P^{\mathbf{1}.14}, P^{\mathbf{1}.18}\}$, $\{P^{\mathbf{4}.4}, P^{\mathbf{4}.11}\}$, $\{Q^{\mathbf{5}.8}, Q^{\mathbf{5}.15}, Q^{\mathbf{5}.19}\}$ and $\{P[a]^{\mathbf{7}.17}, P[a]^{\mathbf{7}.22}\}$ are contextually equivalent because they are implicitly copied as extra formulae. Since $\Diamond P^{\mathbf{7}.6}$ is contextually equivalent to $\Diamond P^{\mathbf{7}.9}$ and their components $P[a]^{\mathbf{7}.13}$ are $P[a]^{\mathbf{7}.17}$ are syntactically equal, the formulae in $\{P[a]^{\mathbf{7}.13}, P[a]^{\mathbf{7}.17}, P[a]^{\mathbf{7}.22}\}$ are contextually equivalent. This causes also the inferences marked with a to be contextually equivalent, since their principal contextually equivalent.

[1]Since formulae are uniquely identified by superscripts, and for reasons of readability, we do not display the polarity of the formulae in this example.

Observe that $\{\Diamond P^{\mathbf{3}.6}, \Diamond P^{\mathbf{6}.12}\}$, $\{\Diamond P^{\mathbf{3}.9}, \Diamond P^{\mathbf{6}.16}\}$ and $\{\Diamond P^{\mathbf{6}.16}, \Diamond P^{\mathbf{8}.20}\}$ are *not* sets of contextually equivalent formulae since one of the formulae in the sets is an explicit copy of the ν-formula in a ν-inference. Notice also that although both $P^{\mathbf{1}.4}$ and $P^{\mathbf{4}.6}$ seem syntactically equal and are copied as extra formulae in the same inference, they are not contextually equivalent as they are assumed differently formula numbered and therefore not syntactically equal. ○

Next, we prove the intuition given earlier: contextually equivalent formulae are in many ways the same formula, and should thus be syntactically equal and furthermore be satisfied in the same model.

Lemma 4.5 Let Ψ be a a set of contextually equivalent formulae. All formulae in the set are syntactically equal. ▪

Proof. Let $X, Y \in \Psi$ and assume they are contextually equivalent by point 1 in Definition 4.3. Then, since the contextually equivalent formulae are implicitly copied, X and Y must be syntactically equal.

Now assume X and Y are contextually equivalent by point 2 in Definition 4.3. Then they are active formulae in two contextually equivalent inferences, and by the same point in the definition they are syntactically equal. □

Corollary 4.6 (Satisfiability of contextually equivalent formulae) Let Ψ be a set of contextual equivalent formulae. If $F \in \Psi$ is satisfiable in a model $(\mathcal{M}, \phi)$ under a label interpretation ρ, then $(\mathcal{M}, \phi)$ satisfies Ψ under ρ. ▪

Proof. Since contextually equivalent formulae are syntactically equal, by Lemma 4.5, the result is immediate. □

Having established the notion of contextually equivalent, we are ready to define the conditions necessary to categorize a skeleton as an LC^{g}-*derivation*.

Definition 4.7 ($\mathsf{LC}^{\mathsf{g}}_L$-derivation)
If δ is a skeleton having an empty labelled root and every inference in δ respects the following conditions set by the logic L, we call it an $\mathsf{LC}^{\mathsf{g}}_L$-derivation:

- every inference is an instance of a rule in the LC^{g}-calculus,
- the *contextually equivalent condition:* The inferences r and r' belong to the same label if the respective inferences are contextually equivalent,
- the *ground label condition:* A ν-inference may only introduce labels $t \in \mathsf{Par}_L$,

- the *eigenparameter condition:* A π-inference may only introduce label parameters fresh to the denominator, and
- the *non-empty* W^+ *condition*: A ν-inference r is applicable only if there is a label $s \in \mathsf{Par}_L$ in r. ∎

The ground label condition ensures that LC^{g} is a ground calculus. Since every label in a skeleton conforming to the rules of LC^{g} belongs to a π- or a ν-rule, and π-inferences may only introduce label parameters, we only have to control the label introduction of ν-inferences to make sure the calculus is ground. As mentioned in the introduction of the thesis, the contextually equivalent condition is of importance for making the LC-calculi available to matrix systems [Wallen 1990]. The eigenparameter and non-empty W^+ condition are vital for establishing soundness of LC^{g}. The eigenparameter condition restricts a π-inference to only introducing fresh label parameters. This is necessary to stay in control of the interpretation of the introduced label, and is best explained by reading through the proof of Lemma 3.15, satisfaction of components, and seeing how a satisfying model for a $\pi_0(c)$-formula is *constructed* by using the fact that the introduced label parameter c is fresh. The non-empty W^+ condition is only of interest to $\mathsf{LC}^{\mathsf{g}}_{K45}$-derivations, since in an $\mathsf{LC}^{\mathsf{g}}_{S5}$-derivation the ground label $\epsilon \in \mathsf{Par}_{S5}$ occurs in every branch of an $\mathsf{LC}^{\mathsf{g}}_{L}$-derivation, meaning that the condition is trivially met in S5. The intuitions behind the non-empty W^+ condition is as follows: Consider a construction of a branch in an LC^{g}-derivation as the gathering of information to build an L-countermodel, an L-model satisfying every formula on the branch. If there are no label parameters occurring in the branch, then we cannot assume that the constructed L-model's W^+-set is non-empty. If W^+ is empty, then applying the ν-rule in LC^{g} is not sound.[2]

Example 4.8 (Contextually equivalent condition) The contextually equivalent condition plays a key role in all the LC-calculi. This example gives insights to it's effects on LC^{g}-derivations. Consider the three skeletons δ_1, δ_2 and δ_3 below, which all have the same root sequent $P \vee Q \vdash \Diamond R$, and direct special attention to the inferences on $\Diamond R_\perp$. The explicit copy of this ν-formula is not displayed.

[2]For a explanation of this, read the proof of Lemma 3.15 and the discussion following it.

$$\dfrac{\dfrac{P \vdash R[a]}{P \vdash \Diamond R}\,a \qquad \dfrac{Q \vdash R[b]}{Q \vdash \Diamond R}\,b}{P \vee Q \vdash \Diamond R}$$
$$\delta_1$$

$$\dfrac{\dfrac{P \vdash R[a] \qquad Q \vdash R[a]}{P \vee Q \vdash R[a]}}{P \vee Q \vdash \Diamond R}\,a$$
$$\delta_2$$

$$\dfrac{\dfrac{P \vdash R[a]}{P \vdash \Diamond R}\,a \qquad \dfrac{Q \vdash R[a]}{Q \vdash \Diamond R}\,a}{P \vee Q \vdash \Diamond R}$$
$$\delta_3$$

The skeleton δ_1 violates the contextually equivalent condition, since the two inferences on $\Diamond R$ in each branch of δ_1 introduce different labels, a and b. These inferences are contextual equivalent, as their principal formulae are contextual equivalent, and they should therefore belong to the same label. The skeleton δ_2 does not violate the contextually equivalent condition. What remedies the situation is the order of application of inferences. In δ_2 we first apply a ν-inference, introducing the label a to the skeleton, and proceed by applying a β-inference. Since the ν-inference in δ_2 is contextually equivalent to no other inference at the time of application, it cannot break the contextually equivalent condition by introducing an "incorrect" label. In δ_1 we first apply a β-inference and continue by applying ν-inferences to two contextually equivalent formulae in both branches. By having two ν-inferences in different branches the possibility of introducing different labels in each branch is present. This is a possibility we want to remove, and this is what the contextually equivalent condition does, by requiring that the two inferences belong to the same label. δ_3 has the same order of rule application as δ_1, but in contrast to δ_1, it upholds the contextually equivalent condition, introducing a in both branches.

Observe that the leaf sequents of δ_2 and δ_3 are equal, while the leaves of δ_1, the skeleton *not* complying with the contextually equivalent condition, differs from the other's leaves. This serves as an early indication that the condition is necessary to have the desired invariance of rule application order in skeletons. ◦

The lesson learned from the example above is that the contextually equivalent condition is necessary to obtain skeletons with equal leaf sequents even though inferences may have been applied in different order. This feature is of key importance in the free-variable calculi we will introduce in chapter 6. We want our derivations to be *invariant under order of rule application* [Waaler 2001], i.e., the form of the leaf sequents in a skeleton should not depend on the order rules are applied. In most calculi of similar type in the literature, one does not require the introduction of equal labels as required by the contextual equivalence condition. This requirement does in some cases in fact give longer derivations than if the requirement is dropped.

So why do we want this seemingly unnecessary restriction? What is the gain of working with skeletons invariant under order of rule application? The reason is that of automated reasoning efficiency. Invariance under order of rule application in a sequent calculus as the LC-calculi makes it convert easily into a system of matrices as described in [Wallen 1990]. The conversion is performed by a set of simple rewriting rule listed in [Waaler 2001, p. 1504]. The matrix systems inhibit the enjoyable feature of being readily available for application by a *goal-directed connection driven* search method. For more on connection driven search and the *connection calculus* consult [Bibel 1987, Holen 2005, Kreitz and Otten 1999]. As this thesis' focus is not implementation, we will not go into these aspects in detail. Although not mentioned more than this, it is important to bear in mind that implementation of the LC-calculi is the ultimate goal of this work.

The contextually equivalent condition is necessary in order to have invariance of order of rule application. This condition is however not enough and the feature of invariance will first be obtained by the free-variable $\mathsf{LC}^{\mathsf{fv}}$-calculus in chapter 6. The reason why the LC^{g}-calculus is not invariant under order of rule application is because of the eigenparameter condition and the non-empty W^+ condition. The reason is quite simple: The eigenparameter condition allows for label parameters to be introduced only when the label does not occur in the denominator. The freshness of a label introduced depends entirely on what labels are already introduced to the skeleton, and this depends in turn on what order inferences are applied. The non-empty W^+ condition represents a kind of inverted eigenparameter condition, so a similar argument is applicable to why invariance of rule application is impossible to obtain in LC^{g}.

Example 4.9 (Invariance of rule application order) Consider the skeletons δ_4, δ_5 and δ_6 all over the same root sequent, but with a different order of inferences and introduced labels. The order of rule application is indicated by marking the horizontal line to which the inference corresponds with both its type and belonging label.

$$\dfrac{\dfrac{\Box P, P[a] \vdash Q[a]}{\Box P \vdash Q[a]}\,\nu, a}{\Box P \vdash \Box Q}\,\pi, a \qquad \dfrac{\dfrac{\Box P, P[a] \vdash Q[a]}{\Box P, P[a] \vdash \Box Q}\,\pi, a}{\Box P \vdash \Box Q}\,\nu, a \qquad \dfrac{\dfrac{\Box P, P[a] \vdash Q[b]}{\Box P, P[a] \vdash \Box Q}\,\pi, b}{\Box P \vdash \Box Q}\,\nu, a$$

$$\delta_4 \qquad\qquad \delta_5 \qquad\qquad \delta_6$$

The skeletons δ_4 and δ_6 both comply with the eigenparameter condition, but δ_5 is not an $\mathsf{LC}^{\mathsf{g}}_L$-derivation as it does not meet the eigenparameter condition by letting the π-formula $\Box Q_\perp$ introduce an unfresh label parameter. Notice that δ_4 and δ_5 have the same leaf sequents, implying that the eigenparameter condition needs to be broken in order to gain invariance of rule application order. δ_6 does not break the eigenparameter condition, since the label parameter belonging to the π-inference on $\Box Q_\perp$ is fresh to the denominator, but it does not have the same leaves as δ_4, since it because of this required freshness must introduce a label parameter different from a.

The skeletons δ_5 and δ_6 both break the non-empty W^+ condition in K45, by applying ν-rules to a sequents containing no labels in Par_{K45}.

As we see, the LC^{g}-calculus is not invariant under rule application order, since a change in the order of the inferences on a skeleton, if it is admissible, results in the leaves ending up different. ◦

Remark (to Example 4.9). The example 4.9 may not convince the reader—and it does not intend to either—of the necessity of the eigenparameter condition, only that it hinders the desired invariance of inferences. The fatal consequence of removing this condition is that the LC^{g}-calculus becomes unsound; we will be able to prove sequents which are not valid. An example of this is delayed until we have established the necessary notions of axiom and proof in LC^{g}. ◦

Example 4.10 ($\mathsf{LC}^{\mathsf{g}}_L$-derivation) We have now seen six examples of skeletons, but only in the spirit of illustrating the eigenparameter, contextually equivalent condition and how these affect the (lack of) invariance of rule application order. The approach of the following example on the same set of skeletons is to exemplify which of the skeletons are $\mathsf{LC}^{\mathsf{g}}_{K45}$- and $\mathsf{LC}^{\mathsf{g}}_{S5}$-derivations.

Of the skeletons δ_1–δ_6 immediately above

- δ_1 is no $\mathsf{LC}^{\mathsf{g}}_L$-derivation as it does not adapt the contextual equivalent condition,
- δ_2 and δ_3 are no $\mathsf{LC}^{\mathsf{g}}_{K45}$-derivations since they fail to conform to the non-empty W^+ condition when introducing the label parameter a. They are both $\mathsf{LC}^{\mathsf{g}}_{S5}$-derivations.
- The skeleton δ_4 is an $\mathsf{LC}^{\mathsf{g}}_{K45}$-derivation, while

- δ_5 and δ_6 break the non-empty W^+ condition and are hence no $\mathsf{LC}^{\mathsf{g}}_{K45}$-derivations.
- δ_5 is not an $\mathsf{LC}^{\mathsf{g}}_{L}$-derivation because it does not agree with the eigenparameter condition, but
- δ_4 and δ_6 are both $\mathsf{LC}^{\mathsf{g}}_{S5}$-derivations. ◦

To be able to define what a proof is in the LC^{g}-calculus, we need to establish the notion of an axiom.

Definition 4.11 (Axiom, $\mathbf{LC^g}$)

An axiom in the LC^{g}-calculus is a labelled sequent of the form $\Gamma, F[s]_\top, F[s]_\bot$, i.e., a sequent containing two labelled formulae comprised by the same core formula and labelling, but having different polarity. The formulae $F[s]_\top, F[s]_\bot$ are referred to as a *closing pair*. ▪

Lemma 4.12 An axiom is not satisfiable. ▪

Proof. Assume that the axiom $\Gamma, P[s]_\top, P[s]_\bot$ is true[3]. Then there must exist a model $(\mathcal{M}, \phi)$ such that $(\mathcal{M}, \phi) \vDash \Gamma, P[s]_\top, P[s]_\bot$. This is a contradiction, since it means that both $\mathcal{M}, \phi(s) \vDash P$ and $\mathcal{M}, \phi(s) \nvDash P$ hold. □

Remark. Lemma 4.12 may come as a surprise to the reader. Axioms are normally thought of as valid, but not being satisfiable seems to contradict this. Well, this is not true. A sequent which is not satisfiable has by Definition 3.20 no L-countermodel, and quoting Definition 3.21: *"A sequent is L-valid if it has no L-countermodel."* So axioms are valid after all. See also the remark on page 26. ◦

If a branch in an $\mathsf{LC}^{\mathsf{g}}_{L}$-derivation contains an axiom, we say that the branch is *closed*. This is the closure condition of the calculus LC^{g}. If every branch in an $\mathsf{LC}^{\mathsf{g}}_{L}$-derivation δ is closed, δ is an $\mathsf{LC}^{\mathsf{g}}_{L}$*-proof*.

Definition 4.13 ($\mathbf{LC}^{\mathbf{g}}_{L}$-proof)

An $\mathsf{LC}^{\mathsf{g}}_{L}$-proof is a finite $\mathsf{LC}^{\mathsf{g}}_{L}$-derivation where every branch contains an axiom. ▪

The empty labelled sequent Γ is $\mathsf{LC}^{\mathsf{g}}_{L}$-provable if there exists an $\mathsf{LC}^{\mathsf{g}}_{L}$-proof δ where Γ is the root sequent of δ.

Next we do a series of examples of LC^{g}-derivations and -proofs. We indicate a closed branch by placing a cross $\times$ above it, while open branches are indicated by $\circ$.

[3]Remember that a ground sequent is either true in an L-model or not satisfied.

Remark. In axiomatic systems different modal logics are identified by a set of formulae in these systems called axioms, where these formulae define the relation on the set of worlds, i.e., the accessibility relation, see e.g. [Blackburn et al. 2005, Chellas 1988]. The formula called **K**, $\Box(P \to Q) \to (\Box P \to \Box Q)$, holds in all normal modal logics, which both K45 and S5 are considered to be. Furthermore, S5 is often identified by the formulae $\Box P \to P$, $\Box P \to \Box\Box P$ and $\Diamond P \to \Box\Diamond P$, called respectively **T**, **4** and **5**. K45 is identified by **K**, **4** and **5**, which is the reason for its name. These formulae form a natural starting point for the first display of the mechanics of the LC^{g}-calculus. The first four examples presented below show that the correct set of formulae is LC^{g}-provable in the two logics. ◦

*Example 4.14 (**T**-axiom:* $\mathsf{LC}^{\mathsf{g}}_{S5}$*-proof, K45-countermodel)* The formula $\Box P \to P$ is the standard axiom of reflexivity in axiomatic systems. By this observation the sequent $\vdash \Box P \to P$ should be $\mathsf{LC}^{\mathsf{g}}_{S5}$-provable, and not $\mathsf{LC}^{\mathsf{g}}_{K45}$-provable.

$$\dfrac{\dfrac{\overset{\times}{\Box P, P \vdash P}}{\Box P \vdash P}\,\epsilon}{\vdash \Box P \to P}$$

Remember that unlabelled formulae in labelled sequents are labelled with ϵ. We see that $P_\bot$ must be one of the formulae in the closing pair of a possible axiom since it is the only formula having polarity $\bot$, and no formula with polarity $\bot$ can arise from an inference on $\Box P_\top$. This means that the only closing pair for a proof of $\Box P \to P_\bot$ must be $P_\top, P_\bot$. The side conditions on ν-inferences allow for the introduction of ϵ only in S5, hence the above skeleton is an $\mathsf{LC}^{\mathsf{g}}_{S5}$-derivation and an $\mathsf{LC}^{\mathsf{g}}_{S5}$-proof, but not an $\mathsf{LC}^{\mathsf{g}}_{K45}$-derivation and hence no $\mathsf{LC}^{\mathsf{g}}_{K45}$-proof either. The following is $\mathsf{LC}^{\mathsf{g}}_{K45}$-derivation of the sequent.

$$\dfrac{\overset{\circ}{\Box P \vdash P}}{\vdash \Box P \to P}$$

By the open branch in the derivation we conclude that the $\mathsf{LC}^{\mathsf{g}}_{K45}$-derivation is not an $\mathsf{LC}^{\mathsf{g}}_{K45}$-proof. Since we claim that the sequent is not $\mathsf{LC}^{\mathsf{g}}_{K45}$-provable, we should be able to find a K45-countermodel satisfying the sequent. A good starting point is to find a K45-model satisfying the leaf sequent of an open branch in the skeleton. To satisfy the sequent $\Box P \vdash P$, a model needs every point in W^+ to satisfy $P_\top$ in order for $\Box P_\top$ to be satisfied, and the point to which the ground label interpretation ϕ assigns ϵ needs to satisfy $P_\bot$ (†). One such K45-countermodel is $(\mathcal{M}, \phi)$ where $\mathcal{M} = (W, W^+, V)$ and where $\phi(\epsilon) \notin V(P)$, and $w_i \in V(P)$ for all $w_i \in W^+ \neq \emptyset$.

An attempt to construct an S5-countermodel for the sequent would fail. By (†) every point in W^+ must satisfy $P_\perp$ and the point $\phi(\epsilon)$ must satisfy $P_\perp$. But in an S5 model, $\phi(\epsilon) \in W^+$, hence $\phi(\epsilon)$ must satisfy both $P_\perp$ and $P_\top$. This is of course impossible. Note that this is the exact same observation the $\mathsf{LC}^{\mathsf{g}}_{S5}$-proof of the sequent $\vdash \Box P \rightarrow P$ makes in its leaf sequent. $\circ$

*Example 4.15 (**K**-axiom: $\mathsf{LC}^{\mathsf{g}}_{K45}$- and $\mathsf{LC}^{\mathsf{g}}_{S5}$-proof)* The sequent

$$\vdash \Box(P \rightarrow Q) \rightarrow (\Box P \rightarrow \Box Q)_\perp$$

is both $\mathsf{LC}^{\mathsf{g}}_{K45}$- and $\mathsf{LC}^{\mathsf{g}}_{S5}$-provable. We let χ abbreviate the explicit copy of the ν-formula $\Box(P \rightarrow Q)_\top$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\times}{\chi, \Box P, P[a] \vdash P[a], Q[a]}}{\chi, \Box P \vdash P[a], Q[a]}\,a \qquad \dfrac{\times}{\chi, Q[a], \Box P \vdash Q[a]}}{\chi, P \rightarrow Q[a], \Box P \vdash Q[a]}}{\Box(P \rightarrow Q), \Box P \vdash Q[a]}\,a}{\Box(P \rightarrow Q), \Box P \vdash \Box Q}\,a}{\Box(P \rightarrow Q) \vdash \Box P \rightarrow \Box Q}}{\vdash \Box(P \rightarrow Q) \rightarrow (\Box P \rightarrow \Box Q)}$$

The skeleton conforms to the conditions set by an $\mathsf{LC}^{\mathsf{g}}_{L}$-derivation in both K45 and S5, and every branch contains an axiom, so the skeleton is both an $\mathsf{LC}^{\mathsf{g}}_{K45}$- and $\mathsf{LC}^{\mathsf{g}}_{S5}$-proof, hence **K** is both $\mathsf{LC}^{\mathsf{g}}_{K45}$- and $\mathsf{LC}^{\mathsf{g}}_{S5}$-provable. The closing pair in the left branch is $P[a]_\top, P[a]_\perp$, and $P[b]_\top, P[b]_\perp$ the closing pair in the right branch. $\circ$

*Example 4.16 (**4**-, **5**-axiom: $\mathsf{LC}^{\mathsf{g}}_{K45}$- and $\mathsf{LC}^{\mathsf{g}}_{S5}$-proof)* The left skeleton is an $\mathsf{LC}^{\mathsf{g}}_{K45}$- and $\mathsf{LC}^{\mathsf{g}}_{S5}$-proof of the sequent $\vdash \Box P \rightarrow \Box\Box P$. The right skeleton shows that the sequent $\vdash \Diamond P \rightarrow \Box\Diamond P$ is $\mathsf{LC}^{\mathsf{g}}_{K45}$- and $\mathsf{LC}^{\mathsf{g}}_{S5}$-provable.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\times}{\Box P, P[b] \vdash P[b]}}{\Box P \vdash P[b]}\,b}{\Box P \vdash \Box P[a]}\,b}{\Box P \vdash \Box\Box P}\,a}{\vdash \Box P \rightarrow \Box\Box P} \qquad\qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\times}{P[a] \vdash \Diamond P[b], P[a]}}{P[a] \vdash \Diamond P[b]}\,a}{P[a] \vdash \Box\Diamond P}\,b}{\Diamond P \vdash \Box\Diamond P}\,a}{\vdash \Diamond P \rightarrow \Box\Diamond P}$$

In order to quickly obtain a closing pair a good strategy is often to apply π-inferences first, introducing fresh label parameters, and proceed by introducing the same labels by ν-inferences. This strategy is used on both of the skeletons displayed above, and avoids breaking the non-empty W^+ condition or breaking the eigenparameter condition by letting π-inferences introduce label parameter already occurring in the denominator. If ν-inferences are applied before π-inferences, and it is not required by the structure of the formulae, the result is often that more rule applications than strictly necessary are needed in order to obtain a proof. $\circ$

Remark (to Example 4.16). Notice that the labelling of π- and ν-formulae is "overwritten" when subject to a rule application, i.e., every label is at most one character long. This is an indication of the relatively low complexity of the SAT-problem—namely NP-complete [Garey and Johnsen 1979]—the logics K45 and S5 enjoy [Ladner 1977]. $\circ$

Example 4.17 (Eigenparameter condition, L-countermodel) The following is an example of the importance of the eigenparameter condition as promised in the remark succeeding Example 4.9.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\overset{\times}{\chi, P[a] \vdash P[a], \Box Q} \qquad \dfrac{\dfrac{\dfrac{\overset{\circ}{\chi, P[b], Q[a] \vdash P[a], Q[b]} \qquad \overset{\times}{\chi, Q[b], Q[a] \vdash P[a], Q[b]}}{\chi, (P \lor Q)[b], Q[a] \vdash P[a], Q[b]}}{\chi, Q[a] \vdash P[a], Q[b]}\,b}{\chi, Q[a] \vdash P[a], \Box Q}\,b}{\chi, (P \lor Q)[a] \vdash P[a], \Box Q}}{\Box(P \lor Q) \vdash P[a], \Box Q}\,a}{\Box(P \lor Q), \vdash \Box P, \Box Q}\,a}{\Box(P \lor Q) \vdash \Box P \lor \Box Q}
$$

No further rule application will succeed in creating an $\mathsf{LC}^{\mathsf{g}}_L$-proof. In fact using the $\mathsf{LC}^{\mathsf{g}}_L$-derivation at hand and looking at the open branch, creating an L-countermodel is easy. An K45-countermodel is model $(\mathcal{M}, \phi)$, where ϕ is the identity function and $\mathcal{M} = (W, W^+, V)$ is such that $W^+ = \{a, b\}$, $W = W^+ \cup \{\epsilon\}$, and $V(P) = \{b\}$ and $V(Q) = \{a\}$. An S5-model is obtained from the adjusting the K45-model such that $W = W^+ = \{a, b, \epsilon\}$. Both models satisfy the set $P[b]_\top, Q[a]_\top, P[a]_\bot, Q[b]_\bot$ and the root sequent of the skeleton.

Now we present an LC^{g}-"proof" of the same sequent by violating the eigenparameter condition.

$$\dfrac{\dfrac{\dfrac{\dfrac{\overset{\times}{\chi, P[a] \vdash P[a], \Box Q} \qquad \dfrac{\overset{\times}{\chi, Q[a] \vdash P[a], Q[a]}}{\chi, Q[a] \vdash P[a], \Box Q}\, a, *}{\chi, (P \lor Q)[a] \vdash P[a], \Box Q}\, a}{\Box(P \lor Q) \vdash P[a], \Box Q}\, a}{\Box(P \lor Q), \vdash \Box P, \Box Q}}{\Box(P \lor Q) \vdash \Box P \lor \Box Q}$$

The inference marked with $*$ does not respect the eigenparameter condition since it belongs to a label parameter occurring in the denominator, hence the skeleton is not an LC^{g}-derivation or an LC^{g}-proof. $\circ$

Example 4.18 (Non-empty W^+ condition, $\mathsf{LC}^{\mathsf{g}}_{S5}$-proof) First we display an $\mathsf{LC}^{\mathsf{g}}_{S5}$-proof of the labelled sequent $\vdash \Box(P \to P) \land \Diamond(P \to P)_\perp$ and afterwards show and explain why the same sequent is not $\mathsf{LC}^{\mathsf{g}}_{K45}$-provable.

$$\dfrac{\dfrac{\dfrac{\overset{\times}{P[a] \vdash P[a]}}{\vdash P \to P[a]}}{\vdash \Box(P \to P)}\, a \qquad \dfrac{\dfrac{\overset{\times}{P[b] \vdash P[b], \Diamond(P \to P)}}{\vdash P \to P[b], \Diamond(P \to P)}}{\vdash \Diamond(P \to P)}\, b}{\vdash \Box(P \to P) \land \Diamond(P \to P)}$$

The above skeleton is an $\mathsf{LC}^{\mathsf{g}}_{S5}$-derivation as it does not break any of the conditions set by an $\mathsf{LC}^{\mathsf{g}}_{S5}$-derivation, and since it contains an axiom in every branch, it is also an $\mathsf{LC}^{\mathsf{g}}_{S5}$-proof. Since the label $\epsilon \in \mathsf{Par}_{S5}$ occurs in the branch of every LC^{g}-derivation, the non-empty W^+ is trivially met by an $\mathsf{LC}^{\mathsf{g}}_{S5}$-derivation.

The reason why the skeleton is not also an $\mathsf{LC}^{\mathsf{g}}_{K45}$-derivation is because the introduction of the label b in the right branch violates the non-empty W^+ condition. b does not occur in the branch prior to its introduction by the ν-inference on $\Diamond(P \to P)_\perp$. The only label occurring in the right branch at this stage is ϵ, but an introduction of ϵ by an inference on the ν-formula is, as Example 4.14 shows, not permissible by a ν-inference in K45. An admissible $\mathsf{LC}^{\mathsf{g}}_{K45}$-derivation with the same root sequent as above is displayed below.

$$\dfrac{\dfrac{\dfrac{\overset{\times}{P[a] \vdash P[a]}}{\vdash P \to P[a]}}{\vdash \Box(P \to P)}\, a \qquad \overset{\circ}{\vdash \Diamond(P \to P)}}{\vdash \Box(P \to P) \land \Diamond(P \to P)}$$

No rule is applicable to the ν-formula in the right branch, and as we see by the lack of axiom in this branch the $\mathsf{LC}^{\mathrm{g}}_{K45}$-derivation is not an $\mathsf{LC}^{\mathrm{g}}_{K45}$-proof.

A K45-countermodel for $\Box(P \to P) \wedge \Diamond(P \to P)_\perp$ is a model where $W^+ = \emptyset$. Let $\mathcal{M} = (W, W^+, V)$, where W^+ is empty. From the definition of a ground label interpretation function ϕ we know that $\phi(\epsilon) \in W \setminus W^+$. The model $(\mathcal{M}, \phi)$ satisfies the right leaf sequent, $(\mathcal{M}, \phi) \vDash \Diamond(P \to P)_\perp$, since $\mathcal{M}, \phi(\epsilon) \vDash \Diamond(P \to P)_\perp$ given that $\mathcal{M}, w' \vDash (P \to P)_\perp$ holds trivially for all $w' \in W^+$, when $W^+ = \emptyset$. Since $(\mathcal{M}, \phi) \vDash \Diamond(P \to P)$, it follows that $(\mathcal{M}, \phi)$ satisfies the root sequent. ◦

Example 4.19 (Non-empty W^+ condition) This example displays the consequences of the non-empty W^+ condition in regards to the lack of invariance under order of rule application in the LC^{g}-calculus. The skeleton will also serve as a comparison of the shortened proof lengths we can achieve in the free variable calculus $\mathsf{LC}^{\mathrm{fv}}$ presented in chapter 6. The sequent $\vdash \Diamond(P \to \Box P), \Box Q$ is valid in both K45 and S5, but $\vdash \Diamond(P \to \Box P)$ is only S5-valid. These sequents and observations will follow us through chapter 5 and especially chapter 6. χ is an abbreviation for the explicit copy of $\Diamond(P \to \Box P)_\perp$.

$$
\begin{array}{rcll}
 & \times & & \\
P[a], P[b] & \vdash & P[b], \Box P[b], \chi, Q[a] & \\
\hline
P[a] & \vdash & P[b], P \to \Box P[b], \chi, Q[a] & b \\
\hline
P[a] & \vdash & P[b], \chi, Q[a] & \\
\hline
P[a] & \vdash & \Box P[a], \chi, Q[a] & b \\
\hline
 & \vdash & P \to \Box P[a], \chi, Q[a] & \\
\hline
 & \vdash & \Diamond(P \to \Box P), Q[a] & a \\
\hline
 & \vdash & \Diamond(P \to \Box P), \Box Q & a
\end{array}
$$

The derivation is both an $\mathsf{LC}^{\mathrm{g}}_{K45}$- and $\mathsf{LC}^{\mathrm{g}}_{S5}$-proof. Below we display the same skeleton as above, except the inference on $\Box Q_\perp$ is the lowermost in the skeleton above, while it is the uppermost in the skeleton below.

$$
\begin{array}{rcll}
P[a], P[b] & \vdash & P[b], \Box P[b], \chi, Q[a] & \\
\hline
P[a], P[b] & \vdash & P[b], \Box P[b], \chi, \Box Q & a \\
\hline
P[a] & \vdash & P[b], P \to \Box P[b], \chi, \Box Q & \\
\hline
P[a] & \vdash & P[b], \chi, \Box Q & b \\
\hline
P[a] & \vdash & \Box P[a], \chi, \Box Q & b \\
\hline
 & \vdash & P \to \Box P[a], \chi, \Box Q & \\
\hline
 & \vdash & \Diamond(P \to \Box P), \Box Q & a
\end{array}
$$

This skeleton is not an $\mathsf{LC}^{\mathsf{g}}_{K45}$-derivation, as it violates the non-empty W^+ condition by the lowermost inference introducing a label parameter. This inference is a ν-inference and can only be applied if there is a label $s \in \mathsf{Par}_L$ occurring in the denominator. Observe that the leaf sequents of the two skeletons above are equal. This shows that we do not have invariance under rule application in $\mathsf{LC}^{\mathsf{g}}_{K45}$, a feature whose absence the non-empty W^+ condition is partly to blame.

Note that without the presence of the π-formula $\Box Q_\perp$ we would not be able to construct an $\mathsf{LC}^{\mathsf{g}}_{K45}$-proof at all.[4] The lack of label parameter in the branch at the stage of rule application to $\Diamond(P \to \Box P)_\perp$ would cause the search to terminate. In S5 the non-empty W^+ condition is trivially satisfied. The skeleton gathered from erasing the formulae $\Box Q_\perp$ and $Q[a]_\perp$ from the above example is an $\mathsf{LC}^{\mathsf{g}}_{S5}$-proof of $\Diamond(P \to \Box P)_\perp$. An $\mathsf{LC}^{\mathsf{g}}_{S5}$-proof of this sequent is displayed below.

$$
\begin{array}{rcl}
 & \times & \\
P[a], P[b] & \vdash & P[b], \Box P[b], \chi \\ \hline
P[a] & \vdash & P[b], P \to \Box P[b], \chi \\ \hline
P[a] & \vdash & P[b], \chi \\ \hline
P[a] & \vdash & \Box P[a], \chi \quad b \\ \hline
 & \vdash & P \to \Box P[a], \chi \\ \hline
 & \vdash & \Diamond(P \to \Box P) \quad a
\end{array}
$$

$\circ$

4.1 Soundness

Soundness of the LC^{g}-calculus means that if the calculus proves a sequent in the logic L, the sequent is true in every L-model—or equivalently, it has no L-countermodel.

A common way of proving soundness of a calculus is to show that all its single steps are correct. If every step it takes is correct then the whole cannot do anything wrong. The steps of the LC^{g}-calculus is of course its inferences, the conditions on an LC^{g}-derivation and its closing condition. The proof is established by showing that if a skeleton has an L-countermodel, then a rule application to the skeleton results in an another skeleton having an L-countermodel, in other words, showing that an LC^{g}-proof cannot arise from a sequent which has an L-countermodel. This technique is similar to the soundness proof found in many standard logic textbooks concerning ground sequent calculi, see e.g., [Fitting 1996].

[4] A K45-countermodel is found in Example 2.9.

Since all labels in the LC^{g}-calculus are ground, we need to specialize Lemma 3.15 to such formulae and the fact that they are either true or not satisfied.

Corollary 4.20 (Satisfaction of grounded components) Let Γ be a set of ground labelled formulae and $(\mathcal{M}, \phi)$ an L-model where $\mathcal{M} = (W, W^+, V)$.

1. If $\alpha \in \Gamma$, then $\Gamma \cup \{\alpha_1, \alpha_2\}$ is true in $(\mathcal{M}, \phi)$, iff Γ is true in $(\mathcal{M}, \phi)$.
2. If $\beta \in \Gamma$, then $\Gamma \cup \{\beta_1\}$ is true in $(\mathcal{M}, \phi)$ or $\Gamma \cup \{\beta_2\}$ is true in $(\mathcal{M}, \phi)$, iff Γ is true in $(\mathcal{M}, \phi)$.
3. If $\nu \in \Gamma$ and $W^+ \neq \emptyset$, then $\Gamma \cup \{\nu_0(s)\}$ is true in $(\mathcal{M}, \phi)$ for every label $s \in \mathsf{Par}_L$.
4. If $\pi \in \Gamma$, then $\Gamma \cup \{\pi_0(c)\}$ is true in $(\mathcal{M}, \phi')$ for some extension ϕ' by c of ϕ, iff Γ is true in $(\mathcal{M}, \phi)$, where c is an arbitrary label parameter not occurring in Γ. ▪

Proof sketch. All cases of the lemma can easily be proven by using Lemma 3.15. We present a only general proof here. Let Γ be a set of ground labelled formulae and assume Γ is true in the L-model $(\mathcal{M}, \phi)$, where $\mathcal{M} = (W, W^+, V)$, under the conditions stated in the claim. Then Γ is satisfied in $(\mathcal{M}, \phi)$ for every label variable assignment ρ. Conclude by applying Lemma 3.15.

The other direction of the points 1, 2 and 4 is similar. □

Now we are ready to prove the lemma which carries the greatest burden in the proof of soundness of LC^{g}, the result that rule applications preserve the satisfaction of skeletons.

Lemma 4.21 (Countermodel preservation) Assume that there is an L-countermodel for the branch θ in the $\mathsf{LC}^{\mathsf{g}}_L$-derivation δ, and let Γ be a leaf sequent of δ. Apply an admissible inference to Γ creating the branch θ' in the $\mathsf{LC}^{\mathsf{g}}_L$-derivation δ'. Then there is an L-countermodel for δ'. ▪

Proof. First assume that Γ is not the leaf sequent of the branch θ. Then θ must be a branch in δ', and δ' must have an L-countermodel.

In the following assume that Γ is the leaf sequent of θ, and that $(\mathcal{M}, \phi)$ is an L-countermodel for θ. We prove the remaining cases of the lemma by checking for each possible type of inference r on Γ producing the branch θ' in δ' that there exists an extension ϕ' of ϕ such that there is an L-countermodel $(\mathcal{M}, \phi')$ for θ'.

α. Assume that $\alpha \in \Gamma$ and that r is an inference on α. By Corollary 4.20 $\Gamma \cup \{\alpha_1, \alpha_2\}$ is true in $(\mathcal{M}, \phi)$. Then $(\mathcal{M}, \phi)$ is an L-countermodel for θ' and δ'.

β. Assume that r is a β-inference on some formula $\beta \in \Gamma$. By Corollary 4.20 $\Gamma \cup \{\beta_1\}$ or $\Gamma \cup \{\beta_2\}$ is true in the model $(\mathcal{M}, \phi)$. $(\mathcal{M}, \phi)$ is then an L-countermodel for at least one of the branches in δ'.

ν. This case splits in two. First assume that r is a rule application to some ν-formula occurring in Γ and that $W^+ \neq \emptyset$. By the ground label condition a ν-inference may only introduce labels from the set Par_L, so θ is extended with the sequent $\Gamma, \nu(s)$, where $s \in \mathsf{Par}_L$. By Corollary 4.20 $\Gamma \cup \{\nu_0(s)\}$ is true in $(\mathcal{M}, \phi)$ for any label $s \in \mathsf{Par}_L$. Thus $(\mathcal{M}, \phi)$ is an L-countermodel for θ'.

Now assume that W^+ is empty. Then L must be the logic K45, as W^+ is non-empty in every S5-model. By the non-empty W^+ condition in K45 there must exist a label parameter in Γ. Since W^+ is empty and $(\mathcal{M}, \phi)$ is by assumption a K45-countermodel for the branch θ, we claim that no ν-inference is applicable.

> *Claim.* If $W^+ = \emptyset$, then no ν-inference is applicable. ◦
>
> *Proof.* Aiming for a contradiction assume that a ν-inference *is* applicable. Then there must, by the non-empty W^+ condition, exist some label parameter c labelled to a formula P in Γ. Since $(\mathcal{M}, \phi)$ is a K45-countermodel for θ, $P[c]$ is true in $(\mathcal{M}, \phi)$ and by the definition of truth there is a point $\phi(c)$ in W^+ satisfying P. This is a contradiction since W^+ is assumed empty. □

Since no ν-inference is applicable to θ, the lemma is in this case trivial.

π. Assume r is a π-inference on some formula $\pi \in \Gamma$. The branch θ is then extended by the sequent containing a formula $\pi_0(c)$. By the eigenparameter condition, the label parameter c is fresh to the denominator. Then, by Corollary 4.20, there is an extension ϕ' by c of ϕ such that $\Gamma \cup \{\pi_0(c)\}$ is true in $(\mathcal{M}, \phi')$, making $(\mathcal{M}, \phi')$ an L-countermodel for δ'. □

Theorem 4.22 (Soundness of LC^{g})
Let δ be an $\mathsf{LC}^{\mathsf{g}}_L$-proof of a sequent Γ. Then Γ is L-valid. ■

Proof. Aiming at a contradiction assume that there is an $\mathsf{LC}^{\mathsf{g}}_L$-proof δ of Γ, but that Γ is not L-valid. Given that Γ is not L-valid, δ_0, the $\mathsf{LC}^{\mathsf{g}}_L$-derivation comprised of Γ as root sequent and only node, must have an L-countermodel. By repeated application of Lemma 4.21 on δ_0, every resulting $\mathsf{LC}^{\mathsf{g}}_L$-derivation from application of inferences on δ_0 must have an L-countermodel. Thus δ must have an L-countermodel.

By the assumption that δ is an $\mathsf{LC}^{\mathsf{g}}_L$-proof of Γ, every branch is closed, i.e., every branch contains an axiom. By Lemma 4.12 no axiom is satisfiable, so there can be no L-countermodel for any branch of δ. This contradicts the fact that δ has an L-countermodel and the assumption of Γ not being L-valid; hence, Γ is L-valid. □

4.2 Completeness

Completeness of the $\mathsf{LC}^{\mathsf{g}}_L$-calculus is the result that we are able to prove every L-valid sequent in $\mathsf{LC}^{\mathsf{g}}_L$. This means showing that every L-valid sequent is $\mathsf{LC}^{\mathsf{g}}_L$-provable. What we do is to show the equivalent contrapositive argument, that if a sequent is not $\mathsf{LC}^{\mathsf{g}}_L$-provable, it is not L-valid, i.e., it has an L-countermodel.

To formalize the repetitive application of inferences to a skeleton, we define the notion of an LC_L-*derivation construction rule*.

Definition 4.23 (LC_L-derivation construction rule)
The application of an LC_L-derivation construction rule $\mathcal{R}$ on an LC_L-derivation δ_0 produces a sequence of LC_L-derivations $\langle \delta_0, \delta_1, \ldots, \delta_n, \delta_{n+1}, \ldots \rangle$ where δ_{i+1} is the resulting LC_L-derivation from applying an admissible inference to δ_i, $i \geq 0$.

If the sequence a LC_L-derivation construction rule $\mathcal{R}$ produces is finite, the last derivation in the sequence is called the *limit object* of $\mathcal{R}$. If the sequence of $\mathcal{R}$ is infinite, the limit object is the minimal derivation that the sequence approximates. ■

To be able to conclude that no proof can be found, we need to know that every possibility is examined. We do so by requiring that a *fair* construction rule is used. The following definition is a modified version of the ones found in [Fitting 1996, Hansen 2005].

Definition 4.24 (Fairness)
An LC_L-derivation construction rule $\mathcal{R}$ is fair provided that the sequence of LC_L-derivations $\langle \delta_0, \delta_1, \ldots \rangle$ which $\mathcal{R}$ constructs, is such that the following holds for every δ_i in the sequence:

1. An inference is eventually applied to every non-atomic formula in every branch of δ_i.

2. An inference introducing s is eventually applied to every ν-formula for every label s admissible in LC_L and for every branch in δ_i on which the formula occurs. ■

The proof of completeness goes by the following lines: First we assume that the $\mathsf{LC}^{\mathsf{g}}_{L}$-derivation at hand is not $\mathsf{LC}^{\mathsf{g}}_{L}$-provable. A fair $\mathsf{LC}^{\mathsf{g}}_{L}$-derivation construction rule lets us conclude that we have applied every admissible inference to the derivation. Since the derivation is not a proof, there must be an open branch in the derivation. This branch is the basis for the construction of an L-countermodel. As the root sequent occurs in every branch of the $\mathsf{LC}^{\mathsf{g}}_{L}$-derivation, the constructed L-countermodel is necessarily also an L-countermodel for the root sequent.

Theorem 4.25 (Completeness of $\mathsf{LC}^{\mathsf{g}}_{L}$)
If a labelled sequent is not $\mathsf{LC}^{\mathsf{g}}_{L}$-provable, it has an L-countermodel. ∎

Proof. Let Γ be an empty labelled sequent. Apply a fair $\mathsf{LC}^{\mathsf{g}}_{L}$-derivation construction rule to Γ producing the limit object δ_∞. Since Γ is not L-provable, there must be an open branch in δ_∞. Let θ denote this open branch and let θ^{fml} be the set of formulae occurring on θ.

Let $\theta^{\mathsf{Par}}_{L}$ be the set of ground labels in Par_L occurring on θ. From the open branch θ we construct a model $(\mathcal{M}, \phi)$, where $\mathcal{M} = (W, W^+, V)$, by the following. Let

- W^+ be the set ground labels $t \in \theta^{\mathsf{Par}}_{L}$,
- $W = W^+ \cup \{\epsilon\}$,
- $\theta^{\mathsf{At.fml}}_{\top}$ denote the set of atomic formulae in θ having polarity $\top$,
- V be such that $c \in V(F)$ iff $F[c] \in \theta^{\mathsf{At.fml}}_{\top}$,
- ϕ be the identity function, i.e., $\phi(s) = s$ for all labels s.

Now we claim that this constructed L-model is an L-model satisfying the set of formulae occurring on θ.

Claim. Every formulae in θ^{fml} is true in $(\mathcal{M}, \phi)$. ∘

Proof. By induction on the formulae in θ^{fml}.

Base step. Assume F is an atomic formula with polarity $\top$, e.g. the formula $Q[s]_\top$. $Q[s]_\top$ is true in $(\mathcal{M}, \phi)$, $(\mathcal{M}, \phi) \models Q[s]_\top$, if and only if $\mathcal{M}, \phi(s) \models Q$ if and only if $\phi(s) \in V(Q)$ and $s \in W$. This holds by construction of $(\mathcal{M}, \phi)$.

Now assume that F is an atomic formula having polarity $\bot$. Then $(\mathcal{M}, \phi) \models Q[s]_\bot$ if and only if $\mathcal{M}, \phi(s) \not\models Q$ if and only if $\phi(s) \notin V(Q)$ and $s \in W$. This holds by construction of $(\mathcal{M}, \phi)$.

Induction step.

α. Let F be an α-formula. Given the assumption that a fair $\mathsf{LC}^{\mathsf{g}}_L$-derivation construction rule has been applied, the components of F, α_1 and α_2 must occur on θ. By the induction hypothesis they are both true in the L-model $(\mathcal{M}, \phi)$. Now conclude by Corollary 4.20; F is true in $(\mathcal{M}, \phi)$.

β. Assume F is a β-formula. Since a fair $\mathsf{LC}^{\mathsf{g}}_L$-derivation construction rule has been applied, one of the components of F, β_i $(i = 1, 2)$, must occur on θ. By the induction hypothesis β_i is true in the L-model $(\mathcal{M}, \phi)$ and by Corollary 4.20 F is true in $(\mathcal{M}, \phi)$

ν. Let F be a ν-formula. Since we have applied a fair $\mathsf{LC}^{\mathsf{g}}_L$-derivation construction rule, $F_0(t)$, where $F_0(t)$ is the component of F, must occur on θ for every label parameter $t \in \theta^{\mathsf{Par}}_L$. By the induction hypothesis every F_0-formula occurring on the branch θ is true in $(\mathcal{M}, \phi)$, i.e., $(\mathcal{M}, \phi) \models F_0[t]$ for every $t \in \theta^{\mathsf{Par}}_L$. Since θ^{Par}_L is exactly the set of points in W^+ we conclude that F is true in $(\mathcal{M}, \phi)$.

π. Assume F is of type π. Then, by the application of a fair $\mathsf{LC}^{\mathsf{g}}_L$-derivation construction rule, the component of F, F_0, occurs on θ and is true in $(\mathcal{M}, \phi)$ by the induction hypothesis. By Corollary 4.20 we conclude F is true in $(\mathcal{M}, \phi)$. □

Since the root sequent occurs in every branch, an L-countermodel for a branch in an $\mathsf{LC}^{\mathsf{g}}_L$-derivation is automatically an L-countermodel for the root sequent of the same $\mathsf{LC}^{\mathsf{g}}_L$-derivation. By this observation, given that every formula in θ^{fml} is true in $(\mathcal{M}, \phi)$, every formulae in the empty labelled sequent Γ must be true, hence $(\mathcal{M}, \phi)$ is an L-countermodel for Γ. □

Chapter 5

The Calculus $\mathsf{LC^{gt}}$

The $\mathsf{LC^{gt}}$-calculus is an intermediate step towards the realization of the free-variable calculus $\mathsf{LC^{fv}}$ defined in the next chapter. As we saw in the proof of completeness of $\mathsf{LC^{g}}$, the possibility of obtaining infinite $\mathsf{LC^{g}}$-derivations is present. The groundwork for the removal of this possibility is laid by the $\mathsf{LC^{gt}}$-calculus and is in the next chapter implemented by the $\mathsf{LC^{fv}}$-calculus. The termination result established is also crucial for the development of the calculus for the logic of Only Knowing. The $\mathsf{LC^{gt}}$-calculus is a ground calculus, and *termination* is reason for the t in $\mathsf{LC^{gt}}$, although we do not establish the termination results for the $\mathsf{LC^{gt}}$-calculus.

We need some kind of indexing of formulae to keep track of the number of copies of a formula and thereby be able to prevent an infinite replication of formulae. What we do is to associate a copy history to every formula occurring in a sequent.

Definition 5.1 (Indexed formula)

An *indexed formula* is a formula of the form F^{κ} where κ is a *copy history* and F is a labelled formula. Copy histories are sequences of natural numbers. The *index* of an indexed formula F^{κ} is the pair (i, κ), where i and κ are respectively the formula number and copy history of F. An indexed formula F^{κ} is atomic (non-atomic) if the labelled formula F is atomic (non-atomic). The type of F^{κ} is the same as the type of F. ▪

The definitions of indexed formula in [Antonsen and Waaler 2006, Hansen 2004], both inspired by the indexing system of formulae used in [Wallen 1990], amount to the same definition above, except the underlying formulae are first-order. The intuition behind indices is that formula numbers identify *labelled* formulae of the same form, and copy histories keep track of the explicit copying of formulae. Together they form an index capable of identifying formulae of the same form occurring in the same context. Later we will see that indexed formulae with the same index are contextually equivalent.

Notation. When the copy history of F^κ is irrelevant we will just denote the formula by F, but we will then explicate that the formula is in fact an indexed formula (and not a core or labelled formula). When identification of the formula number is needed it will be placed below the formula, $\underset{i}{F^\kappa}$. When F^κ is a non-atomic formula, the formula number will be placed under the main connective/modal operator of F. Copy histories will be written $n_1.n_2.\ldots.n_m$. If κ is a copy history, $\kappa + n$, where n is a natural number, denotes the copy history obtained by adding n to the last number of κ. $\kappa.\kappa'$ denotes the copy history resulting from concatenating the two the copy histories κ and κ'. If a copy history is 0, it will not be displayed. ◦

Two indexed formulae are syntactically equal only if their labelled formulae and indices are equal and their subformulae are assigned formula numbers in the same way.

Example 5.2 (Indexed formula) The following formulae are all examples of distinct indexed formulae:

- $\underset{10}{\Diamond}(\underset{6}{P} \underset{4}{\rightarrow} \underset{2}{\Box}\underset{9}{P})[c]_\top$
- $(\underset{1}{P} \underset{2}{\rightarrow} (\underset{3}{Q} \underset{4}{\vee} (\underset{5}{P} \underset{6}{\wedge} \underset{7}{R})))[s]^2_\top$
- $\underset{6}{\Box}\underset{4}{Q}[U]^{2.2}_\top$ (1)
- $\underset{6}{\Box}\underset{4}{Q}[U]^{1.0}_\top$ (2)
- $\underset{2}{\Box}\underset{3}{Q}[U]^{2.2}_\top$ (3)

Notice especially that the three last formulae, marked (1), (2) and (3) are different: The two formulae (1) and (2) differ since their copy histories are not equal, and (1) and (3) are distinct given that they have different formula numbers. ◦

Definition 5.3 (Immediate subformulae)
Let X^κ and $Y^{\kappa'}$ be two indexed formulae. We write $X^\kappa <_1 Y^{\kappa'}$ and say that $Y^{\kappa'}$ is an immediate subformula of X^κ only if $X <_1 Y$ holds for the labelled formulae X and Y, and $\kappa' = \kappa$ or $\kappa' = \kappa.n$, where $n \geq 0$. ■

As done with core and labelled formulae, we use the definition of immediate subformulae together with the subformula relations defined in Definition 2.5 to define the notions of subformulae and proper formulae also for indexed formulae.

Example 5.4 (Subformulae) Let ψ denote $\Box\Box Q[U]^0_\top$. The formula $\Box Q[s]^{0.0}_\top$ is an immediate subformula of ψ, and $\Box Q[s]^{0.1}_\top$ is also an immediate subformula of ψ, since the labelled formula $\Box Q[s]_\top$ is an immediate subformula of the labelled formula $\Box\Box Q[s]_\top$ and the copy history of the proposed subformula is $\kappa.0$, where κ is the copy history of ψ. The formula $\Box Q[s]^{0.1}_\top$ is *not* a subformula of $\Box Q[s]^{0.0}_\top$, given that their labelled formulae are not immediate subformulae and that 0.0 is not a prefix of 0.1.

Every subformulae of $\Box Q[s]^0_\top$ and $\Box Q[s]^1_\top$ is a subformula of ψ, while there is no subformula relation between any pair in the sets of subformulae of $\Box Q[s]^0_\top$ and $\Box Q[s]^1_\top$. ◦

Remark (to Example 5.4). Figure 5.2 displays the above information visually. The reader may consult this figure now, but note that the figure also illustrates the rules of $\mathsf{LC}^{\mathsf{gt}}$ which are established on the next page. ◦

The satisfaction relation defined on labelled formulae and models of the labelled language (Definition 3.11) are easily extended to indexed formulae.

Definition 5.5 (Truth)

Let $\vDash$ be the relation defined in Definition 3.11 and $(\mathcal{M}, \phi)$ be an L-model for the labelled language. An indexed formula F^κ is satisfied in $(\mathcal{M}, \phi)$, $(\mathcal{M}, \phi) \vDash F^\kappa$, if and only if the labelled formula F is satisfied in $(\mathcal{M}, \phi)$, $(\mathcal{M}, \phi) \vDash F$. ▪

Using this definition, the notions of *satisfaction* for sets of indexed formula, and *satisfiable* and *true* for both indexed formulae and sets of indexed formula are obvious and are left for the reader.

Example 5.6 (Truth) Recall the formulae in Example 5.2 marked (1), (2) and (3). These formulae differ because they have different indices, but if one of the formula is satisfied in a model $(\mathcal{M}, \phi)$, then all of the indicated formulae is satisfied in $(\mathcal{M}, \phi)$ since they are identical when removing formula numbers and copy histories. ◦

Definition 5.7 (Indexed sequent)

An indexed sequent is a *set* of indexed formulae. An *empty labelled indexed sequent* is an indexed sequent in which every formula is labelled with ϵ, the copy history of every formula is 0 and every formula number occurring in the indexed sequent is distinct. An unlabelled formula occurring in an indexed sequent is by default labelled with ϵ. ▪

Example 5.8 (Empty labelled indexed sequent) The following two indexed sequents are empty labelled. Remember that we by default do not display copy histories equal to 0.

$$\underset{1}{P}, \underset{2}{Q} \underset{3}{\to} \underset{4}{P}, \underset{5}{P} \vdash \underset{6}{R}, \underset{7}{D} \qquad \underset{1}{P} \underset{4}{\to} (\underset{2}{Q} \underset{5}{\wedge} \underset{3}{R}) \vdash \underset{9}{P} \underset{7}{\wedge} \underset{6}{R}$$

The indexed sequent

$$\underset{1}{P} \underset{4}{\to} (\underset{2}{Q} \underset{5}{\wedge} \underset{3}{R}) \vdash \underset{9}{P} \underset{1}{\wedge} \underset{6}{R}^1$$

is *not* empty labelled for two reasons: The copy history of every formula is not 0 and not all formula numbers are distinct as 1 is the formula number of two formulae, P and $P \wedge R$. ◦

$$\frac{\Gamma, \alpha_1^\kappa, \alpha_2^\kappa}{\Gamma, \alpha^\kappa} \qquad \frac{\Gamma, \beta_1^\kappa \quad \Gamma, \beta_2^\kappa}{\Gamma, \beta^\kappa} \qquad \frac{\Gamma, \underset{i}{\nu}^{\kappa+1}, \nu_0(s)^{\kappa.0}}{\Gamma, \underset{i}{\nu}^\kappa}\, s \qquad \frac{\Gamma, \pi_0^\kappa(c_i)}{\Gamma, \underset{i}{\pi}^\kappa}\, c_i$$

Figure 5.1: Rules of $\mathsf{LC}^{\mathsf{gt}}$. Γ is a set of indexed formulae. The label s in the ν-rule is an arbitrary label, while the label parameter c_i occurring in the π-rule is the letter c subscripted with the formula number of the principal formula π.

The rules of $\mathsf{LC}^{\mathsf{gt}}$ are listed in Figure 5.1. All formulae are indexed formulae of the indicated type. Γ is a placeholder for a possibly empty set of indexed formulae. No rule introduces new formula numbers. Only ν-inferences manipulate copy histories, setting the copy history of the active ν-formula equal to the copy history of the principal ν-formula incremented by one and adding a new last number 0 to the copy history to the ν_0-formula. This means that from a formula's copy history one can read how many times it has gone through an explicit copy—perhaps as the subformula of the formula being copied—and how many ν-formulae it is a subformula of.

The other important thing to notice in the rules of $\mathsf{LC}^{\mathsf{gt}}$ is that the π-rule introduces a parameter c_i where i is the formula number of the principal formula of the inference. This means that two (differently) indexed π-formulae with equal formula numbers will introduce the same label parameter. This permits a reduction in the amount of label parameters occurring in a skeleton and is the key factor in establishing the termination bounds of the calculus $\mathsf{LC}^{\mathsf{fv}}$.

Remark. In [Beckert and Goré 1997] π-formulae introduce the *gödelization* of itself as label. This is not too different from our approach, as the formula number can be interpreted as the gödelization of the formula. However, we formula enumerate every formula differently—even formulae where

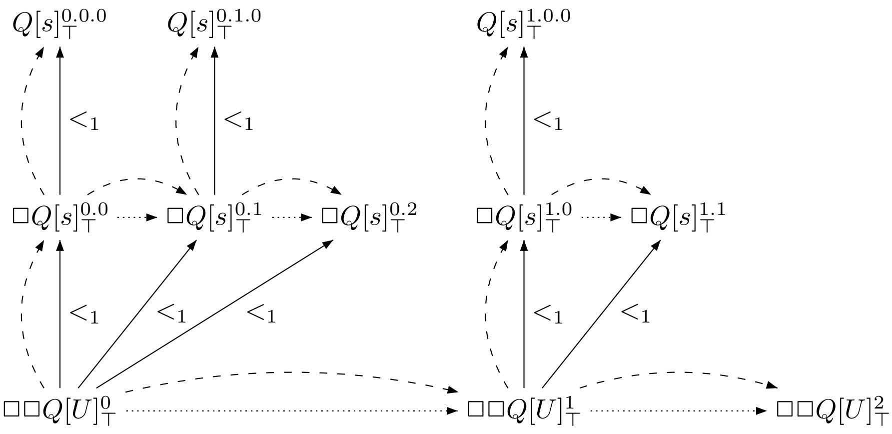


Figure 5.2: Copy histories and subformulae. The solid arrows denote the immediate subformula relation, the dotted arrows explicit copying, and the dashed lines depict how formulae are generated by rule application. All formulae in the figure are ν-formulae. There are two active formulae in a ν-inference, so every inference is in the figure denoted by two outwards dashed arrows.

their labelled formulae are syntactically equal, forcing all differently formula numbered π-formulae to introduce different labels. ◦

Consult Figure 5.2. What we can see from the figure, and as was exemplified in Example 5.4, is how copy histories control the subformula relation: if a formula X is explicitly copied, then no subformula relation between X and its explicit copy X' is formed, nor between X and any of the resulting formulae from rule application on X'. But if Y is a proper subformula of X, then every formula, even explicitly copied formulae, resulting from rule application on Y is a subformula of X.

Definition 5.9 ($\mathsf{LC}^{\mathsf{gt}}_L$-derivation)

A skeleton of indexed sequents having an empty labelled root is an $\mathsf{LC}^{\mathsf{gt}}_L$-derivation if it respects the following conditions set by the logic L:

- every inference is an instance of a rule in the $\mathsf{LC}^{\mathsf{gt}}$-calculus,
- the *contextually equivalent condition:* The inferences r and r' belong to the same label if the respective inferences are contextually equivalent,
- the *ground label condition:* A ν-inference may only introduce ground labels $t \in \mathsf{Par}_L$,

- the *neighbourhood condition:* A ν-inference may only introduce the ground label s if it already occurs in the branch or if s belongs to a ν-inference in a different branch, and
- the *non-empty W^+ condition*: A ν-inference r is applicable only if there is a label $s \in \mathsf{Par}_L$ in r. ∎

Compare the above definition to the definition of an $\mathsf{LC}^{\mathsf{g}}_L$-derivation (Definition 4.7) and observe that an $\mathsf{LC}^{\mathsf{gt}}_L$-derivation does not require a skeleton to respect the eigenparameter condition, nor does the new neighbourhood condition added in Definition 5.9 preserve the immediate effects of the missing condition. Since the label parameter that a π-inference introduces is governed by a formula number, adding side conditions to ensure the "freshness" of the label makes no sense, thus an eigenparameter condition is in this case of no use. Instead we could have made sure that the label was fresh by restricting the label introduction by ν-formulae to only "unfresh" labels, i.e., labels already occurring in the branch. But, as there already is a restriction on what ground labels a ν-inference may introduce, namely the contextually equivalent condition, such a "freshness" condition may cause a "deadlock" in the proof search: A ν-inference may not be applicable as it must by the contextually equivalent condition introduce a certain label, but this violates the unfreshness-condition as the label does not occur in the branch. An example of such a deadlock is displayed below.

Example 5.10 (Freshness condition in $\mathsf{LC}^{\mathsf{gt}}$ causes deadlock) This example illustrates a deadlock situation obtained if a freshness condition equivalent to the eigenparameter condition were to be upheld in $\mathsf{LC}^{\mathsf{gt}}$. The sequent $\Box P \vdash \Box\Box P \wedge \Box\Box P$ is K45-valid. For reasons of readability only the formula numbers of π-formulae are indicated. Copy histories are neglected.

$$\dfrac{\dfrac{\dfrac{\dfrac{\overset{\times}{\Box P, P[c_2] \vdash P[c_2]}}{\Box P \vdash P[c_2]}\,c_2}{\Box P \vdash \underset{2}{\Box} P[c_1]}\,c_2}{\Box P \vdash \underset{1}{\Box}\underset{2}{\Box} P}\,c_1 \qquad \dfrac{\dfrac{\overset{\circ}{\Box P \vdash P[c_4]}}{\Box P \vdash \underset{4}{\Box} P[c_3]}\,c_4}{\Box P \vdash \underset{3}{\Box}\underset{4}{\Box} P}\,c_3}{\Box P \vdash \underset{1}{\Box}\underset{2}{\Box} P \wedge \underset{3}{\Box}\underset{4}{\Box} P}$$

Observe that the ν-formula in the right leaf sequent is caught in a deadlock: it cannot introduce the label c_4, which would close the skeleton since it's contextually equivalent formula in the other branch belongs to c_2, and it cannot introduce c_2 as it does not occur in the right branch, making the search come to a halt. Since the sequent is valid, this example indicates a lack of completeness of an $\mathsf{LC}^{\mathsf{gt}}$-calculus where a freshness condition equivalent to the eigenparameter condition is included.

The skeleton below displays that if we instead use the neighbourhood condition as a freshness condition, a continuation in the search is granted.

$$\dfrac{\dfrac{\dfrac{\dfrac{\overset{\times}{\Box P, P[c_2] \vdash P[c_2]}}{\Box P \vdash P[c_2]}\,c_2}{\Box P \vdash \underset{2}{\Box} P[c_1]}\,c_2}{\Box P \vdash \underset{1}{\Box}\underset{2}{\Box} P}\,c_1 \qquad \dfrac{\dfrac{\dfrac{\dfrac{\overset{\times}{\Box P, P[c_4], P[c_2] \vdash P[c_4]}}{\Box P, P[c_2] \vdash P[c_4]}\,c_4}{\Box P \vdash P[c_4]}\,c_2}{\Box P \vdash \underset{4}{\Box} P[c_3]}\,c_4}{\Box P \vdash \underset{3}{\Box}\underset{4}{\Box} P}\,c_3}{\Box P \vdash \underset{1}{\Box}\underset{2}{\Box} P \land \underset{3}{\Box}\underset{4}{\Box} P}$$

The introduction of c_2 is now allowed in the right branch by the neighbourhood condition, since the label already belongs to a ν-inference in the other branch. The introduction of c_2 in the right branch is enforced by the contextually equivalent condition, but in the uppermost inference in right branch, the ν-inference may, by the neighbourhood condition introduce any of the label parameters c_2, c_3, c_4, and since the principal formula of this inference is not contextually equivalent to any other formula in the skeleton, it may freely choose which of these labels to introduce. $\circ$

Remark. As we saw in Example 4.17, the eigenparameter conation is a necessary condition to ensure soundness of LC^{g}. Is it safe to remove this condition from $\mathsf{LC}^{\mathsf{gt}}$? What remedies the situation are two things:

1. the freshness condition the neighbourhood condition constitutes: *A ν-inference may only introduce the ground label s if it already occurs in the branch or if s belongs to a ν-inference in a different branch.* This means that every label parameter in an $\mathsf{LC}^{\mathsf{gt}}$-derivation is first introduced to the skeleton by a π-inference, and
2. the new π-rule in $\mathsf{LC}^{\mathsf{gt}}$, which hinders two not contextually equivalent π-formulae to introduce equal label parameters.

This is in fact enough to preserve soundness of $\mathsf{LC}^{\mathsf{gt}}$ despite the removal of the eigenparameter condition. $\circ$

Having defined $\mathsf{LC}^{\mathsf{gt}}_L$-derivation we are ready to establish results concerning derivations in $\mathsf{LC}^{\mathsf{gt}}$ and the indices of formulae.

Lemma 5.11 Let X and Y be indexed formulae with equal formula numbers occurring in the $\mathsf{LC}^{\mathsf{gt}}$-derivation. The formulae are equal up to labelling and copy history. ■

Proof. Since every formula and subformula occurring in the empty labelled root of an $\mathsf{LC}^{\mathsf{gt}}$-derivation are distinctly formula numbered and no inference manipulates formula numbers—through either implicit or explicit copy of formulae—the claim holds. □

The following corollary is used in the proof of soundness of $\mathsf{LC}^{\mathsf{gt}}$ together with the contextually equivalent condition and the neighbourhood condition to preserve the effects of the eigenparameter condition.

Corollary 5.12 If two indexed π-formulae X and Y in an $\mathsf{LC}^{\mathsf{gt}}$-derivation have equal formula numbers, their respective π_0-formulae, X_0 and Y_0 are identical up to copy history. ■

Proof. By Lemma 5.11 X and Y are identical up to labelling and copy history. Then X_0 and Y_0 are also identical up to labelling and copy history as they have the same index. By the π-rule in $\mathsf{LC}^{\mathsf{gt}}$, X_0 and Y_0 are labelled using the same label parameter, given that X and Y have equal formula numbers. □

Remark (Conservation of the eigenparameter condition). This remark gives insights to why neglecting to include the eigenparameter condition in $\mathsf{LC}^{\mathsf{gt}}$ is not hazardous for the soundness of $\mathsf{LC}^{\mathsf{gt}}$. Assume the eigenparameter condition is violated by a $\mathsf{LC}^{\mathsf{gt}}$-derivation letting a π-inference r introducing a label parameter c occurring in the denominator. If the label c in the denominator belongs to a π-inference r', when r and r' have equal formula numbers and their active formulae are identical up to copy history, by Corollary 5.12. If c belongs to a ν-inference, there must, by the neighbourhood condition, be a π-inference in the skeleton with the same formula number as the principal formula of r, which has introduced c to the skeleton when c was fresh denominator. These results are vital for establishing soundness of $\mathsf{LC}^{\mathsf{gt}}$. ◦

The indexing of formulae is only used for keeping track of copies and contextually equivalent formulae, and thereby also the admissibility of derivations and proofs. There is no need to revise the semantical interpretation of indexed formulae: An indexed formula is satisfiable if and only if its underlying labelled formula is satisfiable, i.e., when examining the truth of indexed formulae one can treat the formulae as labelled formula by disregarding their copy histories all together. The same method is used when identifying axioms in $\mathsf{LC}^{\mathsf{gt}}$.

Definition 5.13 (Axiom, $\mathsf{LC}^{\mathsf{gt}}$)

An axiom in $\mathsf{LC}^{\mathsf{gt}}$ is an indexed sequent Γ such that the labelled sequent obtained by disregarding all copy histories in Γ is an axiom in LC^{g}. ▪

It immediately follows by Lemma 4.12 that an axiom in $\mathsf{LC}^{\mathsf{gt}}$ is not satisfiable.

Definition 5.14 ($\mathsf{LC}^{\mathsf{gt}}_{L}$-proof)

An $\mathsf{LC}^{\mathsf{gt}}_{L}$-proof is an $\mathsf{LC}^{\mathsf{gt}}_{L}$-derivation where there in every branch exists an axiom. ▪

When calling an indexed formula or sequent *labelled* we address the labelled formula or labelled sequent gathered by removing every copy history from the indexed object in question.

Example 5.15 ($\mathsf{LC}^{\mathsf{gt}}_{S5}$-proof, ground label condition, axiom) This is an $\mathsf{LC}^{\mathsf{gt}}_{S5}$-proof where its labelled root sequent is the same as the root sequent found in Example 4.18.

$$\dfrac{\dfrac{\dfrac{\overset{\times}{\underset{2}{P}[c_1] \vdash \underset{4}{P}[c_1]}}{\vdash \underset{2}{P} \underset{3}{\rightarrow} \underset{4}{P}[c_1]}}{\vdash \underset{1}{\Box}(\underset{2}{P} \underset{3}{\rightarrow} \underset{4}{P})}\,c_1 \qquad \dfrac{\dfrac{\overset{\times}{\underset{7}{P}^{0.0} \vdash \underset{9}{P}^{0.0}, \underset{6}{\Diamond}(\underset{7}{P} \underset{8}{\rightarrow} \underset{9}{P})^{1}}}{\vdash \underset{7}{P} \underset{8}{\rightarrow} \underset{9}{P}^{0.0}, \underset{6}{\Diamond}(\underset{7}{P} \underset{8}{\rightarrow} \underset{9}{P})^{1}}}{\vdash \underset{6}{\Diamond}(\underset{7}{P} \underset{8}{\rightarrow} \underset{9}{P})}\,\epsilon}{\vdash \underset{1}{\Box}(\underset{2}{P} \underset{3}{\rightarrow} \underset{4}{P}) \underset{5}{\wedge} \underset{6}{\Diamond}(\underset{7}{P} \underset{8}{\rightarrow} \underset{9}{P})}$$

The differences between the two proofs, except the obvious, that indexed sequents are indexed and labelled sequents are not, is the difference in the introduced labels. The label parameter introduced in the left branch is and must be c_1 since 1 is the formula number of the π-formula which introduces the label. In the right branch, the ν-formula with index $(6, 0)$ can only introduce ϵ given that ϵ is the only label occurring in the branch at this stage. As a result the derivation is an $\mathsf{LC}^{\mathsf{gt}}_{S5}$-proof. The skeleton is no $\mathsf{LC}^{\mathsf{gt}}_{K45}$-derivation and thus no $\mathsf{LC}^{\mathsf{gt}}_{K45}$-proof as it violates the ground label condition in K45 by introducing ϵ.

Observe that the indexed sequents $\underset{2}{P}[c_1] \vdash \underset{4}{P}[c_1]$ and $\underset{7}{P}^{0.0} \vdash \underset{9}{P}^{0.0}, \underset{6}{\Diamond}(\underset{7}{P} \underset{8}{\rightarrow} \underset{9}{P})^{1}$ are axioms even though their closing pairs are differently indexed. ◦

Example 5.16 ($\mathsf{LC}^{\mathsf{gt}}_{S5}$-proof, neighbourhood-, non-empty W^{+} cond.) Recall Example 4.19. This is the same example adapted to the $\mathsf{LC}^{\mathsf{gt}}$-calculus. χ is an abbreviation for $\underset{1}{\Diamond}(\underset{2}{P} \underset{3}{\rightarrow} \underset{4}{\Box}\underset{5}{P})$.

$$\times$$
$$\underset{2}{P}[c_6]^{0.0}, \underset{2}{P}[c_4]^{1.0} \vdash \underset{5}{P}[c_4]^{0.0}, \underset{4}{\Box}\underset{5}{P}[c_4]^{1.0}, \chi^2, \underset{7}{Q}[c_6]$$
$$\overline{\underset{2}{P}[c_6]^{0.0} \vdash \underset{5}{P}[c_4]^{0.0}, \underset{2}{P} \underset{3}{\rightarrow} \underset{4}{\Box}\underset{5}{P}[c_4]^{1.0}, \chi^2, \underset{7}{Q}[c_6]} \; c_4$$
$$\overline{\underset{2}{P}[c_6]^{0.0} \vdash \underset{5}{P}[c_4]^{0.0}, \chi^1, \underset{7}{Q}[c_6]} \; c_4$$
$$\overline{\underset{2}{P}[c_6]^{0.0} \vdash \underset{4}{\Box}\underset{5}{P}[c_6]^{0.0}, \chi^1, \underset{7}{Q}[c_6]}$$
$$\overline{\vdash \underset{2}{P} \underset{3}{\rightarrow} \underset{4}{\Box}\underset{5}{P}[c_6]^{0.0}, \chi^1, \underset{7}{Q}[c_6]} \; c_6$$
$$\overline{\vdash \underset{1}{\Diamond}(\underset{2}{P} \underset{3}{\rightarrow} \underset{4}{\Box}\underset{5}{P}), \underset{7}{Q}[c_6]} \; c_6$$
$$\overline{\vdash \underset{1}{\Diamond}(\underset{2}{P} \underset{3}{\rightarrow} \underset{4}{\Box}\underset{5}{P}), \underset{6}{\Box}\underset{7}{Q}}$$

Notice that as opposed to what was the case in the LC$^{\mathsf{g}}_{S5}$-derivation in Example 4.19, where we by removing $\Box Q$ and its subformulae occurring in the skeleton obtained an LC$^{\mathsf{g}}_{S5}$-proof of $\vdash \Diamond(P \rightarrow \Box P)$, the removal of $\Box Q$ and $Q[c_6]$ in the current example does not result in an LC$^{\mathsf{gt}}_{S5}$-proof. The reason for this is that the neighbourhood condition makes the introduction of c_6 by the ν-formula $\underset{1}{\Diamond}(\underset{2}{P} \underset{3}{\rightarrow} \underset{4}{\Box}\underset{5}{P})_{\perp}$ illegal. But clearly as the formula is LC$^{\mathsf{g}}_{S5}$-provable, it should also be LC$^{\mathsf{gt}}_{S5}$-provable, given that a L-countermodel for an LC$^{\mathsf{g}}_{L}$-derivation is also an L-countermodel for an LC$^{\mathsf{gt}}_{L}$-derivation. The remedy is that the introduction of ϵ by a ν-formula in LC$^{\mathsf{gt}}_{S5}$ is always legal, since $\epsilon \in \mathsf{Par}_{S5}$ occurs in every branch of every LC$^{\mathsf{gt}}$-derivation. An LC$^{\mathsf{gt}}_{S5}$-proof of $\Diamond(P \rightarrow \Box P)_{\perp}$ is obtained by the following LC$^{\mathsf{gt}}_{S5}$-derivation.

$$\times$$
$$\underset{2}{P}^{0.0}, \underset{2}{P}[c_4]^{1.0} \vdash \underset{5}{P}[c_4]^{0.0}, \underset{4}{\Box}\underset{5}{P}[c_4]^{1.0}, \chi^2$$
$$\overline{\underset{2}{P}^{0.0} \vdash \underset{5}{P}[c_4]^{0.0}, \underset{2}{P} \underset{3}{\rightarrow} \underset{4}{\Box}\underset{5}{P}[c_4]^{1.0}, \chi^2}$$
$$\overline{\underset{2}{P}^{0.0} \vdash \underset{4}{\Box}\underset{5}{P}^{0.0}, \chi^1}$$
$$\overline{\vdash \underset{2}{P} \underset{3}{\rightarrow} \underset{4}{\Box}\underset{5}{P}^{0.0}, \chi^1} \; \epsilon$$
$$\overline{\vdash \underset{1}{\Diamond}(\underset{2}{P} \underset{3}{\rightarrow} \underset{4}{\Box}\underset{5}{P})}$$

The skeleton below displays the attempt of proving $\vdash \underset{1}{\Diamond}(\underset{2}{P} \underset{3}{\rightarrow} \underset{4}{\Box}\underset{5}{P})$ in LC$^{\mathsf{gt}}_{K45}$.

$$\circ$$
$$\vdash \underset{1}{\Diamond}(\underset{2}{P} \underset{3}{\rightarrow} \underset{4}{\Box}\underset{5}{P})$$

The skeleton is an LC$^{\mathsf{gt}}_{K45}$-derivation, but as we see the proof attempt had to be abandoned at a very early stage because of the non-empty W^{+} condition: There are no label parameters $s \in \mathsf{Par}_{K45}$ in the skeleton to be introduced by a ν-inference applied to the only formula in the root sequent. A K45-countermodel for the formula is a model where $W^{+} = \emptyset$. ◦

Observe that by removing the copy histories from the $\mathsf{LC}^{\mathsf{gt}}$-derivations in the above examples, thus transforming the indexed sequents into labelled sequents, the $\mathsf{LC}_L^{\mathsf{gt}}$-proofs in these examples transform into $\mathsf{LC}_L^{\mathsf{g}}$-proofs. Although it does not apply for all $\mathsf{LC}^{\mathsf{gt}}$-derivations, it is no coincidence. This observation forms the basis for the soundness and completeness proofs of $\mathsf{LC}^{\mathsf{gt}}$.

5.1 Soundness

The differences between LC^{g} and $\mathsf{LC}^{\mathsf{gt}}$ is that $\mathsf{LC}^{\mathsf{gt}}$ implements a greater reuse of label parameters than LC^{g}. This may seem as a restriction as we do not have the same choice of what label parameter to introduce by a π-inference, but it may just as well be though of as an added liberty since no eigenparameter condition needs to be enforced in $\mathsf{LC}^{\mathsf{gt}}$. The point is that soundness of $\mathsf{LC}^{\mathsf{gt}}$ does not immediately follow by soundness of LC^{g}, as it would if $\mathsf{LC}^{\mathsf{gt}}$ clearly was a restricted version of LC^{g}.

The proof of soundness of $\mathsf{LC}^{\mathsf{gt}}$ follows the same lines as the soundness proof of LC^{g}. To simplify the proof we introduce the notion of a *balanced* skeleton.

Definition 5.17 (Balanced skeleton)

Let δ be a skeleton and X be a formula in δ to which a rule is applied. The skeleton δ is balanced only if rule is applied to every contextually equivalent formula of X in δ. ▪

If a skeleton contains only one branch, it is trivially balanced. The notion is important when there are multiple branches in a skeleton. If such a skeleton is balanced we know that if an inference is applied on a formula in one branch, then there are contextually equivalent inferences applied in every branch. Notice that every skeleton can be balanced. If a skeleton δ is not balanced, then there are at least two branches in δ where one of the branches contains a formula to which a rule has been applied and an other branch contains a contextually equivalent formula analysed by no rule. Apply a rule to the latter formula and repeat the process until the skeleton is balanced.

Lemma 5.18 (Countermodel preservation) Assume there is an L-countermodel for the branch θ in the $\mathsf{LC}_L^{\mathsf{gt}}$-derivation δ, and let Γ^{κ} be a leaf sequent of δ. Then there is an L-countermodel for the $\mathsf{LC}_L^{\mathsf{gt}}$-derivation δ', the skeleton resulting from applying an admissible inference on Γ^{κ}. ▪

Proof. If Γ is not the leaf of θ, the proof is trivial (see Lemma 4.21), so assume Γ *is* the leaf sequent of θ. Let θ' be the branch in δ' obtained from δ by adding a sequent of the numerator in the inference on Γ. We prove the lemma by checking for each possible type of inference r on Γ^{κ} producing the branch θ' in δ', that there is an L-countermodel for δ'.

α, β, ν. Assume the inference r is of either type α, β or ν. Observe that by removing the indices from the α-, β- and ν-rules of $\mathsf{LC}^{\mathsf{gt}}$ they are identical to the same rules in LC^{g}, and that the restrictions set by the conditions governing inferences of type α, β or ν in an $\mathsf{LC}^{\mathsf{gt}}$-derivation also meet the restrictions set by the conditions for the same inferences in an LC^{g}-derivation. By the definition of truth for indexed formulae (Definition 5.5) we know that every L-countermodel for the indexed sequent Γ^{κ} is an L-countermodel for the labelled sequent Γ, and vice versa. Apply Lemma 4.21 (Countermodel preservation [in LC^{g}]) and we are done.

π. The π-rule and the same as the conditions it has to abide by in $\mathsf{LC}^{\mathsf{gt}}$ are not as in LC^{g}, thus we cannot use the exact same argument as done for the types α, β and ν in this proof.

Let r be of type π and assume that δ is balanced. There are two cases to consider. The first case is the simplest one: Assume the label parameter r introduces is fresh to the denominator, then r enforced the eigenparameter condition in LC^{g}, and the claim holds by the proof of Lemma 4.21.

Now the second case. Assume that the label parameter c introduced by r is *not* fresh to the skeleton, and that the label c occurring in the denominator is introduced by a ν-formula. By the neighbourhood condition the label parameter must have been present in the skeleton prior to the introduction by the ν-formula. The initial introduction of c to δ, when c was fresh to the skeleton, must have been done by a π-inference r', also by conclusion of the neighbourhood condition. The active formulae of r and r' must, by Corollary 5.12, be equal up to copy history, since the principal formula of r and r' have equal formula numbers be the fact they introduce the same label. Given that δ is balanced, a formula X contextually equivalent to the active formula of r', must occur on θ. By Corollary 4.6 and 5.12, X and the active formula or r' are equal up to copy history. Then, as there by assumption is an L-countermodel for θ which satisfies X, the same L-model satisfies the active formula of r' and is an L-countermodel for θ'.

If the label parameter c introduced by r is not fresh because occurs in the denominator and has been introduced by a π-inference, the claim follows from a simplified version of the above argument. □

Theorem 5.19 (Soundness of $\mathsf{LC}^{\mathsf{gt}}$)

Let δ be an $\mathsf{LC}^{\mathsf{gt}}_L$-proof of an indexed sequent Γ. Then Γ is L-valid. ∎

Proof sketch. Utilize Lemma 5.18 and arrive at similar argumentation as in the proof of soundness of LC^{g} (Theorem 4.22). □

5.2 Completeness

Completeness of $\mathsf{LC}^{\mathsf{gt}}$ is established by adapting the proof of completeness in the previous chapter to $\mathsf{LC}^{\mathsf{gt}}$.

Theorem 5.20 (Completeness of $\mathsf{LC}^{\mathsf{gt}}$)

If a labelled sequent is not $\mathsf{LC}^{\mathsf{gt}}$-provable, it has an L-countermodel. ▪

Proof sketch. A proof is found by carrying out the exact same steps as in the proof of completeness of LC^{g} (Theorem 4.25): Let Γ be a sequent which is not $\mathsf{LC}^{\mathsf{gt}}_L$-provable. Then there is an open branch in the $\mathsf{LC}^{\mathsf{gt}}_L$-derivation obtained by application of a fair $\mathsf{LC}^{\mathsf{gt}}_L$-derivation construction rule. From this branch we construct an L-model—also done in the same manner as in Theorem 4.25—satisfying every formula on the branch. Since Γ is a sequent on this branch, the L-model is an L-countermodel for Γ. □

Chapter 6

The Calculus $\mathsf{LC}^{\mathsf{fv}}$

The $\mathsf{LC}^{\mathsf{fv}}$-calculus is a free-variable calculus of indexed formulae allowing tight control over both the relation of contextually equivalence and the introduction of label parameters by π-formulae. By utilizing label variables the calculus conforms to the least commitment requirement set in the introduction of this thesis by postponing the choice of label parameter to introduce by ν-formulae and later decide on what label parameter to introduce by a substitution of labels.

Throughout this chapter we compare the two calculi $\mathsf{LC}^{\mathsf{gt}}$ and $\mathsf{LC}^{\mathsf{fv}}$ alongside each other. This constitutes the scarlet thread of this chapter, making it apparent where the efficiency and flexibility of $\mathsf{LC}^{\mathsf{fv}}$ lies.

$$\frac{\Gamma, \alpha_1^\kappa, \alpha_2^\kappa}{\Gamma, \alpha^\kappa} \qquad \frac{\Gamma, \beta_1^\kappa \quad \Gamma, \beta_2^\kappa}{\Gamma, \beta^\kappa} \qquad \frac{\Gamma, \underset{i}{\nu}^{\kappa+1}, \nu_0(U_{(i,\kappa)})^{\kappa.0}}{\Gamma, \underset{i}{\nu}^\kappa}\; U_{(i,\kappa)} \qquad \frac{\Gamma, \pi_0^\kappa(c_i)}{\Gamma, \underset{i}{\pi}^\kappa}\; c_i$$

Figure 6.1: Rules of $\mathsf{LC}^{\mathsf{fv}}$. Γ is a set of indexed formulae. The label variable $U_{(i,\kappa)}$ in the ν-rule is the letter U subscripted with the index of the principal ν-formula, while the label parameter c_i occurring in the π-rule is the letter c subscripted with the formula number of the principal formula π.

The rules of the calculus $\mathsf{LC}^{\mathsf{fv}}$ are displayed in Figure 6.1. They are the same rules as for the $\mathsf{LC}^{\mathsf{gt}}$-calculus, only differing in the label introduced by a ν-inference: The $\mathsf{LC}^{\mathsf{fv}}$-calculus requires the introduced label to be a label variable and the label is always the label variable U subscripted with the index of the principal formula of the inference. Only ν-inferences manipulate copy histories, setting the copy history of the active ν-formula equal to the copy history of the principal ν-formula incremented by 1, and adding a new last digit 0 to the copy history of the active ν_0-formula.

Even though the formula numbers of the two ν-formulae occurring in the ν-rule are different, we still say that the ν-formula is explicitly copied from the denominator to the numerator.

Label variables are really a way of postponing the choice of label parameters introduced by ν-inferences. The choice is done by applying a *label substitution* to a set of label variables. A label substitution is a function mapping label variables to labels.

Definition 6.1 (Label substitution)
Let σ be a partial function $\sigma : \mathsf{Var} \to \mathsf{Var} \cup \mathsf{Par} \cup \{\epsilon\}$ and let $\textsc{dom}(\sigma)$ denote the set of label variables $U \in \mathsf{Var}$ such that $U \in \textsc{dom}(\sigma)$ if and only if $U\sigma \neq U$. We call σ a label substitution if it accepts the following conditions when extended to all labels:

1. $c\sigma = c$ for every $c \in \mathsf{Par} \cup \{\epsilon\}$, and
2. $U\sigma = U$, if $U \notin \textsc{dom}(\sigma)$. ∎

Let $X[s]$ and F be indexed formulae, Γ an indexed sequent and δ a skeleton. We extend a label substitution σ to indexed formulae, sequents and skeletons as follows;

- $X[s]\sigma = X[s\sigma]$,
- $\Gamma\sigma = \{F\sigma \mid F \in \Gamma\}$ and
- $\delta\sigma$ is the skeleton obtained by applying σ to every sequent in δ.

We say that a label substitution *grounds* or *is grounding for X*, if $X\sigma$ does not contain any label variables.

Remark. Throughout this chapter we will use the sequent

$$\underset{1}{\Diamond}(\underset{2}{P} \underset{3}{\to} \underset{4}{\Box}\underset{5}{P})_\perp, \underset{6}{\Box}\underset{7}{Q}_\perp$$

in examples. To focus attention on the notions being exemplified, skeletons with this sequent as root sequent may be simplified in one or more of the following ways: Indices may not displayed, more readable labels than subscripted c's and U's are introduced, and the explicit copies of ν-formulae may not be shown. In examples using this sequent, the letters c, b, U and V will respectively abbreviate the labels c_4, c_6, $U_{(1,0)}$ and $U_{(2,0)}$. Example 6.23 on page 78 displays a fully annotated skeleton over the above sequent. Fully annotated in the sense that all formula numbers, copy histories and explicit copies are shown, and labels are not abbreviated. ∘

Example 6.2 (Label substitution) Let $\Gamma = \{P[U], Q[V], P[V], R[c], S[W], Q[U]\}$ and $\sigma_2 = \{V \mapsto b, W \mapsto U\}$. Then $\Gamma\sigma_2 = \{P[U], Q[b], P[b], R[c], S[U], Q[U]\}$.

The two skeletons below illustrate the result of applying the label substitution $\sigma = \{U \mapsto b\}$ to a skeleton.

$$\underset{\delta}{\dfrac{\dfrac{\dfrac{\dfrac{P[U] \vdash P[c], Q[b]}{P[U] \vdash \Box P[U], Q[b]}\,c}{\vdash P \to \Box P[U], Q[b]}}{\vdash P \to \Box P[U], \Box Q}\,b}{\vdash \Diamond(P \to \Box P), \Box Q}\,U} \qquad \underset{\delta\sigma}{\dfrac{\dfrac{\dfrac{\dfrac{P[b] \vdash P[c], Q[b]}{P[b] \vdash \Box P[b], Q[b]}\,c}{\vdash P \to \Box P[b], Q[b]}}{\vdash P \to \Box P[b], \Box Q}\,b}{\vdash \Diamond(P \to \Box P), \Box Q}\,b}$$

Call the left skeleton δ. The skeleton to the right is the skeleton $\delta\sigma$ obtained by applying σ to δ. ◦

Example 6.3 (Order of inferences) The skeleton $\delta\sigma$ to the right in the previous example might look like an $\mathsf{LC}_L^{\mathsf{gt}}$-derivation, but since the lowermost rule application is an ν-inference introducing a label parameter fresh to the skeleton, $\delta\sigma$ does not comply with the neighbourhood condition mandatory for an $\mathsf{LC}_L^{\mathsf{gt}}$-derivation. Observe that it is possible to apply the inference in δ to which the label variable U belongs and the inference introducing b in the opposite order, i.e., first applying an inference to $\Box Q_\perp$ and then to $\Diamond(P \to \Box P)_\perp$, obtaining the skeletons δ' and $\delta'\sigma$ in the example below.

Let δ' be the skeleton δ in Example 6.2, except that the two lowermost inferences are interchanged. $\sigma = \{U \mapsto b\}$ is the same label substitution as in Example 6.2.

$$\underset{\delta'}{\dfrac{\dfrac{\dfrac{\dfrac{P[U] \vdash P[c], Q[b]}{P[U] \vdash \Box P[U], Q[b]}\,c}{\vdash P \to \Box P[U], Q[b]}}{\vdash \Diamond(P \to \Box P), Q[b]}\,U}{\vdash \Diamond(P \to \Box P), \Box Q}\,b} \qquad \underset{\delta'\sigma}{\dfrac{\dfrac{\dfrac{\dfrac{P[b] \vdash P[c], Q[b]}{P[b] \vdash \Box P[b], Q[b]}\,c}{\vdash P \to \Box P[b], Q[b]}}{\vdash \Diamond(P \to \Box P), \Box Q[b]}\,b}{\vdash \Diamond(P \to \Box P), \Box Q}\,b}$$

As opposed to the skeleton $\delta\sigma$ of Example 6.2, $\delta'\sigma$ *is* an $\mathsf{LC}^{\mathsf{gt}}$-derivation. The lesson learned is that order of inferences is crucial. This is exactly what we want to relax in $\mathsf{LC}^{\mathsf{fv}}$. Notice that the leaf sequents of δ in Example 6.2 and δ' are equal, indicating that the skeletons are invariant under order of rule application.

The skeleton $\delta\sigma'$ is not an $\mathsf{LC}^{\mathsf{gt}}$-proof as it misses an axiom. A more promising label substitution in creating an $\mathsf{LC}^{\mathsf{gt}}_{L}$-proof is $\sigma' = \{U \mapsto c\}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P[c] \;\vdash\; P[c], Q[b]}{P[c] \;\vdash\; \Box P[c], Q[b]}\; c, \dagger}{\vdash\; P \to \Box P[c], Q[b]}}{\vdash\; \Diamond(P \to \Box P), Q[b]}\; c, \ddagger}{\vdash\; \Diamond(P \to \Box P), \Box Q}\; b}{\delta'\sigma'}$$

Now the leaf sequent contains an axiom, but the skeleton is still not an $\mathsf{LC}^{\mathsf{gt}}_{L}$-proof. This time it is the introduction of c in the second lowermost inference which breaks the neighbourhood condition. Notice that we cannot create an admissible $\mathsf{LC}^{\mathsf{gt}}$-derivation by changing the order of inferences, as we did in when we created $\delta'\sigma$ from $\delta\sigma$, displayed by the two skeletons earlier in this example. To create an admissible $\mathsf{LC}^{\mathsf{gt}}$-derivation from $\delta'\sigma'$ we would have to change the order of inferences such that the label c belonging to the inference marked † is introduced before the inference marked ‡. This is impossible as the principal formula of † is a subformula of the principal formula of ‡. ◦

One of the main objectives in the construction of the free-variable calculus $\mathsf{LC}^{\mathsf{fv}}$ is to make derivations invariant under rule application order. As we see from the skeletons in Examples 6.2 and 6.3 there are dependencies between the inferences. To be able to express such dependencies we define the notion of a *reduction ordering* on inferences in a skeleton.

Definition 6.4 (Reduction ordering)

Let r_1 and r_2 be two inferences on the same branch in a skeleton and σ be a label substitution. The reduction ordering induced by σ is denoted $\rhd_\sigma$ and is defined as the transitive closure of $\ll \cup \prec_\sigma$ where

- $\ll$ is the weakest binary relation such that $r_1 \ll r_2$ if the principal formula of r_2 is an active formula of r_1, and
- $\prec_\sigma$ is the weakest binary relation such that $r_1 \prec_\sigma r_2$ if s belongs to r_1, U belongs to r_2 and $U\sigma = s$. ∎

Let $\lll$ denote the transitive closure of $\ll$, and call the principal formula of r_2 a *descendant* of the principal formula of r_1, if $r_1 \lll r_2$. Notice that the set of descendants of X is not necessarily a set of proper subformulae of X, as the explicit copy of a ν-formula is a descendant of its origin, but not one of its proper subformulae.

Let δ be a skeleton and σ a label substitution. The $\rhd_\sigma$-relation encodes the order of inferences applied to $\delta\sigma$ necessary to achieve an $\mathsf{LC}^{\mathsf{gt}}$-derivation. We say that the pair $\langle \delta, \sigma \rangle$ *conforms to* $\rhd_\sigma$ if $r_1 \rhd_\sigma r_2$ implies that r_1 is below r_2 in the skeleton δ. The $\ll$-relation reflects the fact that an inference cannot be applied directly to a proper subformula of a formula in a sequent, e.g., we cannot apply an inference to $\neg Q$ in $P \wedge \neg Q$ before a rule is applied to $P \wedge \neg Q$. The $\prec_\sigma$-relation relates to the eigenparameter condition in LC^{g}: if a label variable is mapped to a label parameter, the label parameter must be introduced before the label variable to ensure that π-inferences do not introduce label parameters which are not fresh to the denominator.

Example 6.5 (Reduction ordering) The skeletons δ and δ' from Example 6.2 and Example 6.3 are redisplayed below. The inferences are in both skeletons named r_x, where x is either the label it introduces or the main connective of its principal formula.

$$\dfrac{\dfrac{\dfrac{\dfrac{P[U] \vdash P[c], Q[b]}{P[U] \vdash \Box P[U], Q[b]}\,r_c}{\vdash P \to \Box P[U], Q[b]}\,r_\to}{\vdash P \to \Box P[U], \Box Q}\,r_b}{\vdash \Diamond(P \to \Box P), \Box Q}\,r_U$$
$$\delta$$

$$\dfrac{\dfrac{\dfrac{\dfrac{P[U] \vdash P[c], Q[b]}{P[U] \vdash \Box P[U], Q[b]}\,r_c}{\vdash P \to \Box P[U], Q[b]}\,r_\to}{\vdash \Diamond(P \to \Box P), Q[b]}\,r_U}{\vdash \Diamond(P \to \Box P), \Box Q}\,r_b$$
$$\delta'$$

The $\ll$-relation on the inferences in both δ and δ' is $\ll = \{\langle r_U, r_\to \rangle, \langle r_\to, r_c \rangle\}$. That the same relation holds for both skeletons is of course no coincidence as the $\ll$-relation really is the immediate subformula relation recasted into dependencies on an order of inferences. What the $\ll$-relation for the two skeletons encodes in these particular cases is that r_U must be applied prior to $r_\to$, and r_c must be applied after $r_\to$.

Let $\sigma = \{U \mapsto b\}$, causing $\prec_\sigma = \{\langle r_b, r_U \rangle\}$ and $\rhd_\sigma$ to be the transitive closure of $\{\langle r_b, r_U \rangle, \langle r_U, r_\to \rangle, \langle r_\to, r_c \rangle\}$. We see that $\langle \delta', \sigma \rangle$ does conform to $\rhd_\sigma$, while $\langle \delta, \sigma \rangle$ does not. This complies with the results from the examples 6.2 and 6.3, where we saw that $\delta\sigma$ was not an $\mathsf{LC}^{\mathsf{gt}}$-derivation, while $\delta'\sigma$ was. $\circ$

Having established the reduction ordering induced by a label substitution on inferences in a skeleton we are ready to define an $\mathsf{LC}_L^{\mathsf{fv}}$-derivation.

Remark. Notice that we in the following bend the definition of an LC_L-derivation (Definition 3.22) a bit, requiring that an $\mathsf{LC}_L^{\mathsf{fv}}$-derivation is not just a skeleton abiding by some conditions, but a pair comprised of a skeleton and a label substitution where both need to conform to a set of conditions. $\circ$

Definition 6.6 ($\mathsf{LC}^{\mathsf{fv}}_L$-derivation)
Let δ be a skeleton and σ be a label substitution. The pair $\langle \delta, \sigma \rangle$ is an $\mathsf{LC}^{\mathsf{fv}}_L$-derivation if the root sequent of δ is empty labelled and the following conditions are met:

- every inference in δ is an instance of a rule in the $\mathsf{LC}^{\mathsf{fv}}$-calculus,
- σ is an $\mathsf{LC}^{\mathsf{fv}}_L$-admissible label substitution. The admissible substitutions in K45 and S5 differ:
  - σ is an $\mathsf{LC}^{\mathsf{fv}}_{K45}$-admissible label substitution, if σ satisfies the conditions:
    * the *groundness condition:* $U\sigma \in \mathsf{Par}$ for every label variable U in the domain of σ,
    * the *neighbourhood condition*: if $\sigma(U) = c$, then there is a branch in the skeleton containing an occurrence of c and an occurrence of U, and
    * the *non-empty W^+ condition:* if the reduction ordering $\rhd_\sigma$ induced by σ is reflexive, there is an inference r_c belonging to the label parameter c for which there is no inference r_U belonging to a label variable U such that $r_U \lll r_c$.
  - No restrictions apply to a label substitution in an admissible $\mathsf{LC}^{\mathsf{fv}}_{S5}$-derivation. ∎

Example 6.7 ($\mathsf{LC}^{\mathsf{fv}}$-derivation) Let δ, δ', σ and σ' be as in the previous examples (Example 6.2, 6.3 and 6.5). Then $\langle \delta, \sigma \rangle$, $\langle \delta, \sigma' \rangle$, $\langle \delta', \sigma \rangle$ and $\langle \delta', \sigma' \rangle$ are all $\mathsf{LC}^{\mathsf{fv}}$-derivations.

Recall, as stated in the Examples 6.2 and 6.3, that of the skeletons $\delta'\sigma$, $\delta'\sigma'$ and $\delta\sigma$ only the last is an $\mathsf{LC}^{\mathsf{gt}}_L$-derivation. This should serve an early indication of the increased flexibility of $\mathsf{LC}^{\mathsf{fv}}$ compared to $\mathsf{LC}^{\mathsf{gt}}$. ◦

Given that what label an inference introduce is entirely decided by the index of its principal formula, a skeleton abiding by the rules of $\mathsf{LC}^{\mathsf{fv}}$ has leaf sequents invariant under order of rule application.

Example 6.8 (Invariance under rule application) The skeletons δ'_2–δ'_5 below are the same as the skeletons δ_2–δ_5 found on the pages 34 and 36, except the skeletons below are adapted to the rules of $\mathsf{LC}^{\mathsf{fv}}$. Formula numbers are only displayed in the root sequent and copy histories are neglected altogether.

$$\dfrac{\dfrac{P \vdash R[c_4] \qquad Q \vdash R[c_4]}{P \lor Q \vdash R[c_4]}\; c_4}{\underset{1}{P} \underset{2}{\lor} \underset{3}{Q} \vdash \underset{4}{\Diamond} \underset{5}{R}} \qquad\qquad \dfrac{\dfrac{P \vdash R[c_4]}{P \vdash \Diamond R}\; c_4 \qquad \dfrac{Q \vdash R[c_4]}{Q \vdash \Diamond R}\; c_4}{\underset{1}{P} \underset{2}{\lor} \underset{3}{Q} \vdash \underset{4}{\Diamond} \underset{5}{R}}$$

$$\delta'_2 \qquad\qquad\qquad\qquad \delta'_3$$

$$\dfrac{\dfrac{\Box P, P[U_1] \vdash Q[c_3]}{\Box P \vdash Q[c_3]}\; \nu, U_1}{\underset{1}{\Box} \underset{2}{P} \vdash \underset{3}{\Box} \underset{4}{Q}}\; \pi, c_3 \qquad\qquad \dfrac{\dfrac{\Box P, P[U_1] \vdash Q[c_3]}{\Box P, P[U_1] \vdash \Box Q}\; \pi, c_3}{\underset{1}{\Box} \underset{2}{P} \vdash \underset{3}{\Box} \underset{4}{Q}}\; \nu, U_1$$

$$\delta'_4 \qquad\qquad\qquad\qquad \delta'_5$$

Observe that the leaf sequents of the skeletons δ'_2 and δ'_3 are equal. The important feature to notice is that the π-inferences in each branch introduce the same label parameter, since their principal formulae have equal formula numbers. The leaf sequents of δ'_4 and δ'_5 are also equal. Here it the fact that ν-formulae introduce label variables that provides the remedy. The strict control of what label to introduce, enforced by the rules of $\mathsf{LC}^{\mathsf{fv}}$ through the aid of indices, gives us the desired result of invariance of rule application order. ◦

How $\mathsf{LC}^{\mathsf{fv}}$- and $\mathsf{LC}^{\mathsf{gt}}$-derivations are related are explained in detail in the following pages. The discussion is divided into the points

- *grounding*,
- the contextually equivalent condition,
- *permutation*,
- *cycles*, and
- the non-empty W^+ condition.

Grounding, permutation and cycles are concepts yet to be defined. The two latter concepts are especially crucial instruments in explaining the increased efficiency of $\mathsf{LC}^{\mathsf{fv}}$.

Grounding. One difference between $\mathsf{LC}^{\mathsf{gt}}$ and $\mathsf{LC}^{\mathsf{fv}}$ is that an $\mathsf{LC}^{\mathsf{fv}}_L$-derivation may contain label variables, which is not the case for an $\mathsf{LC}^{\mathsf{gt}}_L$-derivation. An $\mathsf{LC}^{\mathsf{fv}}_{S5}$-derivation does not require its label substitution to be grounding for its skeleton, if $\langle \delta, \sigma \rangle$ is an $\mathsf{LC}^{\mathsf{fv}}_{S5}$-derivation, then the skeleton $\delta\sigma$ may contain label variables. The next lemma inspired by [Antonsen 2003, Lemma 2.11] provides the remedy.

Lemma 6.9 (Grounding) Let $\langle \delta, \sigma \rangle$ be an $\mathsf{LC}^{\mathsf{fv}}_L$-derivation, then there is an $\mathsf{LC}^{\mathsf{fv}}_L$-derivation $\langle \delta, \sigma' \rangle$ where σ' grounds δ. ∎

Proof. Let $\langle \delta, \sigma \rangle$ be an $\mathsf{LC}^{\mathsf{fv}}_{L}$-derivation. If $\langle \delta, \sigma \rangle$ is an $\mathsf{LC}^{\mathsf{fv}}_{K45}$-derivation, then σ is grounding for δ by assumption, see Definition 6.6, so assume $\langle \delta, \sigma \rangle$ is an $\mathsf{LC}^{\mathsf{fv}}_{S5}$-derivation and that σ is not ground for δ. Let ψ denote the label variables which occur in δ but not in $\mathrm{DOM}(\sigma)$, i.e., every label variable which is not mapped to a ground label. Let σ_ψ be such that $\psi\sigma_\psi = \{\epsilon\}$, and let $\sigma' = \sigma \circ \sigma_\psi$. Since no restrictions apply to an $\mathsf{LC}^{\mathsf{fv}}_{S5}$-admissible label substitution, $\langle \delta, \sigma' \rangle$ is indeed an $\mathsf{LC}^{\mathsf{fv}}_{S5}$-derivation, and σ' grounds δ. □

Contextually equivalent condition. The contextually equivalent condition present in $\mathsf{LC}^{\mathsf{gt}}$ is missing in $\mathsf{LC}^{\mathsf{fv}}$. This condition is superfluous in $\mathsf{LC}^{\mathsf{fv}}$ as the condition is enforced by the rules of the calculus.

Lemma 6.10 Formulae with identical indices in an $\mathsf{LC}^{\mathsf{fv}}$-derivation are contextually equivalent. ■

Proof. Let X and Y be two different indexed formulae in an $\mathsf{LC}^{\mathsf{fv}}$-derivation δ having equal indices. As every formula in the root sequent of δ is given a distinct formula number and no calculus rule manipulates formula numbers, X and Y must occur in different sequents of δ. There are two ways in which this may happen.

The base case is that X and Y are extra formulae in a β-inference, i.e., X and Y denote the same formula in the denominator of an inference, while in the numerator X occurs in one of the sequents and Y in the other. By the definition of contextually equivalent (Definition 4.3 on page 30), X and Y are contextually equivalent.

Now let r and r' be two inferences of the same type applied to two sequents where the principal formulae in the inferences have equal indices. Let X be an active formula in r and Y be an active formula in r'. This is the second way two different formulae may have the same index. By assumption the principal formulae of r and r' are contextually equivalent as they are indexed identically. If X and Y introduce any labels, then by the rules of $\mathsf{LC}^{\mathsf{fv}}$ and given that X and Y have equal indices, they belong to the same label. Conclude by the rules of $\mathsf{LC}^{\mathsf{fv}}$ and Definition 4.3, that X and Y are syntactically equal and contextually equivalent. □

Let r and r' be two contextually equivalent inferences in the same skeleton. Using Lemma 6.10 and the fact that r and r' are applied to formulae having the same index (since they are contextually equivalent), the two inferences must by the rules of $\mathsf{LC}^{\mathsf{fv}}$ introduce the same label. This complies with the contextually equivalent condition and explains why we do not need this condition in $\mathsf{LC}^{\mathsf{fv}}$.

Permutation. The neighbourhood condition in $\mathsf{LC}^{\mathsf{gt}}$ concerns the introduction of labels by ν-inferences. From Example 6.7 we learned that $\langle \delta, \sigma \rangle$,

$\langle \delta, \sigma' \rangle$ and $\langle \delta', \sigma \rangle$ are $\mathsf{LC}^{\mathsf{fv}}$-derivations, but of the skeletons $\delta\sigma$, $\delta'\sigma'$ and $\delta'\sigma$, only the latter one is an $\mathsf{LC}^{\mathsf{gt}}$-derivation. The only difference between the skeletons δ and δ' is the applied order of inferences.

Permutation is used to relate skeletons which differ only in the order of inferences [Waaler 2001]. If this is the case, we call the skeletons *permutation variants* of each other. We need to be able to permute the skeleton δ of an $\mathsf{LC}^{\mathsf{fv}}_L$-derivation $\langle \delta, \sigma \rangle$ such that the skeleton $\delta'\sigma$, where δ' is a permutation variant of δ, complies with the conditions in $\mathsf{LC}^{\mathsf{gt}}$ and becomes an $\mathsf{LC}^{\mathsf{gt}}_L$-derivation. Figure 6.2 on the following page displays the *permutation schemata* for permuting two sets of contextually equivalent inferences in a skeleton complying to the rules of $\mathsf{LC}^{\mathsf{fv}}$. All the skeletons in the figure have a root containing two formulae of special interest which we in the following call X and Y. The figures 6.2(a), 6.2(b) and 6.2(c) all contain two skeletons where the leftmost skeletons in 6.2(a) and 6.2(b) and upper skeleton in 6.2(c) are obtained by first applying an inference to X succeeded by applying inferences to Y in every sequent resulting from the rule application to X. The rightmost skeletons in 6.2(a) and 6.2(b), and lower skeleton in 6.2(c) are the results of applying inferences by starting with an application to Y and continuing with X in every resulting branch of the inference on Y. It is easy to see that if the skeletons follow the rules of $\mathsf{LC}^{\mathsf{fv}}$ such that contextually equivalent inferences belong to the same label, then skeletons in the same subfigure agree on leaf sequents.

Now we generalize the result of permutation to arbitrary skeletons. To do so we use the notion of a balanced skeleton. This notion simplifies the exposition as it allows us to assume that we can use the permutation schemata in Figure 6.2 to permute inferences which are branching: Consider the skeleton on the right in Figure 6.2(b). If the inference r_ω was not applied in the left branch and the left leaf sequent Γ, Ω, β_1 was missing, then permuting r_ω downwards would not be possible by our formulation. If we require that the skeleton is balanced, we know that the inference r_ω is also present in the left branch and permutation follows easily by the schema.

Lemma 6.11 (Permutation) Let F be a non-atomic formula occurring in the root of a balanced skeleton δ conforming to the calculus rules of $\mathsf{LC}^{\mathsf{fv}}$ and let r be an inference in δ with principal formula F Then there is a permutation variant δ' of δ where F is principal formula in the root sequent of δ. ■

Proof sketch. (See proof of Lemma 2.14 in [Waaler 2001, p. 1509].) Assume δ and r is as in the Lemma and let R denote the set of all inferences in δ contextually equivalent to r. Note that $r \in R$. All $r' \in R$ occur in different branches of δ. Repeatedly choose the uppermost inference $r' \in R$ and permute downwards using the permutation schemata in Figure 6.2. This produces the desired skeleton δ'. □

$$\dfrac{\dfrac{\Gamma, \Omega^1, \Omega^2}{\Gamma, \Omega^1, \omega^2}\, r_{\omega^2}}{\Gamma, \omega^1, \omega^2}\, r_{\omega^1} \qquad\qquad \dfrac{\dfrac{\Gamma, \Omega^1, \Omega^2}{\Gamma, \omega^1, \Omega^2}\, r_{\omega^1}}{\Gamma, \omega^1, \omega^2}\, r_{\omega^2}$$

(a) Permutation schema for two non-branching inferences, r_{ω^1} and r_{ω^2}.

$$\dfrac{\dfrac{\Gamma, \Omega, \beta_1 \quad \Gamma, \Omega, \beta_2}{\Gamma, \Omega, \beta}\, r_\beta}{\Gamma, \omega, \beta}\, r_\omega \qquad\qquad \dfrac{\dfrac{\Gamma, \Omega, \beta_1}{\Gamma, \omega, \beta_1}\, r_\omega \quad \dfrac{\Gamma, \Omega, \beta_2}{\Gamma, \omega, \beta_2}\, r_\omega}{\Gamma, \omega, \beta}\, r_\beta$$

(b) Permutation schema for one branching, r_β, and one non-branching inference, r_ω.

$$\dfrac{\dfrac{\Gamma, \beta^1_1, \beta^2_1 \quad \Gamma, \beta^1_1, \beta^2_2}{\Gamma, \beta^1_1, \beta^2}\, r_{\beta^2} \quad \dfrac{\Gamma, \beta^1_2, \beta^2_1 \quad \Gamma, \beta^1_2, \beta^2_2}{\Gamma, \beta^1_2, \beta^2}\, r_{\beta^2}}{\Gamma, \beta^1, \beta^2}\, r_{\beta^1}$$

$$\dfrac{\dfrac{\Gamma, \beta^1_1, \beta^2_1 \quad \Gamma, \beta^1_2, \beta^2_1}{\Gamma, \beta^1, \beta^2_1}\, r_{\beta^1} \quad \dfrac{\Gamma, \beta^1_1, \beta^2_2 \quad \Gamma, \beta^1_2, \beta^2_2}{\Gamma, \beta^1, \beta^2_2}\, r_{\beta^1}}{\Gamma, \beta^1, \beta^2}\, r_{\beta^2}$$

(c) Permutation schema for two branching inferences, r_{β^1} and r_{β^2}.

Figure 6.2: Permutation schemata. In the following let $i \in \{1, 2, \_\}$. Γ is a set of indexed formulae. ω^i is an arbitrary indexed formula of type α, π or ν. Ω^i is the set of components of ω^i. β^i_1 and β^i_2 are the components of β^i. The possible labels introduced by the inferences are not indicated, instead we identify an inference by letting the principal formula of an inference be subscripted in the name of the inference, and position the inference name by the horizontal line to which the inference corresponds.

The following corollary and proof is gathered from [Waaler 2001].

Corollary 6.12 (Permutation) Let $\langle \delta, \sigma \rangle$ be an $\mathsf{LC}^{\mathsf{fv}}_L$-derivation such that the reduction ordering $\rhd_\sigma$ induced by σ is irreflexive. Then there is permutation variant δ' of δ such that $\langle \delta', \sigma \rangle$ conforms to $\rhd_\sigma$. ∎

Proof sketch. Induction on the sub-skeletons of δ, using Lemma 6.11. □

Permutation is the explanation of why both $\langle \delta, \sigma \rangle$ and $\langle \delta', \sigma \rangle$ may be $\mathsf{LC}^{\mathsf{fv}}_L$-derivations, while of $\delta\sigma$ and $\delta'\sigma$, only $\delta'\sigma$ is an $\mathsf{LC}^{\mathsf{gt}}$-derivation (See Example 6.7). The reason is that δ and δ' are permutation variants, so δ' is obtainable by application of Lemma 6.11 to δ. This is an example of the relaxed restrictions in $\mathsf{LC}^{\mathsf{fv}}$ compared to $\mathsf{LC}^{\mathsf{gt}}$, we do not require the inferences of a skeleton to be applied in the correct order according to an $\mathsf{LC}^{\mathsf{gt}}$-derivation because we know that the correct order is obtainable by interchanging inferences. An irreflexive reduction ordering ensures that the inferences in an $\mathsf{LC}^{\mathsf{fv}}$-derivation are interchangeable. However, the reduction ordering may be reflexive. This is characterized by a *cycle* in the derivation.

Cycles. Permutation is an important result in establishing the link between $\mathsf{LC}^{\mathsf{fv}}$ and $\mathsf{LC}^{\mathsf{gt}}$. The other important result needed is the ability to eliminate cycles. Cycles arise when a label variable U is mapped to a label introduced by an inference on a subformula of the formula to which the inference introducing U was applied.

Definition 6.13 (Cycle)
Let $\langle \delta, \sigma \rangle$ be an $\mathsf{LC}^{\mathsf{fv}}_L$-derivation. The derivation contains a cycle if there is an inference r_U belonging to a label variable U and an inference r_s belonging to s, such that $r_U \rhd_\sigma r_1 \rhd_\sigma \ldots \rhd_\sigma r_n \rhd_\sigma r_s$, $r_s \prec_\sigma r_U$ and $r'_s \lll r_s$ for no inferences $r'_s \neq r_s$ in δ belonging to s, for a finite set of inferences $r_U, r_s, r_1, \ldots, r_n$ in δ. We identify this cycle by the pair (U, s). ∎

Remark. The difference from our definition compared to the ones found in [Antonsen 2003, Antonsen and Waaler 2006] is if there is a cycle (U, s) in δ then the inference to which U belongs only is always related to the *lowermost* inference in δ introducing s. This extra condition is necessary given the fact that different π-formulae may introduce the same label resulting in a branch containing multiple occurrences on the same label. ∘

We call it a cycle because the reduction ordering on a $\mathsf{LC}^{\mathsf{fv}}_L$-derivation $\langle \delta, \sigma \rangle$ containing a cycle is reflexive. Using the intuition that the reduction ordering encodes the order of inferences needed to obtain an $\mathsf{LC}^{\mathsf{gt}}_L$-derivation in the skeleton $\delta\sigma$, reflexivity in the order of rule application makes no sense since it implies that there is an inference in the skeleton which must be applied before itself. This indicates that we must be able to eliminate cycles in order to obtain the desired transformation of $\mathsf{LC}^{\mathsf{fv}}$- to $\mathsf{LC}^{\mathsf{gt}}$-derivations.

Example 6.14 (Cycle) The skeletons δ' and $\delta'\sigma'$ from Example 6.3 are redisplayed below. σ' is $\{U \mapsto c\}$.

$$
\underset{\delta'}{\dfrac{\dfrac{\dfrac{\dfrac{P[U] \vdash P[c], Q[b]}{P[U] \vdash \Box P[U], Q[b]}\, c}{\vdash P \to \Box P[U], Q[b]}}{\vdash \Diamond(P \to \Box P), Q[b]}\, U}{\vdash \Diamond(P \to \Box P), \Box Q}\, b}
\qquad\qquad
\underset{\delta'\sigma'}{\dfrac{\dfrac{\dfrac{\dfrac{P[c] \vdash P[c], Q[b]}{P[c] \vdash \Box P[c], Q[b]}\, c}{\vdash P \to \Box P[c], Q[b]}}{\vdash \Diamond(P \to \Box P), \Box Q[b]}\, c}{\vdash \Diamond(P \to \Box P), \Box Q}\, b}
$$

The $\mathsf{LC}^{\mathsf{fv}}_L$-derivation $\langle \delta', \sigma' \rangle$ contains the cycle (U, c), given that U is mapped to c by σ, and the inference introducing c is applied to a proper subformula of the principal formula of the inference to which U belongs. In the skeleton $\delta\sigma'$ this amounts to a failure in meeting the neighbourhood condition as explained in Example 6.3. This failure reflects that a ν-formula has broken "unfreshness"-requirement by introducing a fresh label parameter, allowing of the possibility that a π-formula cannot abide by a freshness requirement, since it has no choice of what label parameter to introduce. This is the inherent problem of a cycle: a label variable U is mapped to a label which is by formula structure forced to be introduced above U in the skeleton, hence possibly violating the neighbourhood condition in $\mathsf{LC}^{\mathsf{gt}}$. $\circ$

The problem with cycles is that permutation cannot help us. If an $\mathsf{LC}^{\mathsf{fv}}_L$-derivation $\langle \delta, \sigma \rangle$ contains a cycle (U, s) it is impossible to permute the inference introducing s below the inference introducing U to obtain a derivation $\langle \delta', \sigma \rangle$ where $\delta'\sigma$ is an $\mathsf{LC}^{\mathsf{gt}}_L$-derivation. It is impossible since this would mean obtaining a skeleton δ' where a rule is applied directly to a subformula in a sequent. Notice that the definition of $\mathsf{LC}^{\mathsf{fv}}_L$-derivation does not require the reduction ordering to be irreflexive, and in fact there are ways to eliminate cycles, but not by permutation alone.

Example 6.15 (Cycle elimination) The skeleton δ_2 below is the same as δ' in the previous example extended by three extra inferences. The three new inferences are applied to the explicit copy χ of the ν-formula $\Diamond(P \to \Box P)_\perp$, the principal formula of the inference to which U belongs.

$$
\begin{array}{cl}
P[U], P[V] \vdash P[c], P[c], \chi^2, Q[b] & \\ \cline{1-1}
P[U], P[V] \vdash P[c], \Box P[V], \chi^2, Q[b] & c, r'_c \\ \cline{1-1}
P[U] \vdash P[c], P \to \Box P[V], \chi^2, Q[b] & r'_{\to} \\ \cline{1-1}
P[U] \vdash P[c], \chi^1, Q[b] & V, r_V \\ \cline{1-1}
P[U] \vdash \Box P[U], \chi^1, Q[b] & c, r_c \\ \cline{1-1}
\vdash P \to \Box P[U], \chi^1, Q[b] & r_{\to} \\ \cline{1-1}
\vdash \Diamond(P \to \Box P), Q[b] & U, r_U \\ \cline{1-1}
\vdash \Diamond(P \to \Box P), \Box Q & b, r_b
\end{array}
$$

Let $\sigma_2 = \{U \mapsto b, V \mapsto c\}$. The $\mathsf{LC}^{\mathsf{fv}}_L$-derivation $\langle \delta_2, \sigma_2 \rangle$ conforms to $\rhd_{\sigma_2}$. Notice that (V, c) is not a cycle, even though $r_V \lll r'_c$ and $r'_c \prec_\sigma r_V$. This is because there is an inference r_c to which the label c also belongs and $r_c \lll r'_c$. Observe also that the cycle was eliminated by "replicating" the inferences r_U, $r_{\to}$ and r_c to the new inferences r_V, $r'_{\to}$ and r'_c, which are applied to an explicit copy of the principal formula of r_U and its descendants.

We display the skeleton $\delta_2\sigma_2$ below.

$$
\begin{array}{cl}
P[b], P[c] \vdash P[c], P[c], \chi^2, Q[b] & \\ \cline{1-1}
P[b], P[c] \vdash P[c], \Box P[c], \chi^2, Q[b] & c \\ \cline{1-1}
P[b] \vdash P[c], P \to \Box P[c], \chi^2, Q[b] & \\ \cline{1-1}
P[b] \vdash P[c], \chi^1, Q[b] & c \\ \cline{1-1}
P[b] \vdash \Box P[b], \chi^1, Q[b] & c \\ \cline{1-1}
\vdash P \to \Box P[b], \chi^1, Q[b] & \\ \cline{1-1}
\vdash \Diamond(P \to \Box P), \Box Q[b] & b \\ \cline{1-1}
\vdash \Diamond(P \to \Box P), \Box Q & b
\end{array}
$$

Observe that the skeleton conforms to the conditions set by an $\mathsf{LC}^{\mathsf{gt}}_L$-derivation. No label parameter introduced by a ν-inference is fresh to the skeleton. The skeleton is in fact a simplified version of the $\mathsf{LC}^{\mathsf{gt}}_L$-proof found in Example 5.16.

The example suggests that it is possible to eliminate a cycle (U, c) by explicitly copying the ν-formula introducing U and letting V, the label variable introduced by the explicit copy, take the place of U and breaking the cycle (U, c) by removing $\{U \mapsto c\}$ from the label substitution. Given that all of the π-descendants of an explicit copy of a formula X introduce labels already occurring in the skeleton, (V, c) does not identify any cycle. $\circ$

The conjecture and a proof sketch of cycle elimination is delayed until after we have established the notion of an $\mathsf{LC}^{\mathsf{fv}}_L$-proof. Se also [Antonsen 2003, ch. 3] for a thorough and illustrative exposition on cycles and cycle elimination in first-order logic, yet no proof of cycle elimination is found.

Non-empty W^+ condition. The last point in this discussion concerns the non-empty W^+ condition of $\mathsf{LC}^{\mathsf{fv}}$-derivations, which is only applicable in K45. In the definition of an $\mathsf{LC}^{\mathsf{gt}}_{K45}$-derivation $\langle \delta, \sigma \rangle$, the non-empty W^+ condition requires that *if the reduction ordering $\rhd_\sigma$ induced by σ is reflexive, there is an inference r_c belonging to the label parameter c for which there is no inference r_U belonging to a label variable U such that $r_U \lll r_c$.* Intuitively this means that if there are cycles in an $\mathsf{LC}^{\mathsf{fv}}_{K45}$-derivation, then there must exist a π-inference which is capable of being permuted downwards as the first inference introducing a label in the branch containing the cycle. This condition relates to the non-empty W^+ condition which $\mathsf{LC}^{\mathsf{gt}}_{K45}$-derivations have to conform to; the condition in $\mathsf{LC}^{\mathsf{gt}}_{K45}$ states that ν-inferences are applicable only if there exists a label parameter in the denominator. This indicates that no ν-inference may be applied to a branch prior to the application of some π-inference in the same branch. The π-inference ensures that a putative satisfying model has a non-empty W^+-set, making it "safe" to apply ν-rules. This in turn explains the role of the condition in $\mathsf{LC}^{\mathsf{fv}}$. In order to eliminate cycles, ν-formulae must be copied to introduce new label variables, which are used to break the cycle created partly by a label substitution. But to be able to copy ν-formulae we must have a guarantee for a non-empty W^+. A π-inference which relies on to be applicable in a branch provides this guarantee, this is how the effects of the non-empty W^+ condition in $\mathsf{LC}^{\mathsf{fv}}_{K45}$ should be understood.

Example 6.16 (Non-empty W^+ condition, K45) The skeleton below is the same as δ in Example 6.2 except that the formulae $\Box Q_\perp$ and $Q[b]_\perp$ are removed. χ denotes the explicit copy of $\Diamond(P \to \Box P)_\perp$.

$$\cfrac{\cfrac{\cfrac{P[U] \vdash P[c], \chi}{P[U] \vdash \Box P[U], \chi}\,c}{\vdash P \to \Box P[U], \chi}}{\vdash \Diamond(P \to \Box P)}\,U$$

The skeleton together with the label substitution $\sigma = \{U \mapsto c\}$ creates the cycle (U, c). Let r_c be the inference introducing c and r_U be the inference introducing the label variable U. The inference r_c is the only inference in the skeleton introducing a label parameter. As the principal formula of r_c is a descendant of the principal formula of r_U, the inferences r_c and r_U are related such that $r_U \lll r_U$ holds. Since there is no inference introducing a label parameter and where its principal formula is not a descendant of a ν-formula, the skeleton violates the non-empty W^+ condition. Hence $\langle \delta, \sigma \rangle$ is not an $\mathsf{LC}^{\mathsf{fv}}_{K45}$-derivation.

Expanding the explicit copy of the ν-formula and its descendants will not help. This is displayed in the skeleton below.

$$
\dfrac{\dfrac{\dfrac{P[U], P[U_1]^1 \;\vdash\; P[c], P[c]^1, \chi^2}{P[U], P[U_1]^1 \;\vdash\; P[c], \Box P[U_1]^1, \chi^2}\, c}{\dfrac{\dfrac{P[U] \;\vdash\; P[c], P \to \Box P[U_1]^1, \chi^2}{P[U] \;\vdash\; P[c], \chi^1}\, U_1}{\dfrac{P[U] \;\vdash\; \Box P[U], \chi^1}{}\, c}}}{\dfrac{\vdash\; P \to \Box P[U], \chi^1}{\vdash\; \Diamond(P \to \Box P)}\, U}
$$

Further expanding explicitly copied formulae does not help in introducing new label parameters, in the attempt of finding a label parameter b such that $V \lll b$ for no label variable V. Since π-formulae with the same formula number introduce the same label parameter and formula numbers are always left unchanged by explicit copying. In the skeleton immediately above, we need to let σ either map the label variable U or U_1 to the label parameter c. Either ways there are no means possible to permute an inference belonging to a label parameter such that b such that $V \lll b$ for no label variable V.

The skeleton δ'' displayed below is the same as the first skeleton given in this example, except $\Box Q_\perp$ is added to the root sequent and is expanded in the uppermost inference of the skeleton. The skeleton is a permutation variant of both δ and δ found in Example 6.5.

$$
\dfrac{\dfrac{\dfrac{\dfrac{P[U] \;\vdash\; P[c], Q[b]}{P[U] \;\vdash\; P[c], \Box Q}\, b}{P[U] \;\vdash\; \Box P[U], \Box Q}\, c}{\vdash\; P \to \Box P[U], \Box Q}}{\vdash\; \Diamond(P \to \Box P), \Box Q}\, U
$$

Let $\sigma = \{U \mapsto c\}$ be a label substitution, then $\langle \delta, \sigma \rangle$ is an $\mathsf{LC}_l^{\mathsf{fv}}$-derivation. The reduction ordering contains the cycle (U, c), but the for the inference r_b belonging to the label parameter b there is no inference r' such that $r' \lll r_b$. This means that the derivation complies with the non-empty W^+-condition, even though a cycle exists. ∘

This concludes the discussion started on page 69. Next we construct the necessary parts for the definition of $\mathsf{LC}_L^{\mathsf{fv}}$-proofs.

Definition 6.17 (Unifier)

Let Ψ denote a set of labels and $\Psi\sigma$ be the set $\Psi\sigma = \{s\sigma \mid s \in \Psi\}$. A label substitution σ is a unifier for a non-empty set of labels Ψ if $\Psi\sigma$ is singleton, i.e., a set containing exactly one element. ∎

Example 6.18 (Unifier) If s and t are labels, σ is a label substitution and $s\sigma = t\sigma$, then σ is a unifier for $\{s, t\}$.

If U is a label variable, then any label substitution unifies $\{U\}$, even the label substitution $\sigma = \emptyset$. ∘

Definition 6.19 (σ-axiom)

A sequent of the form $\Gamma, A[s]_\top, A[t]_\bot$ is a *σ-axiom* if σ unifies $\{s, t\}$. The formulae $A[s]_\top, A[t]_\bot$ are then referred to as a *closing pair*. ▪

Lemma 6.20 A σ-axiom is not satisfiable. ▪

Proof. Assume that the σ-axiom $\Gamma, A[s]_\top, A[t]_\bot$ is satisfiable. Then there must exist an L-model $(\mathcal{M}, \phi)$ and a label interpretation ρ such that $(\mathcal{M}, \phi) \vDash_\rho (\Gamma, A[s]_\top, A[t]_\bot)\sigma$. This is impossible, since both $(\mathcal{M}, \phi) \vDash_\rho A[\sigma s]_\top$ and $(\mathcal{M}, \phi) \vDash_\rho A[\sigma t]_\bot$ cannot hold, given that σ is a unifier for $\{s, t\}$ □

Definition 6.21 ($\mathsf{LC}_L^{\mathsf{fv}}$-proof)

An *$\mathsf{LC}_L^{\mathsf{fv}}$-proof* is a finite $\mathsf{LC}^{\mathsf{fv}}$-derivation $\langle \delta, \sigma \rangle$ where every branch of δ contains a σ-axiom. ▪

Example 6.22 ($\mathsf{LC}_l^{\mathsf{fv}}$-proof) Of the derivations listed in Example 6.7 $\langle \delta, \sigma' \rangle$ and $\langle \delta', \sigma \rangle$ are both $\mathsf{LC}_{K45}^{\mathsf{fv}}$- and $\mathsf{LC}_{S5}^{\mathsf{fv}}$-proofs. ∘

If a branch in an $\mathsf{LC}^{\mathsf{fv}}$-derivation $\langle \delta, \sigma \rangle$ contains a σ-axiom, σ is said to *close* this branch. If σ closes every branch of $\langle \delta, \sigma \rangle$, we say that σ closes the skeleton δ.

In the following examples all skeletons at hand are denoted δ and every label substitution is denoted by σ. In all but the first example, formula numbers are only displayed in the root sequent and labels are simplified.

Example 6.23 ($\mathsf{LC}_L^{\mathsf{fv}}$-proof) The skeleton below displays a skeleton complying to the rules of $\mathsf{LC}^{\mathsf{fv}}$ with all formula numbers, copy histories, explicit copies and no abbreviated label introductions.

$$\dfrac{\dfrac{\dfrac{\underset{2}{P}[U_{(1,0)}]^{(0.0)} \vdash \underset{1}{\Diamond}(\underset{2}{P} \underset{3}{\rightarrow} \underset{4}{\Box}\underset{5}{P})^1, \underset{5}{P}[c_4]^{(0.0)}, \underset{7}{Q}[c_6]}{\underset{2}{P}[U_{(1,0)}]^{(0.0)} \vdash \underset{1}{\Diamond}(\underset{2}{P} \underset{3}{\rightarrow} \underset{4}{\Box}\underset{5}{P})^1, \underset{4}{\Box}\underset{5}{P}[U_{(1,0)}]^{(0.0)}, \underset{7}{Q}[c_6]}\, c_4}{\vdash \underset{1}{\Diamond}(\underset{2}{P} \underset{3}{\rightarrow} \underset{4}{\Box}\underset{5}{P})^1, \underset{2}{P} \underset{3}{\rightarrow} \underset{4}{\Box}\underset{5}{P}[U_{(1,0)}]^{(0.0)}, \underset{7}{Q}[c_6]}}{\dfrac{\vdash \underset{1}{\Diamond}(\underset{2}{P} \underset{3}{\rightarrow} \underset{4}{\Box}\underset{5}{P}), \underset{7}{Q}[c_6]}{\vdash \underset{1}{\Diamond}(\underset{2}{P} \underset{3}{\rightarrow} \underset{4}{\Box}\underset{5}{P}), \underset{6}{\Box}\underset{7}{Q}}\, c_6}\, U_{(1,0)}$$

The lowermost inference introduces the label c_6 since 6 is the formula number of the principal π-formula. The second lowermost inference introduces the label variable $U_{(1,0)}$ as 1 and 0 is respectively the formula number and copy history of the principal formula. An $\mathsf{LC}^{\mathsf{fv}}_L$-proof is obtained by the $\mathsf{LC}^{\mathsf{fv}}_L$-derivation $\langle \delta, \sigma \rangle$, where $\sigma = \{U_{(1,0)} \mapsto c_4\}$. Compare this proof with the proofs found in Example 4.19 and 5.16 and observe that the $\mathsf{LC}^{\mathsf{fv}}_L$-proof is shorter than both the $\mathsf{LC}^{\mathsf{g}}_L$- and $\mathsf{LC}^{\mathsf{gt}}_L$-proof. ◦

Example 6.24 ($\mathsf{LC}^{\mathsf{fv}}_{S5}$-proof) The $\mathsf{LC}^{\mathsf{gt}}_{S5}$-derivation displayed below is over the same root sequent as found in Example 5.15.

$$\dfrac{\dfrac{\dfrac{P[c_1] \vdash P[c_1]}{\vdash P \rightarrow P[c_1]}}{\vdash \Box(P \rightarrow P)}\, c_1 \qquad \dfrac{\dfrac{P[U_6]^{0.0} \vdash P[U_6]^{0.0}, \Diamond(P \rightarrow P)^1}{\vdash P \rightarrow P[U_6]^{0.0}, \Diamond(P \rightarrow P)^1}}{\vdash \Diamond(P \rightarrow P)^1}\, U_6}{\vdash \underset{1}{\Box}(\underset{2}{P} \underset{3}{\rightarrow} \underset{4}{P}) \underset{5}{\wedge} \underset{6}{\Diamond}(\underset{7}{P} \underset{8}{\rightarrow} \underset{9}{P})}$$

The $\mathsf{LC}^{\mathsf{fv}}$-derivation $\langle \delta, \sigma \rangle$, where $\sigma = \{U_6 \mapsto b\}$, $\sigma = \{U_6 \mapsto c_1\}$ or $\sigma = \emptyset$, is an $\mathsf{LC}^{\mathsf{fv}}_{S5}$-proof, as all these label substitutions unify $\{U_6\}$ making both leaf sequents σ-axioms. In K45 we fail to find a closing substitution: $\sigma = \emptyset$ fails in meeting the condition of $U_6\sigma = U_6$ being ground, and both $\sigma = \{U_6 \mapsto c_1\}$ and $\sigma = \{U_6 \mapsto b\}$ violate the neighbourhood condition since no branch contains both c_1 and U_6; or b and U_6.

A K45-countermodel for the skeleton is an L-model $(\mathcal{M}, \phi)$, where $\mathcal{M} = (W, W^+, V)$ and $W^+ = \emptyset$. This K45-model is a countermodel for the leaf sequent in the right branch of the skeleton. Removing indices and subscripts from the this sequent we obtain the sequent $\{P[U]_\top, P[U]_\bot, \Diamond(P \rightarrow P)_\bot\}$. The formula $P[U]_\top$ is satisfied in (M, ϕ), since $\phi_\rho(U) = \odot$ for any ϕ_ρ and any label variable U when W^+ is empty. If the label of a formula is interpreted to the symbol $\odot$, the formula is by Definition 3.9 satisfiable. By the same observation, the formula $P[U]_\bot$ is also satisfiable in (M, ϕ). Lastly $\Diamond(P \rightarrow P)_\bot$ is trivially satisfied, as all points in W^+ (which are none) satisfy the formula $(P \rightarrow P)_\bot$. ◦

6.1 Cycle elimination

Cycles are possible because of the use of label variables adding flexibility to the free-variable calculus compared to the ground calculus. The ability to eliminate cycles is the last obstacle to be able to transform an $\mathsf{LC}^{\mathsf{fv}}_L$-derivation into an $\mathsf{LC}^{\mathsf{gt}}_L$-derivation. This allows for a syntactical proof sketch of soundness, by relying on the already established soundness of the $\mathsf{LC}^{\mathsf{gt}}$-calculus. This transformation can also be used to research the efficiency of $\mathsf{LC}^{\mathsf{fv}}$ compared to $\mathsf{LC}^{\mathsf{gt}}$, and may provide valuable proof theoretical insights.

The succeeding conjecture indicates that cycles can be eliminated by replicating parts of the skeleton in the derivation containing a cycle.

Conjecture 6.25 (Cycle elimination) Let $\langle \delta, \sigma \rangle$ be an $\mathsf{LC}^{\mathsf{fv}}_L$-proof containing a cycle (V, s). Then there is an $\mathsf{LC}^{\mathsf{fv}}_L$-proof $\langle \delta_2, \sigma_2 \rangle$ containing less cycles than $\langle \delta, \sigma \rangle$. Especially, $\langle \delta_2, \sigma_2 \rangle$ does not contain the cycle (V, s). ▪

Remark. We provide a highly informal plan of how to establish a proof. Consult also [Antonsen 2003, ch. 3].

Let $\langle \delta, \sigma \rangle$ be an $\mathsf{LC}^{\mathsf{fv}}$-derivation where a branch θ contains the cycle (V, s). The inference r_1 introduce the label variable V. The principal formula of r_1 is called F^κ and is a ν-formula, since it introduces the label variable V.

- Assume the skeleton δ is balanced and permute the inference r_1 upwards in θ such that every inference above r_1 in θ is applied to a descendant of F^κ.
- Let all inferences above r_1 constitute a subskeleton δ_F of δ. In the leaf sequent of θ there is a explicit copy of F, possibly through many explicit copies. Call this formula F'.
- "Copy the subskeleton δ_F, creating a new skeleton $\delta_{F'}$, and put $\delta_{F'}$ on top of the branch θ." This done by applying "the same" inferences applied to F and its descendants in θ, to F' and the descendants of F'.
- Let every label variable introduced by a descendant of F' be mapped to what the "equivalent" descendant of F is mapped to by σ. This is done to ensure that every branch of $\delta_{F'}$ is closed, and that no new cycles are created.
- Remove the cycle (V, s) by removing $\{V \mapsto s\}$ from the label substitution.
- Since $\langle \delta, \sigma \rangle$ is an $\mathsf{LC}^{\mathsf{fv}}_L$-proof, all branches in δ_F is closed under σ. The new skeleton δ_2, obtained by extending θ with $\delta_{F'}$; and the new label

substitution extended to ensure that $\delta_{F'}$ is closed, is then an $\mathsf{LC}^{\mathsf{fv}}_L$-proof without the cycle (V, s).

Consult also Figure 6.3 on the next page. ◦

6.2 Soundness

We give a semantical proof of soundness of $\mathsf{LC}^{\mathsf{fv}}$. The proof is easily established by using the groundwork and foundations already laid in the preceding chapters.

Lemma 6.26 (Countermodel preservation) Let δ be a skeleton conforming to the rules of $\mathsf{LC}^{\mathsf{fv}}$, and assume there is an L-countermodel for δ. Then there is an L-countermodel of δ', the result of applying an $\mathsf{LC}^{\mathsf{fv}}$-inference r to δ. ▪

Proof. Assume $(\mathcal{M}, \phi)$ satisfies the branch θ of δ under the label variable assignment ρ and that r is applied to the leaf sequent Γ of θ. If r is applied to a different branch than θ, then the branch occurs untouched in δ' and $(\mathcal{M}, \phi)$ is an L-countermodel for δ'.

Given that $(\mathcal{M}, \phi)$ satisfies θ under the label variable assignment ρ, Γ is satisfied in $(\mathcal{M}, \phi)$ under ρ. Check for each type of inference r, that δ', the resulting skeleton applying a rule to a leaf sequent of δ, has an L-countermodel.

α, β, ν. Γ is satisfiable in $(\mathcal{M}, \phi)$ under the label variable assignment ρ. Apply Lemma 3.15 and conclude that Γ' is satisfiable in $(\mathcal{M}, \phi)$ under ρ.

π. This case follows by the π-case in the proof of Lemma 5.18 (Countermodel preservation [in $\mathsf{LC}^{\mathsf{gt}}$]). □

Theorem 6.27 (Soundness of $\mathsf{LC}^{\mathsf{fv}}_L$)
Let $\langle \delta, \sigma \rangle$ be an $\mathsf{LC}^{\mathsf{fv}}_L$-proof of an indexed sequent Γ. Then Γ is L-valid. ▪

Proof. Aiming for a contradiction, assume that there is an $\mathsf{LC}^{\mathsf{fv}}_L$-proof $\langle \delta, \sigma \rangle$ of Γ, but that Γ is not L-valid. Given that Γ is not L-valid, δ_0, the skeleton comprised of Γ as root sequent and only node, must have an L-countermodel. By repeated application of Lemma 6.26 on δ_0, every resulting skeleton from a rule application to δ_0 must have an L-countermodel. Thus δ must have an L-countermodel.

By the assumption that $\langle \delta, \sigma \rangle$ is an $\mathsf{LC}^{\mathsf{g}}_L$-proof of Γ, every branch in δ is closed by σ. This is a contradiction: By Corollary 6.28 there is an $\mathsf{LC}^{\mathsf{fv}}_L$-proof

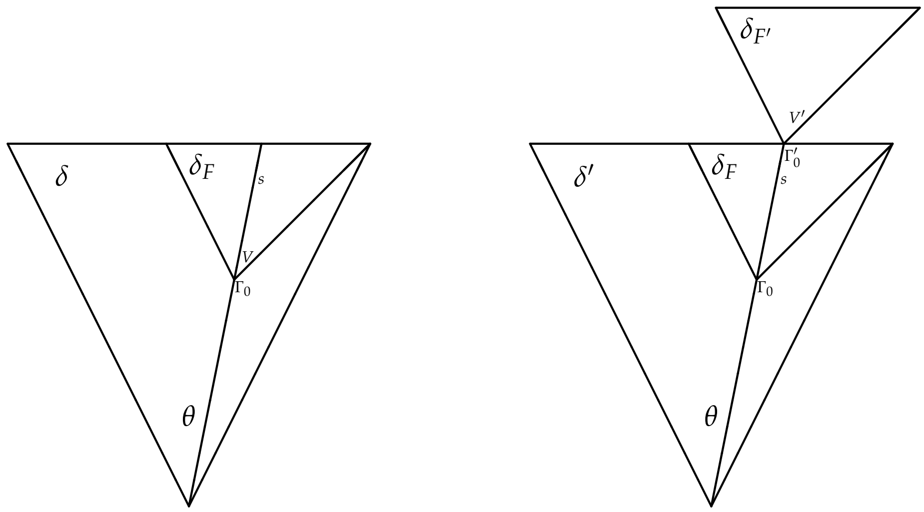


Figure 6.3: Cycle elimination. The skeletons illustrate the constructs used in the proof sketch of Conjecture 6.25. The left skeleton δ contains the cycle (V, s) indicated by V and s in the figure. The cycle occurs in δ_F, a subskeleton of δ, and on the branch θ. Γ_0 is the denominator of the inference introducing the label variable V to θ. Intuitively a cycle is broken by "copying" the subskeleton δ_F and "placing" the copy, $\delta_{F'}$, on top of Γ'_0, the leaf sequent of θ in δ, creating the skeleton δ' displayed as the right skeleton. By permuting inferences we have constructed the skeleton δ such that by applying nearly the exact same inferences to Γ'_0 as applied when creating the subskeleton δ_F, the desired copy subskeleton $\delta_{F'}$ is obtained. Since every branch in δ_F is closed, every branch in $\delta_{F'}$ is also closed by extending the label substitution closing δ to the label variables introduced in $\delta_{F'}$. Finally we remove V from the label substitution, thus breaking the cycle (V, s). Given that every branch in $\delta_{F'}$ is closed, every branch through θ must be closed.

$\langle \delta, \sigma' \rangle$ where σ' is grounding for δ. Let ρ be such that $\rho(U) = \phi(U\sigma')$ for all U in δ. By Lemma 6.20 no σ'-axiom is satisfiable, so cannot be $(\mathcal{M}, \phi)$ an L-countermodel as there is no branch θ in δ where $(\mathcal{M}, \phi)$ satisfies every sequent in θ under ρ. Γ is L-valid. □

We proceed by showing soundness of $\mathsf{LC}^{\mathsf{fv}}$ by indicating that we can convert every $\mathsf{LC}^{\mathsf{fv}}_L$-proof into an $\mathsf{LC}^{\mathsf{gt}}_L$-proof. Since $\mathsf{LC}^{\mathsf{gt}}$ is proven sound, soundness of $\mathsf{LC}^{\mathsf{fv}}$ follows under the assumption that Conjecture 6.25 holds.

We first need to show that the grounding of an $\mathsf{LC}^{\mathsf{fv}}_L$-proof does not destroy the proof.

Corollary 6.28 (Grounding) Let $\langle \delta, \sigma \rangle$ be an $\mathsf{LC}^{\mathsf{fv}}_L$-proof. Then there is an $\mathsf{LC}^{\mathsf{fv}}_L$-proof $\langle \delta, \sigma' \rangle$ where σ' grounds δ. ∎

Proof. Use Lemma 6.9 and create the $\mathsf{LC}^{\mathsf{fv}}_L$-derivation $\langle \delta, \sigma' \rangle$ where σ' grounds δ. Now to show that $\langle \delta, \sigma' \rangle$ is an $\mathsf{LC}^{\mathsf{fv}}_L$-proof. Assume $X[s], Y[t]$ is a closing pair in $\langle \delta, \sigma \rangle$, and $s\sigma$ and $t\sigma$ are not ground labels. Then $X[s], Y[t]$ is still a closing pair in $\langle \delta, \sigma' \rangle$, as $s\sigma' = t\sigma' = \{\epsilon\}$. □

The syntactical proof of soundness goes by the same lines as the proof of soundness in [Waaler 2001], except the proof in [ibid] has one shortcoming: the reduction ordering is assumed well founded and the need for cycle elimination is not appreciated.

Proof sketch (of Theorem 6.27, syntactical). Use Corollary 6.28 to construct an $\mathsf{LC}^{\mathsf{fv}}_L$-derivation $\langle \delta, \sigma' \rangle$ from $\langle \delta, \sigma \rangle$, where σ' is grounding for δ. Utilize Conjecture 6.25 to repeatedly remove all cycles in $\langle \delta, \sigma' \rangle$ obtaining the cycle free $\mathsf{LC}^{\mathsf{fv}}_L$-proof $\langle \delta', \sigma'' \rangle$ of Γ. Use Corollary 6.12 to permute $\langle \delta', \sigma'' \rangle$ into an $\mathsf{LC}^{\mathsf{fv}}_L$-proof $\langle \delta'', \sigma' \rangle$ conforming to $\rhd_{\sigma'}$. Then $\delta''\sigma''$ is an $\mathsf{LC}^{\mathsf{gt}}_L$-proof. Conclude by soundness of $\mathsf{LC}^{\mathsf{gt}}_L$, Theorem 5.19, Γ is L-valid. □

6.3 Termination

This section establishes a *termination condition* applicable to skeletons in $\mathsf{LC}^{\mathsf{fv}}$-derivations. We present the condition and the results concerning it here, after having proved soundness, to emphasize the fact that the termination condition plays no role in the notion of soundness.[1] However a termination criterion imposes a threat to completeness as the proof search may be stopped to early, hindering the discovery of a proof.

[1]In fact as pointed out in the remark of soundness and completeness in chapter 3 if we impose the termination criterion that "no rule is applicable" we end up with a sound (and dull) calculus.

We state the termination condition as follows: Let $\langle\delta,\sigma\rangle$ be an $\mathsf{LC}^{\mathsf{fv}}_L$-derivation, and let ψ be the set of all ν-inferences in a branch in δ having identical formula numbers. The number of label variables introduced by all the inferences in a set ψ, may not exceed the number of labels $s \in \mathsf{Par}_L$ occurring in δ, for all such set ψ constructed from δ.

If we regard a ν-formula and all its explicit copies as the same formula, the condition simplifies to: a ν-formula may not introduce more label variables in a branch than the number of labels $s \in \mathsf{Par}_L$ occurring in the skeleton.

Remark. A more technical variant is: No digit in the copy history of a formula in δ may be greater than the number of labels occurring in δ and in Par_L.

As pointed out in the discussion of the rules or $\mathsf{LC}^{\mathsf{gt}}$ in chapter 5, a copy history contain information on how many times a formula has gone through the explicit copy of a ν-inference or been the active formula—or the subformula of an active formula—in a ν-inference. Copy histories are therefore efficient means of controlling the replication of ν-formulae. $\circ$

This condition is present solely to prevent an unlimited reproduction of formulae, which the explicit copying of formulae in ν-inferences may be responsible for. As we will see from the completeness proof of $\mathsf{LC}^{\mathsf{fv}}$, the upper bound of formula copying that the termination condition imposes, does not compromise completeness.

That the condition in fact is a termination condition is easy to see. There can only be a limited number of π-formulae occurring in a sequent, and since π-formulae of the same formula number introduce the same label parameter, there is also an upper limit to how many label parameters there may be in a skeleton. This limit is adapted to the number of label variables ν-formulae may introduce.

Example 6.29 (Termination condition) This is the $\mathsf{LC}^{\mathsf{fv}}_{S5}$-proof of the **T**-axiom. An $\mathsf{LC}^{\mathsf{g}}_{S5}$-proof is found in Example 4.14.

$$\dfrac{\dfrac{\Box P^1, P[U_{(1,0)}]^{0.0} \;\vdash\; P}{\Box P \;\vdash\; P}\,U_{(1,0)}}{\vdash\; \underset{1\,2}{\Box P} \underset{3}{\rightarrow} \underset{4}{P}}$$

The only closing substitution for $\langle\delta,\sigma\rangle$ is $\sigma = \{U_{(1,0)} \mapsto \epsilon\}$. This substitution is only admissible in S5, hence $\langle\delta,\sigma\rangle$ is an $\mathsf{LC}^{\mathsf{fv}}_{S5}$-proof.

The skeleton below is obtained by applying an inference to the ν-formula in the leaf sequent of the above skeleton.

$$\cfrac{\cfrac{\cfrac{\Box P^2, P[U_{(1,1)}]^{1.0}, P[U_{(1,0)}]^{0.0} \ \vdash \ P}{\Box P^1, P[U_{(1,0)}]^{0.0} \ \vdash \ P} U_{(1,1)}}{\Box P \ \vdash \ P} U_{(1,0)})}{\vdash \ \underset{1}{\Box} \underset{2}{P} \underset{3}{\rightarrow} \underset{4}{P}}$$

The skeleton breaks the termination condition as the ν-formula having formula number 1 is explicit copied once, as we can see from the copy number of the ν-formula in the leaf sequent, but of the set Par_{S5}, only present in the skeleton. Thus the set of ν-inferences having formula number 1 has introduced 2 label variables, while there is only one label parameter $s \in \mathsf{Par}_{S5}$ in the skeleton. ◦

Definition 6.30 (Complete skeleton)
A skeleton δ is said to be a *complete* skeleton for $\mathsf{LC}^{\mathsf{fv}}_L$, if it conforms to the termination condition in L and every resulting skeleton obtained from δ by rule application of one or finitely many more inferences will violate the termination condition in L. ▪

By complete we imply that there are no more rules applicable to the skeleton, either because there are no non-atomic formulae in the leaves of δ or that applying more inferences to δ would cause a condition set by the definition of $\mathsf{LC}^{\mathsf{fv}}_L$-derivation to be broken. Notice that a skeleton violating the termination condition, may conform to the condition after one or more rule applications.

6.4 Completeness

Completeness of $\mathsf{LC}^{\mathsf{fv}}$ is established following the same approach as done in the completeness proofs of LC^{g} and $\mathsf{LC}^{\mathsf{gt}}$. What is important in the case of $\mathsf{LC}^{\mathsf{fv}}$ is of course the treatment of label variables and how to handle the restrictions we have introduced by the termination condition. We assume that a sequent is not $\mathsf{LC}^{\mathsf{fv}}$-provable, and construct a $\mathsf{LC}^{\mathsf{fv}}$-derivation where its skeleton is complete, ensuring that we have introduced the maximal amount of label variables. Then we construct a label substitution in which the different label variables introduced by ν-formulae having the same formula numbers are mapped to different label parameters. This way we achieve a maximal "diffusion" of label *parameters* introduced by ν-formulae having equal formula numbers, and can easily construct an L-countermodel on the basis of an open branch, just as we did in the completeness proof of LC^{g}.

Theorem 6.31 (Completeness of $\mathsf{LC}^{\mathsf{fv}}$)
If an indexed sequent is not $\mathsf{LC}^{\mathsf{gt}}_L$-provable, it has an L-countermodel. ▪

Proof. Let Γ be an empty labelled indexed sequent, and let $\langle \delta, \sigma_0 \rangle$ be an $\mathsf{LC}^{\mathsf{fv}}_L$-derivation, where δ is a complete skeleton over the sequent Γ. Because of the termination condition the skeleton δ is finite. For every ν-formula in δ there occurs as many ν_0-formulae as there are label parameters in the skeleton, and every introduced label variable is different since every label variable in a branch is uniquely subscripted. Let Par_δ denote the set of ground labels occurring in the skeleton δ.

Since Γ is not $\mathsf{LC}^{\mathsf{fv}}_L$-provable, there can be no label substitution σ closing δ. Construct a label substitution σ such that $U\sigma \in \mathsf{Par}_\delta$ for all label variables U occurring in δ and let $\sigma U_{(i,\kappa.m.\lambda)} \neq \sigma U_{(i,\kappa.n.\lambda)}$ if $n \neq m$. The constructed σ ensures that for a set of ν_0-formulae with equal formula numbers the label variable of every formula is mapped to a different ground label. By construction of σ and by the fact that δ is a complete skeleton, we know that by disregarding indices, every ν-formula under σ in every branch has introduced $\nu_0(t)$ for all $t \in \mathsf{Par}_L \cap \mathsf{Par}_\delta$.

$\langle \delta, \sigma \rangle$ is an $\mathsf{LC}^{\mathsf{fv}}_L$-derivation, but no $\mathsf{LC}^{\mathsf{fv}}_L$-proof, so there is a branch θ in δ which is not closed by σ. Let $\theta\sigma$ be the branch from which an L-model is created on the basis of the recipe found in the completeness proof of $\mathsf{LC}^{\mathsf{g}}_L$ (Theorem 4.25). This L-model is an L-countermodel for the skeleton δ and the sequent Γ. □

Chapter 7

Preliminaries on the Logic of Only Knowing

The logic of Only Knowing, $\mathcal{ONL}$, is an autoepistemic logic providing means of introspecting on an agent's own knowledge and ignorance; or belief and co-belief. As the semantics of $\mathcal{ONL}$ assume that models are maximal, meaning that every possible point is represented in the model, the logic is of importance to defeasable reasoning and to the field of artificial intelligence. Consult [Levesque 1990, Rosati 2001, Solhaug 2004, Waaler 2005; 1994, Waaler et al. 2004] for papers and theses concerning $\mathcal{ONL}$ and other autoepistemic logics.

7.1 Syntax

The language of $\mathcal{ONL}$ includes four modalities denoted B (belief) , C (co-belief), b and c. For reasons of simplicity do not include the modalities $\Box$ and $\Diamond$. They act as abbreviation symbols in the language of $\mathcal{ONL}$.

Definition 7.1 (Core $\mathcal{ONL}$-formula)
The set of core $\mathcal{ONL}$-formulae is the smallest set Σ such that

1. $\mathbf{P} \subset \Sigma$, where $\mathbf{P}$ is the set of propositional letters,
2. if $X \in \Sigma$, then $\neg X \in \Sigma$, $\mathsf{B}X \in \Sigma$, $\mathsf{b}X \in \Sigma$, $\mathsf{C}X \in \Sigma$ and $\mathsf{c}X \in \Sigma$, and
3. if $X, Y \in \Sigma$, then $(X \to Y) \in \Sigma$, $(X \vee Y) \in \Sigma$ and $(X \wedge Y) \in \Sigma$.

If $F \in \mathbf{P}$, then F is called *atomic*. F is called *non-atomic* if $F \in \Sigma \setminus \mathbf{P}$. ∎

Remark. The formula $\mathsf{B}P$ is read "I believe at least P" and $\mathsf{C}P$ is read "I believe at most $\neg P$". Combining these we obtain the "all I know" expression $\mathsf{O}P$ [Levesque 1990], an abbreviation for $\mathsf{B}P \wedge \mathsf{C}\neg P$, asserting that "I believe precisely P".

This thesis is not of philosophical character, so we do not defend our interpretation of the new modalities, but refer instead to [Waaler et al. 2004, cp. 3] for a thorough exposition of the interpretation of modalities in the modal logic Æ, a logic in the family of "Only Knowing" logics. ◦

7.2 Semantics

Next we define a model and satisfaction relation for the language of $\mathcal{ONL}$.

Definition 7.2 ($\mathcal{ONL}$-model)
A *$\mathcal{ONL}$-model* $\mathcal{M}$ for the core $\mathcal{ONL}$-language is a quadruple (W, W^+, W^-, V) where $W^+ \cup W^- = W$ and $V : \mathbf{P} \to \mathcal{P}(W)$. An $\mathcal{ONL}$-model satisfying these conditions is called *weak*.

An $\mathcal{ONL}$-model $\mathcal{M} = (W, W^+, W^-, V)$ is *maximal* if W is the set of all possible points. ▪

We call W^+ the set of *plausible* points in a core model and W^- the set of *implausible* points. The condition stating that W is the set of all possible points, is called the *maximality* condition for a model. "The set of all possible points" means that every possible valuation of propositional formulae is represented in the model. If we only consider two propositional letters P and Q, a maximal model contains at least four points: one point where both P and Q are satisfied, one there both P and Q are not satisfied, one where only P is satisfied and a last point only satisfying Q.

The maximality demand is a model condition for $\mathcal{ONL}$-models. We allow the notion of a weak model, such that we can address $\mathcal{ONL}$-models which are not necessarily *maximal*.

Remark. The maximality condition is the solution to what is called Levesque's diamond axiom: *$\Diamond P$ is provable, if P is consistent in the system*, i.e., $\vdash \Diamond P$, if $P \nvdash \bot$. ◦

Since new formulae and a new model is introduced we need to redefine the satisfaction relation.

Definition 7.3 (Satisfaction)
A satisfaction relation $\models$ is defined by each $\mathcal{ONL}$-model $\mathcal{M} = (W, W^+, W^-, V)$ as the weakest relation closed under the following clauses. $\not\models$ represents the compliment of $\models$.

$$
\begin{array}{lll}
\mathcal{M}, w \models P & \text{iff} & w \in V(P),\ \ P \in \mathbf{P}, \\
\mathcal{M}, w \models (X \wedge Y) & \text{iff} & \mathcal{M}, w \models X \text{ and } \mathcal{M}, w \models Y, \\
\mathcal{M}, w \models (X \vee Y) & \text{iff} & \mathcal{M}, w \models X \text{ or } \mathcal{M}, w \models Y, \\
\mathcal{M}, w \models \neg X & \text{iff} & \mathcal{M}, w \not\models X, \\
\mathcal{M}, w \models (X \rightarrow Y) & \text{iff} & \mathcal{M}, w \not\models X \text{ or } \mathcal{M}, w \models Y, \\
\mathcal{M}, w \models \mathsf{B}X & \text{iff} & \mathcal{M}, w' \models X \text{ for all } w' \in W^+, \\
\mathcal{M}, w \models \mathsf{b}X & \text{iff} & \mathcal{M}, w' \models X \text{ for at least one } w' \in W^+, \\
\mathcal{M}, w \models \mathsf{C}X & \text{iff} & \mathcal{M}, w' \models X \text{ for all } w' \in W^-, \\
\mathcal{M}, w \models \mathsf{c}X & \text{iff} & \mathcal{M}, w' \models X \text{ for at least one } w' \in W^-,
\end{array}
$$

where $w \in W$. ∎

From these to definitions we see that $\mathsf{B}, \mathsf{b}, \mathsf{C}$ and c are K45-modalities, as their set of "accessible points"[1] is either the set of plausible or implausible points, both being a subset of the domain W of a model.

As for the core formulae defined in chapter 2, $\mathcal{M}, w \models F$ is read "F is *satisfied by* the $\mathcal{ONL}$-model $\mathcal{M}'$, *in* the point w". The $\mathcal{ONL}$-formula $\mathcal{M}$ is said to *satisfy* F, hence F is *satisfiable*.

Example 7.4 ($\mathcal{ONL}$-model) Let $\{P, Q\} \subset \mathbf{P}$ and $\mathcal{M} = (W, W^+, W^-, V)$ be a $\mathcal{ONL}$-model, where $W^+ = \{w_1, w_2\}$, $W^- = \{w_3, w_4, w_5\}$ and V is such that $V(P) = \{w_1, w_3, w_5\}$, $V(Q) = \{w_1, w_2, w_4, w_5\}$. The $\mathcal{ONL}$-model is not maximal as there is no point w_x in W such that $w_x \notin V(P) \cup V(Q)$. The points w_1 and w_5, and w_2 and w_4 are equivalent as the same set of propositional letters are satisfied in these points.

- $\mathcal{M}, x \models \mathsf{B}Q$ holds for any $x \in W$ since $\mathcal{M}, w' \models Q$ for all $w' \in W^+$.
- $\mathcal{M}, x \models \mathsf{c}Q$ is holds for every $x \in W$. There is a point $w' \in W^-$, e.g., w_4, such that $\mathcal{M}, w' \models Q$ holds.
- $\mathcal{M}, x \models \mathsf{c}(Q \wedge \mathsf{B}P)$ does not hold for any $x \in W$ as $\mathsf{B}P$ is not satisfied in any point of $\mathcal{M}$.

We include a point w_6 in W^+ such that $w_6 \notin V(P) \cup V(Q)$ to create the new *maximal* $\mathcal{ONL}$-model $\mathcal{M}_2$. The formula $\mathsf{B}Q$ is not satisfied in M_2 as $M_2, x \models Q$ does not hold for every $w' \in W^+$. $\mathcal{M}_2, x \models \mathsf{c}Q$ holds, and $\mathcal{M}_2 \models \mathsf{c}(Q \wedge \mathsf{B}P)$ does not hold, by the fact that $\mathcal{M}, x \models \mathsf{B}P$ does not hold for any point x in $\mathcal{M}$. ∘

[1]See the discussion on page 13.

Because of the demand on $\mathcal{ONL}$-models that $W = W^+ \cup W^-$, the modalities $\Box$ and $\Diamond$ are superfluous. They are instead defined as abbreviation symbols, where $\Box P$ is abbreviated by $\mathsf{B}P \wedge \mathsf{C}P$ and $\mathsf{b}P \vee \mathsf{c}P$ abbreviates $\Diamond P$. If the satisfaction relation were to be defined also for formulae of the form $\Box X$ and $\Diamond X$ if would be as follows:

$$\begin{aligned} \mathcal{M}, w \models \Box X \quad &\text{iff} \quad \mathcal{M}, w' \models X \text{ for all } w' \in W, \\ \mathcal{M}, w \models \Diamond X \quad &\text{iff} \quad \mathcal{M}, w' \models X \text{ for at least one } w' \in W, \end{aligned}$$

where $w \in W$. Of this we can conclude that $\Box$ and $\Diamond$ are S5-modalities in $\mathcal{ONL}$.

Remark. Let $\mathcal{M} = (W, W^+, W^-, V)$ be an $\mathcal{ONL}$-model. An observation which makes the interpretation and understanding of the semantics of $\mathcal{ONL}$ easier to comprehend, is that the constructs $\mathcal{M}_1 = (W, W^+, V)$ and $\mathcal{M}_2 = (W, W^-, V)$ are models are described in Definition 3.10 on page 19. W is a non-empty set of points, W^+ and W^- are subsets of W and V is a valuation. The possibility of W^+ or W^- being equal to the set W creates the opportunity of $\mathcal{M}_1$ and M_2 being S5-models, otherwise the models are K45. The condition $W^+ \cup W^- = W$ on the $\mathcal{ONL}$-model $\mathcal{M}$ lets us conclude that W^+ and W^- cannot both be empty as this contradicts the fact that W is non-empty. ◦

Chapter 8

The Calculus $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$

8.1 Syntax

The syntax of LC_L includes a set of labels, but in order to capture the added richness of the model in the logic of $\mathcal{ONL}$ compared to the K45- and S5-model, we need to fine grain the notion of a label in $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$ from the definition in LC_L. The set of label parameters is comprised by two disjoint sets of label parameters denoted Par^+ and Par^-, as are the set of label variables: $\mathsf{Var}^+ \subseteq \mathsf{Var}$, $\mathsf{Var}^- \subseteq \mathsf{Var}$ and $\mathsf{Var}^+ \cap \mathsf{Var}^- = \emptyset$.

To simplify the exposition let $\pm$ be either $+$ or $-$, e.g. so that $\mathsf{Var}^\pm$ either represents the set Var^+ or the set Var^-. Note that $\pm$ must be treated as parameterized placeholders for the indicated symbols: The statement '$\mathsf{Par}^\pm = \mathsf{Par}^\pm$' is never an abbreviation for '$\mathsf{Par}^+ = \mathsf{Par}^-$' or '$\mathsf{Par}^- = \mathsf{Par}^+$', but always '$\mathsf{Par}^+ = \mathsf{Par}^+$' and '$\mathsf{Par}^- = \mathsf{Par}^-$'.

Uppercase letters denote label variables, $U^\pm, V^\pm, W^\pm, \ldots \in \mathsf{Var}^\pm$, and lowercase letters represent label parameters, $c^\pm, d^\pm, e^\pm \ldots \in \mathsf{Par}^\pm$. $s^\pm$ and $t^\pm$ are used to denote labels in $\mathsf{Var}^\pm \cup \mathsf{Par}^\pm$, while s and t will be used to indicate an arbitrary label.

The notion of an *indexed $\mathcal{ONL}$-formula* is easily lifted from Definition 5.1 by requiring that the underlying formula is a core $\mathcal{ONL}$-formula.

As in the labelled language every non-atomic $\mathcal{ONL}$-formula is of a specific type. The α- and β-formulae are in $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$ defined as in LC_L, but new types ν^+, ν^-, π^+ and π^- are added. With the new types, new components are also specified.

Definition 8.1 (Type, Component)

The type and component of a non-atomic indexed $\mathcal{ONL}$-formula is defined by its outermost connective/modal operator as defined by the tables for the types α and β in the Figure 3.1 and the types in Figure 8.1. ∎

ν^+	$\nu_0^+(t)$
$\mathsf{B}X[s]_\top$	$X[t^+]_\top$
$\mathsf{b}X[s]_\bot$	$X[t^+]_\bot$

π^+	$\pi_0^+(t)$
$\mathsf{b}X[s]_\top$	$X[t^+]_\top$
$\mathsf{B}X[s]_\bot$	$X[t^+]_\bot$

ν^-	$\nu_0^-(t)$
$\mathsf{C}X[s]_\top$	$X[t^-]_\top$
$\mathsf{c}X[s]_\bot$	$X[t^-]_\bot$

π^-	$\pi_0^-(t)$
$\mathsf{c}X[s]_\top$	$X[t^-]_\top$
$\mathsf{C}X[s]_\bot$	$X[t^-]_\bot$

Figure 8.1: $\mathcal{ONL}$-formula types ν^+, ν^-, π^+ and π^-, and their components.

By Definition 8.1 the notions of immediate subformula and (proper) subformula are easily defined by using Definition 3.8 and 5.3.

8.2 Semantics

The introduction of the new more fine grained labels are followed up by a more sensitive label interpretation then used for the modal logics K45 and S5..

Definition 8.2 (Label interpretation)
Let ϕ_ρ be a *label interpretation function* where $\phi_\rho = \phi \cup \rho$, and ϕ and ρ are defined the following functions:

- $\phi^+ : \mathsf{Par}^+ \to W^+$,
- $\phi^- : \mathsf{Par}^- \to W^-$,
- $\phi : \mathsf{Par} \cup \{\epsilon\} \to W$ such that $\phi(c^\pm) = \phi^\pm(c^\pm)$ for all $c^\pm \in \mathsf{Par}^\pm$.

ϕ is called a *ground label interpretation*. To interpret label variables we define the following functions:

- $\rho^+ : \mathsf{Var}^+ \to W^+$,
- $\rho^- : \mathsf{Var}^- \to W^-$,
- $\rho : \mathsf{Var} \to W$ such that $\rho(U^\pm) = \rho^\pm(U^\pm)$ for all $U^\pm \in \mathsf{Var}^\pm$,

We call ρ a *label variable assignment*. ∎

Definition 8.3 ($\mathcal{ONL}$-model)
An $\mathcal{ONL}$*-model* is a pair $(\mathcal{M}, \phi)$, where $\mathcal{M} = (W, W^+, W^-, V)$ is a model in the core $\mathcal{ONL}$-language and ϕ is a ground label interpretation on $\mathcal{M}$.

An $\mathcal{ONL}$-model $(\mathcal{M}, \phi)$ is *weak* if $\mathcal{M}$ is weak, and *maximal* if $\mathcal{M}$ is maximal. ▪

Truth of an $\mathcal{ONL}$-formula in an $\mathcal{ONL}$-model $(\mathcal{M}, \phi)$ is always defined relative to a label variable assignment ρ on $\mathcal{M}$. The definition is the result of extending the definition of truth in Definition 3.11 (truth for labelled formulae) and 5.5 (truth for indexed formulae) in the obvious way to indexed $\mathcal{ONL}$-formulae using Definition 7.3 and 8.3.

Now we are ready to extend Lemma 3.15 (Satisfaction of components) to all the types of indexed $\mathcal{ONL}$-formulae.

Lemma 8.4 (Satisfaction of components) Let Γ be a set of indexed $\mathcal{ONL}$-formulae, $(\mathcal{M}, \phi)$ an $\mathcal{ONL}$-model where $\mathcal{M} = (W, W^+, W^-, V)$, and ρ some variable assignment. For all α-, β-, $\nu^\pm$- and $\pi^\pm$-formulae, and their respective components:

1. if $\alpha \in \Gamma$, then $\Gamma \cup \{\alpha_1, \alpha_2\}$ is satisfiable in $(\mathcal{M}, \phi)$ under ρ, iff $(\mathcal{M}, \phi)$ satisfies Γ under ρ,
2. if $\beta \in \Gamma$, then $\Gamma \cup \{\beta_1\}$ is satisfiable in $(\mathcal{M}, \phi)$ under ρ or $\Gamma \cup \{\beta_2\}$ is satisfiable in $(\mathcal{M}, \phi)$ under ρ, iff $(\mathcal{M}, \phi)$ satisfies Γ under ρ,

3.1. if $\nu^\pm \in \Gamma$, $W^\pm \neq \emptyset$ and $(\mathcal{M}, \phi)$ satisfies Γ, then $\Gamma \cup \{\nu_0^\pm(s)\}$ is satisfiable in $(\mathcal{M}, \phi)$ under ρ for every label $s \in \mathsf{Par}^\pm \cup \{\epsilon\}$,

3.2. if $\nu^\pm \in \Gamma$ and $(\mathcal{M}, \phi)$ satisfies Γ, then $\Gamma \cup \{\nu_0^\pm(U^\pm)\}$ is satisfiable in $(\mathcal{M}, \phi)$ under ρ for every $U^\pm \in \mathsf{Var}^\pm$, and

4. if $\pi^\pm \in \Gamma$, then $\Gamma \cup \{\pi_0^\pm(c^\pm)\}$ is satisfiable in $(\mathcal{M}, \phi')$ under ρ, iff $(\mathcal{M}, \phi)$ satisfies Γ under ρ, where $c^\pm$ is an arbitrary label parameter not occurring in Γ or π, and ϕ' is an extension by $c^\pm$ of ϕ. ▪

Corollary 8.5 (Satisfaction of grounded components) Let Γ be a set of ground indexed $\mathcal{ONL}$-formulae and $(\mathcal{M}, \phi)$ an $\mathcal{ONL}$-model where $\mathcal{M} = (W, W^+, W^-, V)$.

1. If $\alpha \in \Gamma$, then $\Gamma \cup \{\alpha_1, \alpha_2\}$ is true in $(\mathcal{M}, \phi)$, iff Γ is true in $(\mathcal{M}, \phi)$.
2. If $\beta \in \Gamma$, then $\Gamma \cup \{\beta_1\}$ is true in $(\mathcal{M}, \phi)$ or $\Gamma \cup \{\beta_2\}$ is true in $(\mathcal{M}, \phi)$, iff Γ is true in $(\mathcal{M}, \phi)$.
3. If $\nu^\pm \in \Gamma$ and $W^\pm \neq \emptyset$, then $\Gamma \cup \{\nu_0^\pm(s^\pm)\}$ is true in $(\mathcal{M}, \phi)$ for every label $s^\pm \in \mathsf{Par}^\pm$.

4. If $\pi^{\pm} \in \Gamma$, then $\Gamma \cup \{\pi_0^{\pm}(c^{\pm})\}$ is true in $(\mathcal{M}, \phi')$ iff Γ is true in $(\mathcal{M}, \phi)$, where $c^{\pm} \in \mathsf{Par}^{\pm}$ is an arbitrary label parameter not occurring in Γ or π and ϕ' is an extension by $c^{\pm}$ of ϕ. ▪

Proof sketch (of Lemma 8.4 and Corollary 8.5). The proofs are almost identical to the proofs of Lemma 3.15 and Corollary 4.20, and are left to the reader. The proofs of cases 1 and 2 are obtained by rewriting the proofs found in chapter 3 and 4 using the an $\mathcal{ONL}$-model and applying definitions established for formulae in $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$, and not the formulae in LC_{K45} and LC_{S5} to show the equivalence steps used in the proofs. For the cases 3.1, 3.2 and 4 one additionally has to put '$\pm$' as superscript on the correct places. □

8.3 Calculus

The proof search in $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$ is comprised by two steps. First we search for a weak $\mathcal{ONL}$-model satisfying the root sequent. This is done through the use of a derivation similar to the one defined in the calculi defined in earlier chapters. The derivation combines two label substitutions, each simulating one of the modalities B and C as a S5-modality by positioning the point to which ϵ is interpreted to either the set W^+ or the set W^-. If this search is successful, we have found a weak model satisfying the root sequent of the derivation. The second step is to test this model for maximality. This is done by an Aux-*derivation* looking for the existence of a point $w \notin W^+ \cup W^-$. If such a point can exist in the weak model retrieved by the first step in the proof search, the model is not maximal and the root sequent is hence not satisfiable in a maximal model. If every weak model with the root sequent is found not maximal, there is no maximal $\mathcal{ONL}$-model satisfying the root sequent and the sequent must be $\mathcal{ONL}$-valid.

Remark. The idea of an Aux-derivation is gathered from Rosati [2001] notion of an auxiliary tableau. The construction of a free-variable Aux-derivation and the ideas of finding a weak model using nearly exact same constructs as in $\mathsf{LC}^{\mathsf{fv}}_{K45/S5}$ together with the use of two label substitutions, are due to me. ◦

Definition 8.6 ($\mathcal{ONL}$-sequent)
An $\mathcal{ONL}$-sequent is an indexed sequent (Definition 5.7) of indexed $\mathcal{ONL}$-formulae. ▪

The rules of $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$ is the set of rules listed in Figure 8.2. These rules operate on skeletons of indexed sequents of $\mathcal{ONL}$-formulae.

$$\frac{\Gamma, \alpha_1^\kappa, \alpha_2^\kappa}{\Gamma, \alpha^\kappa} \qquad \frac{\Gamma, \beta_1^\kappa \quad \Gamma, \beta_2^\kappa}{\Gamma, \beta^\kappa} \qquad \frac{\Gamma, \underset{i}{\nu}^{\pm\kappa+1}, \nu_0(U^\pm_{(i,\kappa)})^{\kappa.0}}{\Gamma, \underset{i}{\nu}^{\pm\kappa}} \; U^\pm_{(i,\kappa)} \qquad \frac{\Gamma, \pi_0^{\pm\kappa}(c_i^\pm)}{\Gamma, \underset{i}{\pi}^{\pm\kappa}} \; c_i^\pm$$

Figure 8.2: The rules of $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$. The rules from left to right are: the α- β- ν^+- and ν^--, and π^+- and π^--rule.

Note that the $\nu^\pm$- and $\pi^\pm$-rules are specializations of the ν- and π-rules formulated in $\mathsf{LC}^{\mathsf{fv}}_{K45/S5}$, and by consulting Lemma 8.4 and Corollary 8.5 observe that the semantics of the types $\nu^\pm$ and $\pi^\pm$ in $\mathcal{ONL}$ share the same qualities as the related types in $\mathsf{LC}^{\mathsf{fv}}_{K45/S5}$ except they are limited to restricted subsets of the Par and Var.

Definition 8.7 ($\sigma^{\mathsf{B}}, \sigma^{\mathsf{C}}$)
Let $\sigma^\pm$ and $\sigma_\epsilon\pm$ be label substitutions such that $\sigma^\pm : \mathsf{Var}^\pm \to \mathsf{Par}^\pm$ and $\sigma_\epsilon^\pm : \mathsf{Var}^\pm \to \mathsf{Par}^\pm \cup \{\epsilon\}$. Then σ^{B} and σ^{C} are the label substitutions $\sigma^{\mathsf{B}} = \sigma_\epsilon^+ \cup \sigma^-$ and $\sigma^{\mathsf{C}} = \sigma_\epsilon^- \cup \sigma^+$. ∎

The intuitions behind the label substitutions σ^{B} and σ^{C} are that σ^+ treats B and b as K45-modalities and σ_ϵ^+ treats them as S5-modalities, while σ^- acts as though C and c are K45-modalities and σ_ϵ^- as they are S5-modalities. This gives us that σ^{B} treats B and b as S5-modalities, and C and c as K45-modalities. σ^{C} treats the modalities vice versa. The technical difference between them is that σ^{B} only may map label variables $U^+ \in Par^+$ to ϵ, while σ^{C} only may map label variables $U^- \in Par^-$ to ϵ.

In the LC-calculi the label ϵ represents the initial or start point in W, the point from with we start the search for a countermodel. In the cases of the logic being K45 ϵ was always interpreted to a point in $W \setminus W^+$ to ensure that the model would not be reflexive. The situation in an $\mathcal{ONL}$-model is slightly different. Because of the condition $W = W^+ \cup W^-$, ϵ is always interpreted to either W^+ or W^- (or both[1]). This has consequences for the satisfaction of modalized formulae. Assume ϵ is interpreted to W^+, then the formula $\mathsf{B}P \to P$ holds, making B an S5-modality. But we cannot assume that ϵ is in fact interpreted to a point in W^+, so $\mathsf{B}P \to P$ should not

[1]No condition states that $W^+ \cap W^- = \emptyset$, that the model is *bisected* [Waaler et al. 2004], and this is not a model condition for a (core) $\mathcal{ONL}$-model. However, the model is perhaps more comprehendible if one thinks of the sets W^+ and W^- as disjunct, at least if this thesis is one's first encounter with an $\mathcal{ONL}$-model. If $W^+ \cap W^- \neq \emptyset$, then there are points with are both plausible and implausible, quoting [Waaler et al. 2004, p. 13]: *"A non-bisected model may be interpreted as a model of a subject that has discrepancy between caution and belief; between caution to not accept evidence and actual acceptance. In a bisected model , however, there is a co-belief to match every belief; just the right amount of caution to match what is believed"*.

be (and is not) provable in $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$. However, in every $\mathcal{ONL}$-model $(\mathcal{M}, \phi)$ either $\mathsf{B}P \rightarrow P$ or $\mathsf{C}P \rightarrow P$ should be provable, as ϵ must be interpreted to either W^+ or W^-. By this observation the formula $(\mathsf{B}P \rightarrow P) \vee (\mathsf{C}P \rightarrow P)$ should be valid in an $\mathcal{ONL}$-model. In fact $(\mathsf{B}P \rightarrow P) \vee (\mathsf{C}P \rightarrow P)$ is equivalent to the formula $(\mathsf{B}P \wedge \mathsf{C}P) \rightarrow P$, to which $\Box P \rightarrow P$ abbreviates, making $\Box$ an S5-modality which is exactly what we want.

Given that the label substitutions σ^{B} and σ^{C} consider the modalities B, b, C and c as either S5- and K45-modalities, the $\mathcal{ONL}$-calculus can be seen as a compound calculus of $\mathsf{LC}^{\mathsf{fv}}_{K45/S5}$-calculi with one exception addressed by the Aux-derivation defined later: in contrast to an $\mathcal{ONL}$-model a K45/S5-model is not required to be maximal. By these observations an $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-derivation is defined as follows.

Definition 8.8 ($\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-derivation)
The triplet $\langle \delta, \sigma^{\mathsf{B}}, \sigma^{\mathsf{C}} \rangle$, where δ is a skeleton, and $\sigma^{\mathsf{B}} = \sigma^{+}_{\epsilon} \cup \sigma^{-}$ and $\sigma^{\mathsf{C}} = \sigma^{-}_{\epsilon} \cup \sigma^{+}$ are label substitutions as defined in Definition 8.7, is an $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-derivation if

- the root sequent of δ is empty labelled,
- every inference in δ is an instance of a rule in the $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-calculus,
- δ is a complete skeleton,
- $\sigma^{\pm}$ conform to the same conditions as for an $\mathsf{LC}^{\mathsf{fv}}_{K45}$-admissible label substitution σ adapted to the domain and codomain of $\sigma^{\pm}$. ■

Definition 8.9 (($\sigma^{\mathsf{B}}, \sigma^{\mathsf{C}}$)-axiom)
Let σ^{B} and σ^{C} be label substitutions as defined in Definition 8.7. A sequent Γ is a $(\sigma^{\mathsf{B}}, \sigma^{\mathsf{C}})$-axiom if Γ is a σ^{B}- and a σ^{C}- axiom by Definition 6.19. ■

Lemma 8.10 A $(\sigma^{\mathsf{B}}, \sigma^{\mathsf{C}})$-axiom is not satisfiable in any $\mathcal{ONL}$-model $(\mathcal{M}, \phi)$. ■

Proof. Left for the reader. Study proof of Definition 6.19. □

If every branch in the skeleton of an $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-derivation contains a $(\sigma^{\mathsf{B}}, \sigma^{\mathsf{C}})$-axiom we say the derivation is *closed*, otherwise the derivation is *open*.

Conjecture 8.11 (Satisfiability of open $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-derivation) An $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-derivation $\langle \delta, \sigma^{\mathsf{B}}, \sigma^{\mathsf{C}} \rangle$ is closed if and only if there is no weak $\mathcal{ONL}$-model $(\mathcal{M}, \phi)$ satisfying the root sequent of δ. ■

Proof sketch. Use Lemma 8.4 and 8.10 and conclude by soundness and completeness of $\mathsf{LC}^{\mathsf{fv}}_{K45}$ and $\mathsf{LC}^{\mathsf{fv}}_{S5}$. □

Example 8.12 (□ is an S5-modality) Call the skeleton below δ. The root sequent is the rewritten T-axiom for the modal operator □ using the abbreviation $\Box X \equiv \mathsf{B}X \wedge \mathsf{C}X$. Formula numbers and copy histories are not displayed.

$$\dfrac{\dfrac{\dfrac{\dfrac{\mathsf{B}P, P[U^+], \mathsf{C}P, P[U^-] \vdash P}{\mathsf{B}P, P[U^+], \mathsf{C}P \vdash P}\,U^-}{\mathsf{B}P, \mathsf{C}P \vdash P}\,U^+}{\mathsf{B}P \wedge \mathsf{C}P \vdash P}}{\vdash (\mathsf{B}P \wedge \mathsf{C}P) \rightarrow P}$$

Let $\sigma^{\mathsf{B}} = \{U^+ \mapsto \epsilon\}$ and $\sigma^{\mathsf{C}} = \{U^- \mapsto \epsilon\}$ then $\langle \delta, \sigma^{\mathsf{B}}, \sigma^{\mathsf{C}} \rangle$ is an $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-derivation and it is closed since the leaf sequent of δ is a $(\sigma^{\mathsf{B}}, \sigma^{\mathsf{C}})$-axiom. By Conjecture 8.11 there is no weak $\mathcal{ONL}$-model satisfying the root sequent. Thus, there can be no $\mathcal{ONL}$-model satisfying the root sequent, hence the sequent is $\mathcal{ONL}$-valid. ◦

Definition 8.13 (Aux-derivation)
An Aux-skeleton δ_{Aux} for a set of formulae Γ is a skeleton whose leaf sequents Γ_i, are constructed as follows: Let Γ_{Var} be the set of atomic formulae of Γ whose label is a label variable, and let c be a label parameter. For all $F[U] \in \Gamma_{\mathsf{Var}}$ construct a leaf sequent Γ_i such that

$$\Gamma_i = \begin{cases} \Gamma_{\mathsf{Var}} \cup \{F[c]_\bot\} & \text{if the polarity of } \underset{i}{F}[U] \text{ is } \top \\ \Gamma_{\mathsf{Var}} \cup \{F[c]_\top\} & \text{if the polarity of } \underset{i}{F}[U] \text{ is } \bot. \end{cases}$$

Let $\sigma_c^+ : \mathsf{Var}^+ \rightarrow \{c\}$ and $\sigma_c^- : \mathsf{Var}^- \rightarrow \{c\}$.

An Aux-derivation is a triplet $\langle \delta_{\mathsf{Aux}}, \sigma_c^+, \sigma_c^- \rangle$. An Aux-derivation is closed if all leaf sequents in the Aux-skeleton δ_{Aux} is both a σ_c^+- and a σ_c^--axiom by Definition 6.19. ▪

Note that an Aux-derivation with no leaves is trivially closed.

Conjecture 8.14 (Maximality test) Assume $(\mathcal{M}, \phi)$ is a weak $\mathcal{ONL}$-model satisfying the leaf sequent Γ of an open branch in the complete skeleton of an $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-derivation. There is no closed Aux-derivation for Γ, if and only if $(\mathcal{M}, \phi)$ is not maximal. ▪

Before of proof sketch is given we first explain the ideas behind the makings of an Aux-derivation. Let Γ be as in the Lemma. The formulae in Γ labelled with label variables guide the construction of the sets W^+ and W^- of a possible $\mathcal{ONL}$-countermodel for Γ. If the formulae $P[U^+]_\top$ and $Q[U^-]_\top$ occur in Γ, then $P_\top$ is satisfied in every point in W^+ and $Q_\top$ is satisfied in every point in W^- of a weak $\mathcal{ONL}$-model satisfying Γ. If an $\mathcal{ONL}$-model is to be maximal, there can be no point where both $\neg P$ and $\neg Q$ is satisfied, since this implies the existence of a point not in $W^+ \cup W^-$. To check this we test if the set $\{P[U^+]_\top, Q[U^-]_\top, \Diamond(\neg P \wedge \neg Q)_\top\}$ is satisfied in a weak $\mathcal{ONL}$-model. Applying rules to this sequent, without using the abbreviation for the $\Diamond$-symbol and assuming the inference on the π-formula belongs to the label parameter c, we obtain exactly the leaves as described in the definition of an Aux-derivation. Omitting the rewriting of the π-formula to one π^+- and one π^--formula is compensated by the label substitutions σ_c^+ and σ_c^-.

If a leaf sequent of the Aux-derivation is no (σ_c^+, σ_c^-)-axiom, then there is a weak $\mathcal{ONL}$-model satisfying the set $\{P[U^+]_\top, Q[U^-]_\top, \Diamond(\neg P \wedge \neg Q)_\top\}$. If such an $\mathcal{ONL}$-model is found, it means that the initial model satisfying Γ cannot be maximal, since the formulae in Γ do not force the weak model for Γ to be a maximal. If no such model is found, i.e., the Aux-derivation is closed, then the initial weak model satisfying Γ is maximal.

Proof sketch (of Conjecture 8.14). Conclude by soundness and completeness of $\mathsf{LC}^{\mathsf{fv}}_{K45/S5}$. □

Definition 8.15 ($\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-proof)
An $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-derivation $\langle \delta, \sigma^{\mathsf{B}}, \sigma^{\mathsf{C}} \rangle$ is an $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-proof if for every branch θ in δ,

1. θ contains a $(\sigma^{\mathsf{B}}, \sigma^{\mathsf{C}})$-axiom, or
2. if θ does not contain a $(\sigma^{\mathsf{B}}, \sigma^{\mathsf{C}})$-axiom, there is no closed Aux-derivation of the leaf sequent of θ. ■

Example 8.16 (Complete skeleton, $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-proof) The skeleton δ displayed below is a complete skeleton. We do not display formula numbers or copy numbers in the skeleton, but use subscripts to indicate explicit copies, descendants of explicit copies and labels introduced by explicit copied formulae.

As the skeleton contains one label parameter from the set Par^+ formulae of the type ν^+ may introduce two label variables.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{Q[c^+],\mathsf{C}P,P[U^-] \vdash P[c^+]}{Q[c^+],\mathsf{C}P \vdash P[c^+]}U^- \qquad \dfrac{\dfrac{\dfrac{\dfrac{Q[c^+],\mathsf{B}\neg Q_3, \vdash P[c^+],Q[U^+],Q_2[U_2^+]}{Q[c^+],\mathsf{B}\neg Q_3,\neg Q_2[U_2^+] \vdash P[c^+],Q[U^+]}}{Q[c^+],\mathsf{B}\neg Q_2 \vdash P[c^+],Q[U^+]}U_2^+}{Q[c^+],\mathsf{B}\neg Q_2,\neg Q[U^+] \vdash P[c^+]}}{Q[c^+],\mathsf{B}\neg Q \vdash P[c^+]}U^+}{Q[c^+],\mathsf{C}P \lor \mathsf{B}\neg Q \vdash P[c^+]}}{\neg P[c^+],Q[c^+],\mathsf{C}P \lor \mathsf{B}\neg Q \vdash}}{\neg P \land Q[c^+],\mathsf{C}P \lor \mathsf{B}\neg Q \vdash}c^+}{\mathsf{b}(\neg P \land Q),\mathsf{C}P \lor \mathsf{B}\neg Q \vdash}}{\mathsf{b}(\neg P \land Q),\mathsf{C}P \lor \mathsf{B}\neg Q \vdash}$$

The $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-derivation $\langle \delta, \sigma^{\mathsf{B}}, \sigma^{\mathsf{C}} \rangle$, where $\sigma^{\mathsf{B}} = \sigma^{\mathsf{C}} = \{U^+ \mapsto c^+\}$ has open branches. The right branch contains an $(\sigma^{\mathsf{B}}, \sigma^{\mathsf{B}})$-axiom, the leaf sequent, while the left branch is not closable. (Note that we cannot map U^- to c^+ to close the left branch.)

Since the left branch is open we perform a maximality test to check if the weak model satisfying the open branch of δ is maximal. $P[U^-]_\top$ is the only formula in the leaf sequent of the open branch, giving $\{P[U^-]_\top, P[s]_\bot\}$ as the only leaf sequent in the Aux-derivation. This sequent is not a (σ_s^+, σ_s^-)-axiom as the label substitution σ_s^+ does not create a closing pair. Since the Aux-derivation of the leaf sequent of an open branch is not closed, the $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-derivation is an $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-proof. ◦

8.4 Soundness and Completeness

The soundness and completeness results for $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$ are established by relying on the Conjecture 8.11 and 8.14.

Conjecture 8.17 (Soundness of $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$) Let $\langle \delta, \sigma^{\mathsf{B}}, \sigma^{\mathsf{C}} \rangle$ be an $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-derivation be a proof on the indexed $\mathcal{ONL}$-sequent Γ. Then Γ is $\mathcal{ONL}$-valid. ■

Proof. Let $\langle \delta, \sigma^{\mathsf{B}}, \sigma^{\mathsf{C}} \rangle$ and Γ be as in the theorem. We split the proof in two. First assume $\langle \delta, \sigma^{\mathsf{B}}, \sigma^{\mathsf{C}} \rangle$ is an $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-proof because every branch in $\langle \delta, \sigma^{\mathsf{B}}, \sigma^{\mathsf{C}} \rangle$ contains an $(\sigma^{\mathsf{B}}, \sigma^{\mathsf{C}})$-axiom. By Conjecture 8.11 there is no weak model satisfying Γ, and there can be no $\mathcal{ONL}$-model satisfying Γ. Γ has no $\mathcal{ONL}$-countermodel, hence Γ is $\mathcal{ONL}$-valid.

Lastly assume $\langle \delta, \sigma^{\mathsf{B}}, \sigma^{\mathsf{C}} \rangle$ is an $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-proof and that one of more of the branches in the proof is open. Then there is a weak $\mathcal{ONL}$-model for every open branch which by Conjecture 8.11 satisfies Γ. But since $\langle \delta, \sigma^{\mathsf{B}}, \sigma^{\mathsf{C}} \rangle$ is an $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-proof, there is no closed Aux-derivation for any of the leaf sequents of the open branches, which let us conclude by Conjecture 8.14 that none of the weak $\mathcal{ONL}$-models satisfying Γ are maximal. No $\mathcal{ONL}$-model is an $\mathcal{ONL}$-countermodel for Γ, Γ is $\mathcal{ONL}$-valid. □

The next lemma is an easy adaptation of a result in $\mathsf{LC}^{\mathsf{fv}}_{K45/S5}$ and is needed in the proof of completeness of $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$.

Lemma 8.18 (Weak model preservation of inferences in $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$) If there is a weak $\mathcal{ONL}$-model satisfies the denominator of an inference, then there is a weak $\mathcal{ONL}$-model satisfying one of the sequents in the numerator. ■

Proof sketch. Adapt the proof of Lemma 6.26, countermodel preservation [in $\mathsf{LC}^{\mathsf{fv}}_{K45/S5}$]. □

Conjecture 8.19 (Completeness of $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$) If an indexed $\mathcal{ONL}$-sequent is not $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-provable, it has an $\mathcal{ONL}$-countermodel. ■

Proof. Let Γ be an indexed $\mathcal{ONL}$-sequent which is not $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-provable. When there exists no $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-proof $\langle \delta, \sigma^{\mathsf{B}}, \sigma^{\mathsf{C}} \rangle$ where Γ is the root sequent of δ.

Let $\langle \delta, \sigma^{\mathsf{B}}, \sigma^{\mathsf{C}} \rangle$ be an $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-derivation where Γ is the root sequent of Γ. Then there is an open branch in $\langle \delta, \sigma^{\mathsf{B}}, \sigma^{\mathsf{C}} \rangle$. By Conjecture 8.11 and in turn Lemma 8.18 there is a weak $\mathcal{ONL}$-model $(\mathcal{M}, \phi)$ satisfying Γ and all the sequents in a branch θ in δ. Since $\langle \delta, \sigma^{\mathsf{B}}, \sigma^{\mathsf{C}} \rangle$ is no $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-proof, there is an open branch where there is no closed Aux-derivation of the leaf sequent of the open branch. Let θ be such a branch. By Conjecture 8.14 the weak $\mathcal{ONL}$-model satisfying Γ is maximal, and is hence an maximal $\mathcal{ONL}$-countermodel for Γ. □

Chapter 9

Future work

The work on this thesis has given valuable knowledge concerning difficult issues, and opened up for new interesting problems.

$\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$*-calculus.* The sketch on the $\mathsf{LC}^{\mathsf{fv}}_{\mathcal{ONL}}$-calculus needs to be further developed.

Cycle elimination. Establishing cycle elimination for $\mathsf{LC}^{\mathsf{fv}}$.

Removal of the neighbourhood condition. There are reasons to believe that the neighbourhood condition adapted from [Waaler 2001] may be dropped entirely in $\mathsf{LC}^{\mathsf{gt}}$, and also in $\mathsf{LC}^{\mathsf{fv}}$, but here on the account of a small alteration in the non-empty W^+-condition to better accommodate the non-empty W^+ condition in $\mathsf{LC}^{\mathsf{gt}}$. The reason why the author thinks the neighbourhood condition may be dropped from $\mathsf{LC}^{\mathsf{gt}}$, is that there exists an eigenparameter condition between π-inferences which is upheld by the fact that only equally formula numbered π-formulae introduce equal label parameters. This is believed to be as a sufficient condition to ensure soundness of the system. The time of discovery for this feature was however to late to establish a complete proof of the countermodel preservation lemma for π-inferences. The problem is that an unfresh label parameter introduction by a π-inference can no longer be traced back to a fresh introduction of the same label by another π-inference, an assumption which is present due to the neighbourhood condition in $\mathsf{LC}^{\mathsf{gt}}$.

Splitting calculus. Applying the splitting calculus [Antonsen and Waaler 2006] to the $\mathsf{LC}^{\mathsf{fv}}$-calculus. The splitting calculus provides a branch-wise restriction to the search space. This quality would lower the upper bounds used in the termination condition dramatically. The indexing of formulae and reduction ordering used in this thesis are similar to constructs used in [ibid].

The sequent

$$\Box P \vdash \Box\Box P \wedge \Box\Box P \wedge \ldots \wedge \Box\Box P$$

creates a derivation containing one branch for each conjunction in the large conjunct. In $\mathsf{LC}^{\mathsf{fv}}$ the formula will cause the ν-formula in one of the branches to be replicated as many times as there are conjuncts in the large conjunction. This is because of the regulations enforced by the contextually equivalent relation. The splitting calculus liberates this regulation leading to a decrease in the necessary explicit copies. The proof in the splitting calculus would in all likelihood require no explicit copying of ν-formulae.

The author believes that the application of the splitting calculus can result in a dramatic reduction on the size of derivation, by making explicit copying a redundant action. This may however require the input formula to be rewritten to a normal form without nested modalities.

Implementation. The goal of every constructor of calculi should is to properly test it, and the best way to do so is build a prototype implementation.

Bibliography

Roger Antonsen. Free variable sequent calculi. Master's thesis, University of Oslo, 2003.

Roger Antonsen and Arild Waaler. Liberalized variable splitting. To appear in Journal of Automated Reasoning 2006, 2006.

Bernhard Beckert and Rajeev Goré. Free variable tableaux for propositional modal logics. In *Proc. International Conference on Theorem Proving with Analytic Tableaux and Related Methods, Pont-a-Mousson, France*, volume 1227, pages 91–106. Springer-Verlag, 1997.

E. Bencivenga. Free logic. In D. Gabbay and F. Günther, editors, *Handbook of Philsophical Logic*, volume 3. Kluwer, Dordrecth, 1986.

Wolfgang Bibel. *Automated Theorem Proving*. Vieweg, Braunschweig, 2nd revised edition, 1987.

Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. Cambridge University Press, 2nd paperback edition, 2005.

Brian F. Chellas. *Modal logic: an introduction*. Cambridge University Press, 1988.

Melvin Fitting. *First-Order Logic and Automated Theorem Proving*. Springer-Verlag, 2nd edition, 1996.

Melvin C. Fitting. *Proof Methods for Modal and Intuitionistic Logics*. D. Reidel Publishing Co., Dordrecht, 1983.

D. Gabbay. *Labelled Deductive Systems*. Oxford University Press, 1996.

Michael R. Garey and David S. Johnsen. *Computers and Intractability. A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

Gerhard Gentzen. Untersuchungen über das logische Schließen. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1934-35. English translation in

M.E. Szabo *The Collected Papers of Gerhard Gentzen*, North-Holland, Amsterdam, 1969.

Christian Mahesh Hansen. Incremental Proof Search in the Splitting Calculus. Master's thesis, Dept. of Informatics, University of Oslo, 2004.

Christian Mahesh Hansen. Completeness for First-Order Tableaux. Lecture notes INF4170, University of Oslo, 2005.

Bjarne Holen. A Reflective Theorem Prover for the Connection Calculus. Master's thesis, Dept. of Informatics, University of Oslo, 2005.

G. E. Hughes and M. J. Cresswell. *An Introduction to Modal Logic*. Methuen, London, 1968.

S. Kanger. A Simplified Proof Method of Elementary Logic. In J. Siekmann and G. Wrightson, editors, *Automation of Reasoning 1: Classical Paperss in Computational Logic 1957–1966*, pages 364–371. Berlin Heidelberg: Springer, 1983.

Christoph Kreitz and Jens Otten. Connection-based Theorem Proving in Classical and Non-classical Logics. *Journal of Universal Computer Science*, 5(3):88–112, 1999.

Richard E. Ladner. The Computational Complexity of Provability in Systems of Modal Propositional Logic. *SIAM J. Comput.*, 6(3):467–480, 1977.

Hector J. Levesque. All I Know: A Study in Autoepistemic Logic. *Artificial Intelligence*, 42:263–306, 1990.

Fabio Massacci. Single Step Tableaux for Modal Logics. *Journal of Automated Reasoning*, 24:319–364, 2000.

Grigori Mints. *A Short Introduction to Modal Logic*. The University of Chicago Press, 1992.

Jens Otten. ileanTAP: An Intuitionistic Theorem Prover. In *TABLEAUX*, pages 307–312, 1997.

John Alan Robinson. A Machine-Oriented Logic Based on the Resolution Principle. *J. ACM*, 12(1):23–41, 1965.

Riccardo Rosati. A Sound and Complete Tableau Calculus for Reasoning about only Knowing and Knowing at Most. *Studia Logica*, 69:171–191, 2001.

Raymond M. Smullyan. *First-Order Logic*, volume 43 of *Ergebnisse der Mathematik und ihrer Grenzgebiete*. Springer-Verlag, New York, 1968.

Bjørnar Solhaug. Logical Spaces in Multi-Modal Only Knowing Logics. Master's thesis, Dept. of Linguistics, University of Oslo, 2004.

Arild Waaler. Connections in Nonclassical Logics. In Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, volume II, chapter 22, pages 1487–1587. Elsevier Science, 2001.

Arild Waaler. Consistency proosf for systems of multi-agent only knowing. *Advances in Modal Logic*, 5, 2005.

Arild Waaler. *Logical studies in Complementary Weak S5*. PhD thesis, University of Oslo, 1994.

Arild Waaler, Johan W. Klüver, Tore Langholm, and Espen H. Lian. Only Knowing with Degrees of Confidence. To appear in Journal of Applied Logic, 2004.

Michal Walicki. Introduction to Logic. `http://www.ii.uib.no/~michal/und/i227/book/book.pdf`, 2006. Lecture notes IN-227, University of Bergen.

Lincoln A. Wallen. *Automated Deduction in Nonclassical Logics*. MIT Press, 1990.

List of Definitions

List of Examples

List of Figures