A Lattice-Theoretic Framework For Circular Assume-Guarantee Reasoning

Dissertation

zur Erlangung des Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
der Naturwissenschaftlich-Technischen Fakultät I
der Universität des Saarlandes
von

Patrick Maier

Saarbrücken
2003

Tag des Kolloquiums:	23. Juli 2003
Dekan:	Prof. Dr.-Ing. Philipp Slusallek
Berichterstatter:	Prof. Dr. Harald Ganzinger
	Prof. Dr. Andreas Podelski
	Sriram K. Rajamani, Ph.D.

Kurzzusammenfassung

Wir entwickeln einen abstrakten verbandstheoretischen Rahmen in dem wir die Korrektheit und andere Eigenschaften bedingter zirkulärer Assume-Guarantee-Regeln (A-G-Regeln) untersuchen. Wir isolieren eine besondere Nebenbedingung, *non-blockingness*, die zu einem verständlichen induktiven Beweis der Korrektheit zirkulärer A-G-Regeln führt. Ausserdem sind durch non-blockingness eingeschränkte zirkuläre Regeln vollständig und stärker als eine grosse Klasse von korrekten bedingten A-G-Regeln. So gesehen erhellt unsere Arbeit die Grundlagen des zirkulären A-G-Paradigmas.

Aufgrund seiner Abstraktheit kann unser Rahmen zu vielen konkreten Formalismen instanziiert werden. Wir zeigen, dass mehrere bekannte A-G-Regeln zur kompositionalen Verifikation Instanzen unserer generischen Regeln sind. So ist der zirkularitätsauflösende Beweis der Korrektheit nur einmal für unsere generische Regeln zu führen, dann erben alle Instanzen Korrektheit, ohne dass noch einmal ein zirkularitätsauflösender Beweis nötig ist. In dieser Hinsicht stellt unser Rahmen eine einheitliche Plattform dar, die verschiedene Ausformungen des zirkulären A-G-Paradigmas umfasst und von der ausgehend systematisch neue zirkuläre A-G-Regeln entwickelt werden können.

Abstract

We develop an abstract lattice-theoretic framework within which we study soundness and other properties of circular assume-guarantee (A-G) rules constrained by side conditions. We identify a particular side condition, *non-blockingness*, which admits an intelligible inductive proof of the soundness of circular A-G reasoning. Besides, conditional circular rules based on non-blockingness turn out to be complete in various senses and stronger than a large class of sound conditional A-G rules. In this respect, our framework enlightens the foundations of circular A-G reasoning.

Due to its abstractness, the framework can be instantiated to many concrete settings. We show several known circular A-G rules for compositional verification to be instances of our generic rules. Thus, we do the circularity-breaking inductive argument once to establish soundness of our generic rules, which then implies soundness of all the instances without resorting to technically complicated circularity-breaking arguments for each single rule. In this respect, our framework unifies many approaches to circular A-G reasoning and provides a starting point for the systematic development of new circular A-G rules.

Acknowledgments

First and foremost I would like to thank Prof. Dr. Harald Ganzinger for accepting me as a PhD student in his group at the Max-Planck-Institut für Informatik. Even more I owe him for his continued support and interest in my work, which showed up the form of brilliant questions. There is no answer without a question — many of the ideas in this thesis emerged while thinking about Harald's questions.

I was most fortunate to have Prof. Dr. Andreas Podelski as my advisor. He encouraged my interest in circular assume-guarantee reasoning. I thank him for many fruitful discussions, for his patience and for teaching me simple and effective rules how to write.

I am grateful to Dr. Sriram Rajamani for his instant commitment to become a referee of this thesis.

I am indebted to Dr. Friedrich Eisenbrand, Prof. Dr. Andreas Podelski and Dr. Emil Weydert for reading drafts of the introduction. I owe immense gratitude to Dr. Viorica Sofronie-Stokkermans and Dr. Uwe Waldmann for reading drafts of the thesis in great detail and providing me with valuable comments and corrections.

Over the years there were a number of colleagues who were always willing to discuss a problem. The ones I recall are Werner Backes, Dr. Witold Charatonik, Dr. Giorgio Delzanno, Dr. Friedrich Eisenbrand, Lilia Georgieva, Thomas Hillenbrand, Dr. Manfred Jaeger, Dr. Hans de Nivelle, Andrey Rybalchenko, Dr. Viorica Sofronie-Stokkermans, Dr. Jürgen Stuber, Dr. Jean-Marc Talbot, Dr. Uwe Waldmann and Dr. Emil Weydert. Apologies to those I have forgotten.

Special thanks go to Lilia for providing emotional support in hard times.

Contents

Zusammenfassung

In dieser Arbeit untersuchen wir die Grundlagen für die Korrektheit scheinbar zirkulärer Schlussregeln, der so genannten zirkulären *Assume-Guarantee*-Regeln, wie sie zur kompositionalen Verifikation nebenläufiger Systeme verwendet werden.

Kompositionale Verifikation [Pnu85, CLM89, GL94, Kur94] nutzt die hierarchische Struktur komplexer Systeme und Spezifikationen, um ein Verifikationsproblem in Teilprobleme aufzuteilen. Sind die Teilprobleme gelöst, kann mit einer Schlussregel die Lösung des Gesamtproblems geschlossen werden. Zwei solcher Schlussregeln sind beispielsweise die folgenden:

$$\frac{s_1 \sqsubseteq p_1 \qquad p_1 \sqcap s_2 \sqsubseteq p_2}{s_1 \sqcap s_2 \sqsubseteq p_1 \sqcap p_2} \qquad\qquad \frac{p_2 \sqcap s_1 \sqsubseteq p_1 \qquad p_1 \sqcap s_2 \sqsubseteq p_2}{s_1 \sqcap s_2 \sqsubseteq p_1 \sqcap p_2}$$

Hier stellen die s_i Systeme und die p_j Spezifikationen dar. Die Konklusion beider Regeln formuliert das Verifikationsproblem, dass die parallele Komposition $s_1 \sqcap s_2$ der beiden Systeme die Konjunktion $p_1 \sqcap p_2$ der beiden Spezifikationen erfüllen soll. Die Prämissen der linken Regel lesen wir als die Teilprobleme, s_1 *erfüllt* p_1 und *angenommen* p_1 *ist erfüllt, dann garantiert (das heisst, erfüllt)* s_2 p_2; der zweiten Prämisse wegen nennen wir solche Regeln *Assume-Guarantee*-Regeln (oder kurz A-G-Regeln).

Die rechte Regel unterscheidet sich von der linken nur in der ersten Prämisse, welche besagt, dass s_1 p_1 garantiert, angenommen p_2 ist erfüllt. Interpretieren wir die beiden Regeln jedoch in einem natürlichen algebraischen Rahmen, nämlich einem Halbverband, wobei $\sqsubseteq$ die Ordnung und $\sqcap$ der zweistellige Infimumoperator ist, so folgt die Konklusion der linken Regel nach den Rechenregeln für Halbverbände aus den Prämissen, während dies für die rechte Regel nicht gilt — aufgrund der Zirkularität in ihren Prämissen ist sie inkorrekt. Das heisst, wir können aus der erfolgreichen Verifikation der Prämissen der rechten Regel nicht schliessen, dass das Verifikationsproblem gelöst ist.

Wie Fallstudien belegen [McM98, HQR98, HLQR99, HQR00, TB97], kommen bei der Verifikation nebenläufiger Systeme Situationen, in denen es natürlich wäre, eine zirkuläre A-G-Regel anzuwenden — wenn sie denn korrekt wäre — nicht eben selten vor. Die Ursache dafür ist die hohe Symmetrie nebenläufiger Systeme, die häufig aus vielen gleichartigen Subsystemen aufgebaut sind. Zirkuläre A-G-Regeln würden es erlauben, diese Symmetrie auf eine natürliche Weise für die kompositionale Verifikation zu nutzen.

Eine Möglichkeit, die Inkorrektheit zirkulärer A-G-Regeln zu heilen, ist ihre Einschränkung durch eine Nebenbedinung, welche die Zirkularität auflöst. Beispielsweise könnten wir obige inkorrekte zirkuläre Regel durch folgende bedingte Regel ersetzen:

$$\frac{p_2 \sqcap s_1 \sqsubseteq p_1 \qquad p_1 \sqcap s_2 \sqsubseteq p_2}{s_1 \sqcap s_2 \sqsubseteq p_1 \sqcap p_2} \text{ if } \Gamma[s_1, s_2, p_1, p_2]$$

Die Nebenbedingung $\Gamma[s_1, s_2, p_1, p_2]$ können wir als Orakel auffassen, das uns für jedes Quadrupel von Systemen s_i und Spezifikationen p_j sagt, ob wir die Regel anwenden dürfen.

Wir präsentieren in dieser Arbeit eine Klasse von vollständigen Verbänden, in welchen wir eine einfache abstrakte Nebenbedingung — *Non-Blockingness* genannt — identifizieren, die zirkuläre A-G-Regeln wie die obige korrekt macht. Die Abstraktheit der Nebenbedingung ermöglicht einen einfachen und direkten induktiven Korrektheitsbeweis zirkulärer A-G-Regeln. Darüberhinaus ist Non-Blockingness symmetrisch, so dass die attraktive Symmetrie zirkulärer A-G-Regeln erhalten bleibt.

Es stellt sich heraus, dass derart bedingte zirkuläre A-G-Regeln weitere vorteilhafte Eigenschaften haben. Zum einen sind solche Regeln sowohl vorwärts als auch rückwärts vollständig. Dabei entspricht Rückwärtsvollständigkeit der üblichen Vollständigkeit logischer Kalküle, welche in unserem Kontext besagt, dass in allen Fällen, in denen das Gesamtsystem die Gesamtspezifikation erfüllt, dies mit Hilfe der rückwärtsvollständigen A-G-Regel geschlossen werden kann. Dagegen sagt Vorwärtsvollständigkeit, dass in allen Fällen, in denen aus der Lösung der Teilprobleme tatsächlich die Lösung des Gesamtproblems folgt, in denen also kein inkorrekter Zirkelschluss vorliegt, die Nebenbedingung die Anwendung der A-G-Regel erlaubt, so dass die Lösung des Gesamtproblems geschlossen werden kann.

Zum anderen sind unsere zirkulären A-G-Regeln mit Non-Blockingness-Nebenbedingungen die stärksten Regeln in der Klasse der voll korrekten A-G-Regeln, einer wichtigen Unterklasse aller korrekten bedingten A-G-Regeln. Das bedeutet, dass sich diese stärksten A-G-Regeln auf jedes Verifikationsproblem anwenden lassen, auf das irgendeine Regel aus der genannten Unterklasse anwendbar ist.

Im letzten Teil der Arbeit zeigen wir ausführlich anhand von zwei verschiedenen Beispielen, wie sich unser abstrakter verbandstheoretischer Rahmen instanziieren lässt zu konkreten Formalismen, die — im Gegensatz zur allgemeinen Verbandstheorie — zu Modellierung und Verifikation nebenläufiger Systeme tatsächlich verwendet werden. Auf diese Weise ergeben sich einige der in der Literatur bekannte A-G-Regeln als Instanzen unserer verbandstheoretischen Regeln. Da die Instanzen die einmal bewiesene Korrektheit der verbandstheoretischen Regeln erben, werden technisch komplizierte induktive Korrektheitsbeweise im jeweiligen konkreten Modellierungsformalismus überflüssig. In dieser Hinsicht

bietet unser verbandstheoretischer Rahmen eine einheitliche Plattform, die verschiedene Ausformungen des zirkulären Assume-Guarantee-Paradigmas umfasst und von der ausgehend systematisch neue zirkuläre Assume-Guarantee-Regeln entwickelt werden können.

Extended Abstract

In this thesis we study the foundations of the soundness of apparently circular inference rules, the so-called *assume-guarantee* rules that are applied to compositional verification of concurrent systems.

Compositional verification [Pnu85, CLM89, GL94, Kur94] exploits the hierarchical structure of complex systems and specifications to divide a verification task into subtasks. Once all subtasks are solved, one may infer by a suitable inference rule that the general task is solved. For instance, two such inference rules are the following:

$$\frac{s_1 \sqsubseteq p_1 \qquad p_1 \sqcap s_2 \sqsubseteq p_2}{s_1 \sqcap s_2 \sqsubseteq p_1 \sqcap p_2} \qquad\qquad \frac{p_2 \sqcap s_1 \sqsubseteq p_1 \qquad p_1 \sqcap s_2 \sqsubseteq p_2}{s_1 \sqcap s_2 \sqsubseteq p_1 \sqcap p_2}$$

Here, the s_i and p_j represent systems and specifications, respectively. The conclusions formulate the general verification task, which is to establish that the parallel composition $s_1 \sqcap s_2$ of both systems satisfies the conjunction $p_1 \sqcap p_2$ of both specifications. The premises of the left rule are to be read as the verification subtasks, which are s_1 *satisfies* p_1 and *assuming* p_1, s_2 *guarantees (i. e., satisfies)* p_2; thanks to this reading of the second premise we call such inference rules *assume-guarantee* rules (A-G rules for short).

The rule to the right only differs from the one to the left in the first premise, which now states that s_1 guarantees p_1 provided that p_2 is assumed. Yet, when we interpret both rules in a natural algebraic framework, namely in a meet-semilattice, where $\sqsubseteq$ denotes the order relation and $\sqcap$ the binary infimum operator, then they exhibit a big difference. For the rule to the left, the conclusion logically follows from the premises by the laws for meet-semilattices. For the rule to the right, however, this is not the case — it is unsound due to the circularity in its premises. Therefore, after having established the verification subtasks corresponding to the premises of the right rule, we cannot infer that the general verification task is solved.

A number of case studies [McM98, HQR98, HLQR99, HQR00, TB97] show that situations in which circular A-G rules were naturally applicable — provided they were sound — arise frequently in the verification of concurrent systems. This is caused by the abundant symmetries of concurrent systems, which frequently consist of a large number of similar subsystems. Circular A-G rules would admit to exploit these symmetries naturally for compositional verification.

One way to heal the unsoundness of circular A-G rules is to restrict them by circularity-breaking side conditions. For example, we might replace the above unsound circular rule with the following conditional rule:

$$\frac{p_2 \sqcap s_1 \sqsubseteq p_1 \qquad p_1 \sqcap s_2 \sqsubseteq p_2}{s_1 \sqcap s_2 \sqsubseteq p_1 \sqcap p_2} \text{ if } \Gamma[s_1, s_2, p_1, p_2]$$

The side condition $\Gamma[s_1, s_2, p_1, p_2]$ may be perceived as an oracle which decides, for every quadruple of systems s_i and specifications p_j, whether it is admissible to apply the rule or not.

In this thesis we present a lattice-theoretic framework within which we identify a class of complete lattices — the *well-approximable* lattices — and a simple abstract side condition — *non-blockingness* — which ensures soundness of circular A-G rules like the one above. Because non-blockingness is so abstract, soundness of circular A-G rules is provable by a straightforward inductive argument. Furthermore, non-blockingness is symmetric and thus retains the attractive symmetry of circular A-G rules.

It turns out that conditional A-G rules based on non-blockingness possess further desirable properties. First, they are both forward and backward complete. Thereby, backward completeness corresponds to the usual completeness of logical calculi, which in our context means that whenever a complex system actually satisfies a complex specification then this fact is derivable via the backward complete A-G rule. Forward completeness, however, means that whenever the results of the verification subtasks actually imply that the complex system satisfies the complex specification, i. e., whenever there is not really an unsound circular argument involved, then the side condition admits application of the A-G rule so as to infer the fact that the general verification task is solved.

Second, our circular A-G rules based on non-blockingness are the strongest rules among the fully sound A-G rules, which form a substantial subclass of the class of sound conditional A-G rules. As a consequence, these strongest A-G rules are applicable to any verification task to which some fully sound A-G rule is applicable.

The last part of the thesis is concerned with instantiating our abstract lattice-theoretic framework to concrete formalisms that — contrary to lattice theory — are used in practice for modeling and verifying concurrent systems. On two rather different examples, we demonstrate the technique of deriving A-G rules for particular settings as instances of our generic lattice-theoretic rules. Thus, we do the circularity-breaking inductive argument once to establish soundness of our generic rules, which then implies soundness of all the instances without resorting to technically complicated circularity-breaking arguments for each single rule. In this respect, our framework unifies many approaches to circular assume-guarantee reasoning and provides a starting point for the systematic development of new circular assume-guarantee rules.

Chapter 1

Introduction

> Noli turbare circulos meos.
> *Archimedes*

The central motivation of this thesis is to study the foundations of apparently circular patterns of reasoning that are well-known in the realm of compositional verification as the *assume-guarantee paradigm*. We propose an abstract, lattice-theoretic framework in which assume-guarantee reasoning can be modeled as conditional inference rules, i. e., inference rules constrained by side conditions. In this framework, we identify the key concepts of the assume-guarantee paradigm and give an intelligible inductive proof of its soundness. By means of instantiating the abstract framework to concrete settings, this soundness result carries over, i. e., the soundness of well-known circular reasoning patterns in verification can be derived. Besides soundness, the abstract framework opens the way to investigate other proof-theoretic questions about the assume-guarantee paradigm, such as completeness and the relative strength of different assume-guarantee rules.

1.1 Motivation

Verification. In verification, the objective is to establish that a given system conforms to a given specification. How that can be done depends on how systems and specifications are presented. For example, if the system is a finite labeled graph and the specification a formula in some propositional temporal logic then conformance corresponds to the satisfaction relation of the logic, which can usually be established via model checking [QS82, LP85, CES86]. In this case, the class of systems (finite labeled graphs) is distinct from the class of specifications (propositional temporal formulas).

It is quite common in verification, however, to present systems and specifications uniformly. For instance, in the automata-theoretic approach to model checking [VW86], both are presented as finite automata of some kind and conformance of the system to the specification corresponds to language containment

of automata. In simulation-based approaches [Mil71], systems and specifications are presented as labeled graphs, and conformance corresponds to simulation of graphs. And finally in theorem proving, systems and specifications are presented as formulas in some logic, e. g., propositional or first order logic (possibly with modalities) [MP95] or even higher order logic [Mel93], and conformance corresponds to entailment of formulas. In all these approaches, conformance is a reflexive and transitive binary relation, i. e., a pre-order on the class of systems and specifications.

Composition. Real-world systems are of high complexity. Often this complexity is unmanageable by a single human being, therefore it is good engineering practice to build up complex systems from simpler subsystems, which in turn may be built up from even simpler subsystems, and so on. The same applies to complex specifications, which are usually composed from simpler sub-specifications in a similar hierarchical manner.

The complexity of real-world systems often makes verification infeasible. Particularly in model checking of concurrent systems, this problem is considered so serious that it is referred to by the intimidating name *state explosion problem* [Val98]. It arises because the parallel composition of two subsystems results in a system whose size (as a transition graph) often is proportional to the the product of the sizes of the subsystems. Intuitively speaking, the size of a system tends to grow exponentially in the number of its subsystems.

Compositional verification[1] [Pnu85, CLM89, GL94, Kur94] is one possible way to fight complexity. It exploits the hierarchical structure of systems and specifications through a divide-and-conquer approach: Separately for each subsystem, one establishes conformance to parts of the specification. Having established these verification subtasks, either directly or again via compositional verification techniques, one deduces that the full system conforms to the full specification by a suitable proof rule.

Besides alleviating the complexity of verification, there are other methodological advantages of compositional approaches. One is separation of concerns: Different verification subtasks may fall into different categories, e. g., finite state versus infinite state problems, and for each category one may use specialized methods. Another advantage is reuse: Subsystems that are already known to be verified need not be verified again (provided that their specifications have not changed).

Framework. Above, we have argued that it is common to deal with systems and specifications uniformly as elements of a pre-ordered set, where the pre-order is naturally induced by the conformance relation. In order to fit composition into the picture, we must extend the pre-ordered set with a binary operation, whose

[1] Sometimes also called modular verification.

properties depend on the type of composition one intends to model. Actually, any reasonable type of composition should at least respect the conformance pre-order, i. e., it should be monotonous in both arguments. In fact, we may assume systems and specifications to form an ordered set (ordered by conformance) with a monotonous binary operation (composition); this can always be achieved by quotienting with the equivalence induced by the pre-order.

In our framework, we model the set of systems and specifications as a meet-semilattice, i. e., as an ordered set in which every pair of elements has a greatest lower bound, called meet. Throughout this section, we denote the order by the symbol $\sqsubseteq$, and by $x \sqcap y$ we denote the meet of the elements x and y. We may interpret the order in three different ways, as conformance, refinement or entailment.

- For a system s and a specification p, $s \sqsubseteq p$ means that s conforms to p.
- For two systems s and s', $s \sqsubseteq s'$ means that s refines s'.
- For two specifications p and p', $p \sqsubseteq p'$ means that p entails p'.

There are also three different readings for the meet, namely composition of systems, composition of specifications or constraining a system to conform to a specification.

- For two systems s_1 and s_2, $s_1 \sqcap s_2$ is the system which arises from composing s_1 and s_2.
- For two specifications p_1 and p_2, $p_1 \sqcap p_2$ is the specification which arises from composing p_1 and p_2.
- For a system s and a specification p, $p \sqcap s$ is the system which arises from constraining s such that it conforms to p; more precisely, $p \sqcap s$ is the coarsest refinement of s which conforms to p.

Note that our model abstracts many formalisms for compositional verification of concurrent systems. All we demand from composition is it to be an associative, commutative and idempotent operation. Actually, associativity and commutativity are inherent to parallel composition. Idempotency is naturally present in many formalisms, for instance when parallel composition amounts to conjunction of logical formulas as in deductive formalisms, or to intersection of languages as in formalisms based on automata-theoretic model checking. Even formalisms based on Moore or Mealy machines, where parallel composition is a partial operation, fit in the framework because we can extend composition conservatively to an associative, commutative and idempotent operation (by identifying it with language intersection). It should be noted, however, that the framework is not suited for all possible formalisms. For instance, the parallel composition operator in process

algebras like CCS or the π-calculus is associative and commutative but inherently non-idempotent. Also, the sequential composition operator in imperative programming languages is associative but neither idempotent nor commutative.

Proof rules. Above, we have explained that the backbone of compositional verification (the conquer step, so to speak) is a proof rule for deducing conformance of the system to the specification from the results of the verification subtasks. In the following, we will present a few examples of such proof rules that appear commonly in compositional verification of concurrent systems. For instance, one of the rules most often used is (1.1).

$$\frac{s_1 \sqsubseteq p_1 \qquad s_2 \sqsubseteq p_2}{s_1 \sqcap s_2 \sqsubseteq p_1 \sqcap p_2} \tag{1.1}$$

Here, the s_i and p_j are variables representing (sub-)systems and (sub-)specifications, respectively. The conclusion of the rule expresses the goal of compositional verification, namely that the composition of both systems conforms to (the composition of) both specifications. The premises state that each s_i conforms to p_i. Given that we establish these verification subtasks, we may complete our verification task by applying (1.1).

What can we do if we are not able to establish the premises of such a simple rule? Assume for the moment that we can actually establish the first premise of (1.1), i. e., s_1 conforms to p_1, but not the second. Because the system s_2 has been designed to cooperate with s_1, it may actually only conform to its specification p_2 if it interacts with an environment that behaves similar to s_1, i. e., if its environment conforms to some specification of s_1, say p_1. So as a verification subtask, we would have to establish that s_2 conforms to p_2 under the assumption p_1 for the environment. Expressed in a more succinct phrase, this reads as "assuming p_1, s_2 guarantees p_2" — hence the wide-spread term *assume-guarantee rule* for rules like (1.2).

$$\frac{s_1 \sqsubseteq p_1 \qquad p_1 \sqcap s_2 \sqsubseteq p_2}{s_1 \sqcap s_2 \sqsubseteq p_1 \sqcap p_2} \tag{1.2}$$

Here, we use the fact that systems and properties are treated uniformly, so we can compose p_1 with s_2, i. e., constrain s_2 such that it conforms to p_1.

What if we are able to establish the second premise of rule (1.2) but not the first? After all, the system s_1 may have been designed to cooperate with s_2, so it might require the assumption p_2 in order to guarantee p_1. In this case, we would need a rule like (1.3).

$$\frac{p_2 \sqcap s_1 \sqsubseteq p_1 \qquad p_1 \sqcap s_2 \sqsubseteq p_2}{s_1 \sqcap s_2 \sqsubseteq p_1 \sqcap p_2} \tag{1.3}$$

In both verification subtasks, the guaranteed property relies on the other one as an assumption, hence such rules are named *circular assume-guarantee rules*.

There is a big difference between (1.1) and (1.2) on one hand and (1.3) on the other hand, namely soundness. The rules (1.1) and (1.2) are sound, i. e., the conjunction of their premises logically implies their conclusion by the laws for meet-semilattices. For the circular rule (1.3), however, this is not the case; in fact, in every non-trivial meet-semilattice there exist counter-examples to soundness, i. e., values for the variables s_i and p_j such that $p_2 \sqcap s_1 \sqsubseteq p_1$ and $p_1 \sqcap s_2 \sqsubseteq p_2$ are true but $s_1 \sqcap s_2 \sqsubseteq p_1 \sqcap p_2$ is not. Hence, (1.3) may not be used for compositional verification.

Still, such circular rules are desirable for several reasons. They naturally generalize non-circular assume-guarantee rules like (1.2), and their symmetry naturally reflects structural symmetries in the design of complex systems. This raises the question whether unsound circular rules can be modified so that they become sound. One way to do this is by attaching a circularity-breaking side condition, which leads to a *conditional rule* like (1.4).

$$\frac{p_2 \sqcap s_1 \sqsubseteq p_1 \qquad p_1 \sqcap s_2 \sqsubseteq p_2}{s_1 \sqcap s_2 \sqsubseteq p_1 \sqcap p_2} \text{ if } \Gamma[s_1, s_2, p_1, p_2] \tag{1.4}$$

Here, $\Gamma[s_1, s_2, p_1, p_2]$ is a quaternary relation on the meet-semilattice of systems and specifications, which constrains the application of the rule to specific quadruples of values for the s_i and p_j. If every quadruple of values that satisfies $\Gamma[s_1, s_2, p_1, p_2]$ either validates the conclusion of (1.4) or falsifies one of the premises then the conditional proof rule (1.4) is sound, i. e., cannot deduce false conclusions. In this case, we may view $\Gamma[s_1, s_2, p_1, p_2]$ as an oracle which tells us when it is safe to apply the rule.

In general, there exist many side conditions which ensure soundness of the rule (1.4), but not all conditions are reasonable. For instance, the side condition which is always false yields a sound rule which is never applicable and thus useless. This raises the question which side conditions are reasonable. We identify the following list of criteria according to which one might judge the quality of a circularity-breaking side condition $\Gamma[s_1, s_2, p_1, p_2]$:

- **Intelligibility.** $\Gamma[s_1, s_2, p_1, p_2]$ should be intelligible and admit an intelligible argument justifying soundness of (1.4).
- **Symmetry.** $\Gamma[s_1, s_2, p_1, p_2]$ should retain the symmetry of (1.4).
- **Completeness.** With $\Gamma[s_1, s_2, p_1, p_2]$, the conditional rule (1.4) should be complete.
- **Generality.** $\Gamma[s_1, s_2, p_1, p_2]$ should be as general as possible.

The basic goal of this thesis is to find circularity-breaking side conditions (or more general, sound conditional assume-guarantee rules) that meet all four of the above criteria. In the next section, we briefly outline to what extent we have achieved this goal. Thereby, we also clarify the last two criteria, completeness and generality, whose meaning is rather fuzzy as yet.

1.2 Outline

Chapter 2. Here, we briefly review basic notions from the theory of ordered sets and lattices and introduce notation which we will use throughout the thesis.

Chapter 3. This chapter lays the foundations for investigating conditional proof rules. We formally define the concept of conditional inference rules in a meet-semilattice, which will be refined to the concept of (conditional) assume-guarantee rules at the end of the chapter.

Towards the criterion of generality, we develop a stronger-than order on conditional inference rules which captures the intuition that a stronger rule can mimic a weaker one. More precisely, the stronger rule is applicable in all (and possibly more) situations where the weaker one is, and the stronger rule derives a stronger conclusion from weaker premises.

Furthermore, we formally define soundness of conditional inference rules. It turns out that there are several reasonable ways to 'reverse' soundness, resulting in different notions of completeness. The most important of these are backward completeness and forward completeness. Roughly speaking, backward completeness ensures that all true consequences are derivable, i. e., whenever the conclusion of a backward complete rule is true then the premises and the side condition can be made true, so that the conclusion is derivable via the rule. On the other hand, forward completeness ensures that all consistent consequences are derivable, i. e., whenever the truth of the conclusion of a forward complete rule is consistent with the truth of the premises then the side condition admits application of the rule, so that the conclusion is derivable. To summarize, forward completeness judges whether the side condition of a rule is not overly restrictive, i. e., not more restrictive than necessary to ensure soundness. Thus, forward completeness determines which rules suit best for forward reasoning, whereas backward completeness determines which rules are desirable for backward reasoning.

Chapter 4. This chapter is the main contribution of the thesis, a lattice-theoretic framework in which we present a number of circular assume-guarantee rules that meet all of the quality criteria stated in the previous section, namely intelligible proofs of soundness, symmetry, completeness and — to some extent — generality.

To explain why we move from meet-semilattices to lattices, recall that the single most important property of a circular assume-guarantee rule is its soundness. Ever since the first publications on circular assume-guarantee reasoning [MC81, Jon81], soundness has been established via inductive arguments. This strongly suggests to use induction in our semilattice-based framework as well, which requires a notion of approximation on semilattice elements, and the approximants must be ordered in a well-founded way. A natural notion of approximation is obtained by embedding the meet-semilattice into a complete lattice,

which is always possible, e. g., via completion with order ideals. In the complete lattice, we identify a set of approximants which is join-dense, i. e., every lattice element is the least upper bound of a subset of the approximants. This leads us to the first key concept of our assume-guarantee framework, the definition of *well-approximable lattices*, i. e., lattices with a set of approximants which is join-dense and well-founded.

The second key concept is the *non-blockingness* relation, which is defined in terms of the order on the approximants and forms the circularity-breaking side conditions of our assume-guarantee rules. The name non-blockingness is somewhat reminiscent of conditions that ensure the absence of deadlocks in products of transition systems or automata.

In the framework of well-approximable lattices, we prove soundness of a basic circular assume-guarantee rule constrained by the non-blockingness relation. We extend this result to an infinite family of circular assume-guarantee rules, whose side conditions are conjunctions of non-blockingness constraints. The proofs of soundness are by a straightforward well-founded induction. Furthermore, all rules exhibit the same perfect symmetry as the rule (1.3) from the previous section, and all the rules are both forward and backward complete.

Concerning generality, we show that in well-approximable lattices, our assume-guarantee rules are the strongest (w. r. t. the stronger-than order) among the fully sound assume-guarantee rules, which form a substantial subclass of the class of sound assume-guarantee rules. Thus, the goal of generality, which is to cover all sound assume-guarantee rules, is not completely achieved. Another limitation to generality lies in our restriction to well-approximable lattices. At the end of Chapter 4, we have a brief look at lattices that lack this requirement. Although we cannot prove in general that the concept of well-approximable lattices is necessary for the soundness of circular assume-guarantee rules, we collect some evidence that it is often necessary for the soundness of rules based on non-blockingness.

Chapter 5. This chapter addresses generality from a different perspective. We show several circular assume-guarantee rules from the literature to be special instances of our rules. Thus, we do the circularity-breaking inductive argument once to establish soundness of our generic rules, which then implies soundness of all the instances without resorting to technically complicated circularity-breaking arguments for each single rule. In this respect, our framework unifies many approaches to circular assume-guarantee reasoning and provides a starting point for the systematic development of new circular assume-guarantee rules.

1.3 Related Work

This section briefly and abstractly reviews work by other authors which is related to the topic of this thesis, the foundations of circular assume-guarantee reasoning. A more detailed account on related work can be found in Chapter 5.

Historically, Misra and Chandy [MC81] and independently Jones [Jon81] were the first to propose and prove sound circular assume-guarantee patterns in the early 1980s. Their work focused on proof systems and deductive methods to establish safety properties of a class of concurrent systems. Since then, their work has been adapted and generalized to many modeling formalisms and specification languages for concurrent systems, see the book by de Roever et al. [dRdBH$^+$00] for a survey. Central to all this work is the distinction between a system and an environment which is reflected by assume-guarantee specifications, i. e., specifications that explicitly distinguish the system's assumptions about the environment from its guarantee towards that environment. Such assume-guarantee specifications have been investigated (in the context of various linear-time temporal logics) in depth by Abadi and others [AP93, AL93, AL95, AM95, JT96, Tsa00] in the 1990s. We discuss this approach to assume-guarantee reasoning in more detail at the end of Section 5.2.

Another approach to circular assume-guarantee reasoning arose in the 1990s in the realm of compositional model checking as a way to fight the state explosion problem. Systems and specifications are viewed as transition systems, and the goal is to establish a refinement relation between a system and its specification. The central observation in this field is that subsystems in isolation may not refine their corresponding sub-specifications, however, they may do so when they are composed with other sub-specifications; we have explained this phenomenon and the arising circularity at length in Section 1.1. Circular assume-guarantee rules for various modeling formalisms based on transition systems and various notions of refinement were studied by Alur, Henzinger, McMillan, and others [AH99, McM97, TAKB96, RR01, HQRT02]. We present this work in more detail at the end of Section 5.1.

Most of the work on circular assume-guarantee reasoning has been concerned with soundness only. Completeness has rarely been an issue, except for work on assume-guarantee proof systems for deductive verification, see [dRdBH$^+$00]. In the realm of compositional model checking, Namjoshi and Trefler [NT00] were the first to investigate completeness of circular assume-guarantee rules. They showed a number of circular rules to be backward incomplete and proposed generalizations which ensure backward completeness, at the expense of introducing auxiliary variables.

Not only is investigating soundness and completeness of circular reasoning patterns a challenging intellectual activity, but assume-guarantee reasoning does in fact play a role in the verification of complex systems. It has proven a valuable tool in the verification of real-world systems in a number of case studies [McM98,

HQR98, HLQR99, HQR00, TB97], mostly from the area of hardware verification. Assume-guarantee rules are also successfully applied in tools [FFQ02] for the verification of thread-parallel software, e. g., system code of distributed operating systems.

It should be noted, however, that assume-guarantee reasoning cannot save the world from state explosion. In [Var95, KV00], Kupferman and Vardi study the complexity of checking whether a system conforms to an assume-guarantee specification, i. e., the complexity of establishing a verification subtask. Depending on the temporal logic used for assumptions and guarantees, they show this problem to be PSPACE- or EXPSPACE-complete, even though checking, whether the system conforms to the guarantee without considering the assumptions, may be polynomial. Thus, it may happen that the exponential complexity of the full verification task, which arises from the size of the state space, is still retained in the verification subtasks, through the assumptions; in [KV00], this phenomenon is termed the *assumption-explosion problem*.

Originally, this work was stimulated by the question whether compositional verification techniques and in particular assume-guarantee reasoning could be combined with the symbolic model checking approach based on constraint logic programs (CLP) due to Delzanno and Podelski [DP99, Pod00, DP01]. However, this question is not pursued further here.

This thesis builds on our preliminary work about a set-theoretic framework for investigating the soundness of circular assume-guarantee reasoning [Mai01]. A sketch of the instantiation of the lattice-theoretic framework to assume-guarantee specifications (cf. Section 5.2) can be found in [Mai02].

Chapter 2

Preliminaries

This chapter reviews basic notions from the theory of ordered sets and lattices. Most of these definitions can be found in the textbook by Davey and Priestley [DP90], although our notation may depart from theirs occasionally. For the definitions of pseudo-complements and Heyting algebras, the reader is referred to the books by Johnstone [Joh82] or Goldblatt [Gol84]. At the end of this chapter, partial maps, natural numbers, words and trees are studied as particular examples of ordered sets, whereby we introduce notation that will be used frequently in later chapters.

2.1 Ordered Sets, Semilattices, Lattices, etc.

Ordered sets. An *ordered set* is a tuple $\langle X, \sqsubseteq \rangle$ where X is a set and $\sqsubseteq$ reflexive, transitive and antisymmetric binary relation on X. We call the elements of X *points* and the relation $\sqsubseteq$ an *order*[2] on X. If $\sqsubseteq$ is an order on X, we define the *corresponding strict order* $\sqsubset$ on X as the binary relation on X such that for all $x, y \in X$, $x \sqsubset y$ if and only if $x \sqsubseteq y$ and $x \neq y$. We say that $\langle X, \sqsubseteq \rangle$ is *linear* if for all $x, y \in X$, $x \sqsubseteq y$ or $y \sqsubseteq x$. We call $\langle X, \sqsubseteq \rangle$ *trivial* if X is empty or a singleton.

If $\langle X, \sqsubseteq \rangle$ is an ordered set, $Y \subseteq X$ and $\leq \; = \; \sqsubseteq \cap\, Y \times Y$ then $\langle Y, \leq \rangle$ is an ordered set, too, and we say that $\langle Y, \leq \rangle$ is a *suborder* of $\langle X, \sqsubseteq \rangle$. More precisely, we say that $\langle Y, \leq \rangle$ is the suborder of $\langle X, \sqsubseteq \rangle$ which is *generated* by Y and that $\leq$ is the order on Y which is *induced* by $\sqsubseteq$. To save order symbols, we may denote the induced order also by the symbol $\sqsubseteq$ if no confusion can arise.

We say that two ordered sets $\langle X, \sqsubseteq \rangle$ and $\langle Y, \leq \rangle$ are *isomorphic* if there exists a bijective map $\varphi : X \to Y$ such that for all $x, y \in X$, $x \sqsubseteq y$ if and only if $\varphi(x) \leq \varphi(y)$. The map φ is called an *isomorphism* from $\langle X, \sqsubseteq \rangle$ to $\langle Y, \leq \rangle$. An isomorphism from $\langle X, \sqsubseteq \rangle$ to a suborder of $\langle Y, \leq \rangle$ is called an *embedding* of $\langle X, \sqsubseteq \rangle$ into $\langle Y, \leq \rangle$.

[2] An order may also be called a *partial* order; and ordered sets are also known as *partially* ordered sets or *posets* for short.

Duality. Let $\langle X, \sqsubseteq \rangle$ be an ordered set. We define the binary relation $\sqsupseteq$ on X by for all $x, y \in X$, $x \sqsupseteq y$ if and only if $y \sqsubseteq x$. Then $\langle X, \sqsupseteq \rangle$ is an ordered set. The orders $\sqsubseteq$ and $\sqsupseteq$ are *dual* to each other in that each is the converse of the other. In the following, we will define notions only in terms of $\sqsubseteq$, and we will just briefly mention name and denotation of the corresponding dual notions, which arise from reversing the order symbol.

Bounds, extremal points, joins, and meets. Let $\langle X, \sqsubseteq \rangle$ be an ordered set and let $S \subseteq X$. We say that an element $x \in X$ is a *lower bound* of S if $x \sqsubseteq s$ for all $s \in S$. The dual to a lower bound of S is an *upper bound* of S.

An element $s \in S$ is called *minimal* in S if for all $t \in S$, $t \sqsubseteq s$ implies $t = s$. A *least element* of S is a point $s \in S$ such that $s \sqsubseteq t$ for all $t \in S$. Note that neither minimal nor least elements need exist, but if a least element exists then it is unique and minimal. The dual to a minimal element is a *maximal* element and the dual to a least element is a *greatest element*.

If the set of upper bounds of S has a least element then this element is called the *join* of S and denoted by $\bigsqcup S$. For $x, y \in X$, the *binary join* $x \sqcup y$ exists if the join of $\{x, y\}$ exists, in which case $x \sqcup y = \bigsqcup\{x, y\}$. By duality, we define $\bigsqcap S$, the *meet* of S, which is the greatest element of the set of lower bounds of S, and the *binary meet* of $x, y \in X$, denoted by $x \sqcap y$.

Notation: Families of symbols. So far, we have introduced the order $\sqsubseteq$ and its converse order $\sqsupseteq$, both with their corresponding strict orders $\sqsubset$ and $\sqsupset$. Furthermore, we have the join $\bigsqcup$ and the meet $\bigsqcap$, both with their binary versions $\sqcup$ and $\sqcap$. By their appearance, this whole family of rectangular symbols is associated with the order symbol $\sqsubseteq$. In order to distinguish operators that are associated with different orders, we will use different symbol families, e. g., the family of pointed symbols ($\leq$, $\geq$, $<$, $>$, $\bigvee$, $\bigwedge$, $\vee$, $\wedge$), the family of triangular symbols ($\trianglelefteq$, $\trianglerighteq$, $\triangleleft$, $\triangleright$, $\bigtriangledown$, $\bigtriangleup$, $\triangledown$, $\triangle$) or the family of curly symbols ($\preceq$, $\succeq$, $\prec$, $\succ$, $\curlyvee$, $\curlywedge$, $\curlyvee$, $\curlywedge$). The family of round symbols ($\subseteq$, $\supseteq$, $\subset$, $\supset$, $\bigcup$, $\bigcap$, $\cup$, $\cap$) is reserved for operations and relations on sets.

Semilattices. An ordered set $\langle X, \sqsubseteq \rangle$ is called a *join-semilattice* if all binary joins $x \sqcup y$ exist for $x, y \in X$. The dual to a join-semilattice is a *meet-semilattice*. We say that a join-semilattice $\langle Y, \leq \rangle$ is a *sub-join-semilattice* of another join-semilattice $\langle X, \sqsubseteq \rangle$ if $\langle Y, \leq \rangle$ is a suborder of $\langle X, \sqsubseteq \rangle$. We say that two join-semilattices $\langle X, \sqsubseteq \rangle$ and $\langle Y, \leq \rangle$ are *isomorphic* if there exists an isomorphism φ from $\langle X, \sqsubseteq \rangle$ to $\langle Y, \leq \rangle$. Note that although φ views $\langle X, \sqsubseteq \rangle$ and $\langle Y, \leq \rangle$ just as ordered sets, the term isomorphism is justified since φ preserves joins as well as their existence, i. e., for all $S \subseteq X$, the join of S exists if and only if the join of $\varphi(S)$ exists, in which case $\varphi(\bigsqcup S) = \bigvee \varphi(S)$.

Since in a join-semilattice $\langle X, \sqsubseteq \rangle$, $\sqcup$ is an idempotent associative and commutative binary operation, $\langle X, \sqcup \rangle$ is an idempotent Abelian semigroup. On the other hand, for every idempotent Abelian semigroup $\langle X, \circ \rangle$, we can define a binary relation $\sqsubseteq$ on X by for all $x, y \in X$, $x \sqsubseteq y$ if and only if $x \circ y = y$, and it turns out that $\langle X, \sqsubseteq \rangle$ is an ordered set in which the joins of all finite non-empty subsets of X exist and in which in fact $x \sqcup y = x \circ y$ for all $x, y \in X$. Consequently, every idempotent Abelian semigroup can be viewed as a join-semilattice, and vice versa. Furthermore, a least element in $\langle X, \sqsubseteq \rangle$ functions as neutral element in $\langle X, \sqcup \rangle$, and vice versa. Note that since a join-semilattice is a meet-semilattice for the converse order, every idempotent Abelian semigroup can also be viewed as a meet-semilattice.

If a join-semilattice $\langle X, \sqsubseteq \rangle$ has a least element then we say that $\langle X, \sqsubseteq \rangle$ is a *bottomed* join-semilattice. Note that all finite joins, i. e., all joins of finite sets, exist in a bottomed join-semilattice. Dually, all finite meets exist in a *topped* meet-semilattice. which is a meet-semilattice with greatest element. Usually, we denote the least element of a bottomed join-semilattice by $\bot$ or 0, whereas the greatest element of a topped meet-semilattice is denoted by $\top$ or 1.

Lattices. An ordered set $\langle X, \sqsubseteq \rangle$ is called a *lattice* if it is both a join-semilattice and a meet-semilattice. Note that a lattice satisfies the *absorption laws*, i. e., $x \sqcup (x \sqcap y) = x$ and $x \sqcap (x \sqcup y) = x$ for all $x, y \in X$. A lattice $\langle X, \sqsubseteq \rangle$ is *distributive* if it satisfies the *distributive laws*, i. e., $x \sqcup (y \sqcap z) = (x \sqcup y) \sqcap (x \sqcup z)$ and $x \sqcap (y \sqcup z) = (x \sqcap y) \sqcup (x \sqcap z)$ for all $x, y, z \in X$. A lattice $\langle X, \sqsubseteq \rangle$ is *bounded* if join and meet of the empty set exist. A lattice $\langle X, \sqsubseteq \rangle$ is *complete* if joins and meets of every subset of X exist. Note that a complete lattice is bounded, and a bounded lattice is both a bottomed join- and a topped meet-semilattice.

Let $\langle X, \sqsubseteq \rangle$ be a complete lattice. A subset $S \subseteq X$ is called *join-dense* in $\langle X, \sqsubseteq \rangle$ if for every $x \in X$ there is $T \subseteq S$ such that $x = \bigsqcup T$. An element $x \in X$ is *join-irreducible* in $\langle X, \sqsubseteq \rangle$ if $x \neq \bot$ and for all $a, b \in X$, $x = a \sqcup b$ implies $x = a$ or $x = b$. We call x *completely join-irreducible* in $\langle X, \sqsubseteq \rangle$ if for every subset $S \subseteq X$, $x = \bigsqcup S$ implies $x \in S$. Note that x being completely join-irreducible implies x being join-irreducible but not the other way round. By $\mathcal{J}(\langle X, \sqsubseteq \rangle)$, we denote the set of completely join-irreducible elements in $\langle X, \sqsubseteq \rangle$. Note that $\mathcal{J}(\langle X, \sqsubseteq \rangle) \subseteq S$ for every S which is join-dense in $\langle X, \sqsubseteq \rangle$.

Like in the case of semilattices, we say that a lattice $\langle Y, \leq \rangle$ is a *sub-lattice* of another lattice $\langle X, \sqsubseteq \rangle$ if $\langle Y, \leq \rangle$ is a suborder of $\langle X, \sqsubseteq \rangle$. And with the same arguments as for semilattices, it is justified to define two lattices $\langle X, \sqsubseteq \rangle$ and $\langle Y, \leq \rangle$ as *isomorphic* if there exists an isomorphism from $\langle X, \sqsubseteq \rangle$ to $\langle Y, \leq \rangle$. If there is an embedding from an ordered set $\langle X, \sqsubseteq \rangle$ into a complete lattice $\langle Y, \leq \rangle$ then $\langle Y, \leq \rangle$ is called a *completion* of $\langle X, \sqsubseteq \rangle$.

An example of a complete lattice is the two-element lattice $\langle \{0, 1\}, \leq \rangle$, which is unique up to isomorphism. It consists only of the least element 0 and the greatest

element 1, so join must be the maximum and meet the minimum operation on $\{0,1\}$. To give another example of a complete lattice, let A be an arbitrary set and let $\mathcal{P}(A)$ denote the *power set* of A, i. e., the family of all subsets of A. This family is ordered by set inclusion, and it is easy to see that join amounts to set union and meet to set intersection. Hence, $\langle \mathcal{P}(A), \subseteq \rangle$ is a complete lattice. On the other hand, consider $\mathcal{P}_{\text{fin}}(A)$, the family of all finite subsets of A, which is also ordered by inclusion. Thus, $\langle \mathcal{P}_{\text{fin}}(A), \subseteq \rangle$ is also a lattice — in fact, it is a proper sub-lattice of $\langle \mathcal{P}(A), \subseteq \rangle$ — where join amounts to union and meet to intersection. However, the union of infinitely many finite sets need not be finite, so $\langle \mathcal{P}_{\text{fin}}(A), \subseteq \rangle$ can't be complete. Nonetheless, both $\langle \mathcal{P}(A), \subseteq \rangle$ and $\langle \mathcal{P}_{\text{fin}}(A), \subseteq \rangle$ are distributive.

Pseudo-complements. Let $\langle X, \sqsubseteq \rangle$ be a lattice with least element $\bot$ and greatest element $\top$ and let $x, y \in X$. If $\{p \in X \mid x \sqcap p \sqsubseteq y\}$ has a greatest element then this element is called the *pseudo-complement* of x *relative* to y and denoted by $x \Rightarrow y$. We define the *pseudo-complement* of x, denoted by $\neg x$, as $x \Rightarrow \bot$. If $\neg x$ exists and $x \sqcup \neg x = \top$ then we say that $\neg x$ is called the *complement* of x. Note that if $\langle X, \sqsubseteq \rangle$ is distributive and $\neg x$ is the complement of x then $\neg x$ is the unique $z \in X$ with $x \sqcap z = \bot$ and $x \sqcup z = \top$.

Heyting and Boolean algebras. A bounded lattice $\langle X, \sqsubseteq \rangle$ is called a *Heyting algebra* if all relative pseudo-complements $x \Rightarrow y$ exist for $x, y \in X$. Note that every Heyting algebra is distributive. Furthermore in a Heyting algebra $\langle X, \sqsubseteq \rangle$ with greatest element $\top$, the equivalence $x \sqsubseteq y \Rightarrow z$ iff $y \sqcap x \sqsubseteq z$ and the equations

$$\begin{aligned} x \sqcap (x \Rightarrow y) &= x \sqcap y & \qquad x \Rightarrow x &= \top \\ y \sqcap (x \Rightarrow y) &= y & \qquad x \Rightarrow (y \sqcap z) &= (x \Rightarrow y) \sqcap (x \Rightarrow z) \end{aligned}$$

hold for all $x, y, z \in X$.

A Heyting algebra $\langle X, \sqsubseteq \rangle$ is called a *Boolean algebra* if $\neg x$ is the complement of x for all $x \in X$. Thus, a Boolean algebra $\langle X, \sqsubseteq \rangle$ is a Heyting algebra that satisfies the law of *excluded middle*, i. e., $x \sqcup \neg x = \top$ for all $x \in X$.

Notation: Typefaces. Given a set X, we may denote an ordered set (or semilattice, lattice, Heyting algebra, Boolean algebra) with point set X by typesetting the point set X in bold face, i. e., $\boldsymbol{X} = \langle X, \sqsubseteq \rangle$ for some order $\sqsubseteq$ on X. On the other hand, we may name an unknown ordered set (or semilattice, lattice, et cetera) by a letter typeset in bold face, in which case that letter typeset in normal face denotes the point set. E. g., if $\boldsymbol{Y}$ denotes an ordered set then Y denotes the point set of $\boldsymbol{Y}$.

Order ideals. Let $\boldsymbol{X} = \langle X, \sqsubseteq \rangle$ be an ordered set. A subset $S \subseteq X$ is called an *order ideal*[3] in $\boldsymbol{X}$ if for all $x, y \in X$, $x \sqsubseteq y$ and $y \in S$ implies $x \in S$. By $\mathcal{O}(\boldsymbol{X})$ we denote the family of all order ideals in $\boldsymbol{X}$. Note that $\langle \mathcal{O}(\boldsymbol{X}), \subseteq \rangle$ is a complete distributive lattice where join amounts to union and meet to intersection.

For every $x \in X$, we define the *downward closure* of x as $\{y \in X \mid y \sqsubseteq x\}$, denoted by $\sqsubseteq(x)$, and we define the *strict downward closure* of x as $\{y \in X \mid y \sqsubset x\}$, denoted by $\sqsubset(x)$. Note that $\sqsubseteq(x)$ and $\sqsubset(x)$ are order ideals in $\boldsymbol{X}$. In fact, $\sqsubseteq(x)$ is the least order ideal containing x; it is also called the *principal ideal*[4] generated by x. Obviously, the function which maps every $x \in X$ to $\sqsubseteq(x) \in \mathcal{O}(\boldsymbol{X})$ is an order embedding. In fact, via this embedding, $\langle \mathcal{O}(\boldsymbol{X}), \subseteq \rangle$ is a completion, the *ideal completion* of $\boldsymbol{X}$. Note that for every $S \subseteq X$, $\bigcup_{s \in S} \sqsubseteq(s) = \bigcap \{O \in \mathcal{O}(\boldsymbol{X}) \mid S \subseteq O\}$, i. e., the join of all principal ideals generated by elements in S is the least order ideal containing S. Therefore, we call $\bigcup_{s \in S} \sqsubseteq(s)$ the order ideal generated by S. Obviously, the set of principal ideals $\{\sqsubseteq(x) \mid x \in X\}$ is join-dense in the ideal completion of $\boldsymbol{X}$. Moreover, the principal ideals are the completely join-irreducible elements of this completion, i. e., $\mathcal{J}(\langle \mathcal{O}(\boldsymbol{X}), \subseteq \rangle) = \{\sqsubseteq(x) \mid x \in X\}$.

We say that $\boldsymbol{X}$ is *forest-like* if for all $x \in X$, the suborder of $\boldsymbol{X}$ generated by $\sqsubseteq(x)$ is linear. We say that $\boldsymbol{X}$ is *tree-like* if it is forest-like and there is a least element of X.

Chain conditions. An ordered set $\boldsymbol{X} = \langle X, \sqsubseteq \rangle$ satisfies the *ascending chain condition*, (ACC), if every infinite ascending sequence eventually stabilizes, i. e., if for every infinite sequence $x_0 \sqsubseteq x_1 \sqsubseteq x_2 \sqsubseteq \ldots$ of elements in X there is an index $i \in \mathbb{N}$ such that $x_i = x_j$ for all $j > i$. By duality, $\boldsymbol{X}$ satisfies the *descending chain condition*,, (DCC), if every infinite descending sequence eventually stabilizes. Note that an ordered set $\boldsymbol{X}$ satisfies ACC if and only if every non-empty subset of X has a maximal element. Dually, $\boldsymbol{X}$ satisfies DCC if and only if every non-empty subset of X has a minimal element.

Given an ordered set $\boldsymbol{X} = \langle X, \sqsubseteq \rangle$ satisfying DCC, one can establish the fact that for every subset $P \subseteq X$, $P = X$ if for all $x \in X$, $\sqsubset(x) \subseteq P$ implies $x \in P$. This is known as the principle of *well-founded induction* (over $\boldsymbol{X}$), and a more common way to state it is: For every property P over X, if for all $x \in X$, the fact that all $y \in \sqsubset(x)$ have P implies that x has P, then all $x \in X$ have P.

2.2 Some Examples of Ordered Sets

Partial maps. Let A and B be sets. We denote the set of *partial maps* from A to B by $A \rightharpoonup B$. For every $f \in A \rightharpoonup B$, we denote the *domain* of f by $dom(f)$.

[3] Order ideals are also known as *downward-closed sets*, or *down-sets* for short.

[4] Note that for $x \in X$, $\sqsubseteq(x)$ is an *ideal*, i. e., an order ideal which is closed under join.

We define a binary relation $\trianglelefteq$ on $A \rightharpoonup B$ by $f \trianglelefteq g$ if and only if $dom(f) \subseteq dom(g)$ and $f(a) = g(a)$ for all $a \in dom(f)$. Then $\langle A \rightharpoonup B, \trianglelefteq \rangle$ is an ordered set with least element, which is denoted by $\bot$. The set of maximal elements in $\langle A \rightharpoonup B, \trianglelefteq \rangle$, i. e., the set of *total maps* from A to B, is denoted by $A \to B$. Given $f, g \in A \rightharpoonup B$, note that the join $f \triangledown g$ of f and g exists if and only if the set of upper bounds of $\{f, g\}$ is non-empty. Finally, given $f \in A \rightharpoonup B$ and $X \subseteq A$, we define the *restriction* of f to X as the partial map $f|_X \in A \rightharpoonup B$ such that $f|_X \trianglelefteq f$ and $dom(f|_X) = dom(f) \cap X$. We extend restriction to sets of partial maps in the canonical way, i. e., $F|_X = \{f|_X \mid f \in F\}$ for every $F \subseteq A \rightharpoonup B$ and $X \subseteq A$.

Let A and B be sets and let $\sqsubseteq$ be an order on B. Let $\leq$ be the binary relation on $A \rightharpoonup B$ such that for all $f, g \in A \rightharpoonup B$, $f \leq g$ if and only if $dom(f) \subseteq dom(g)$ and $f(a) \sqsubseteq g(a)$ for all $a \in dom(f)$. Then $\leq$ is an order on $A \rightharpoonup B$; we say that $\sqsubseteq$ has been *lifted* or more precisely, the order $\leq$ is called the *lifting* of $\sqsubseteq$ to $A \rightharpoonup B$. Note that $\trianglelefteq$ may be viewed as the lifting of the identity on B to $A \rightharpoonup B$.

Let A, B, C and D be sets. Given $f \in A \rightharpoonup B$, we define the *range* of f, denoted by $rng(f)$, as the set $\{f(a) \mid a \in dom(f)\}$. Given $f \in A \rightharpoonup B$ and $g \in B \rightharpoonup C$, we define the *composition* of g and f as the partial map $g \circ f \in A \rightharpoonup C$ such that $dom(g \circ f) = \{a \in dom(f) \mid f(a) \in dom(g)\}$ and $(g \circ f)(a) = g(f(a))$ for all $a \in dom(g \circ f)$. Note that composition is associative, i. e., given $f \in A \rightharpoonup B$, $g \in B \rightharpoonup C$ and $h \in C \rightharpoonup D$, $(f \circ g) \circ h = f \circ (g \circ h)$.

Let A, B and C be sets and $n \in \mathbb{N}$. Given partial maps $f_1, \ldots, f_n \in A \rightharpoonup B$ and $g \in B^n \rightharpoonup C$, we define the *tuple composition* of g and $f_1, \ldots, f_n$ as the partial map $g[f_1, \ldots, f_n] \in A \rightharpoonup C$ such that $dom(g[f_1, \ldots, f_n]) = \{a \in \bigcap_{i=1}^{n} dom(f_i) \mid \langle f_1(a), \ldots, f_n(a) \rangle \in dom(g)\}$ and $g[f_1, \ldots, f_n](a) = g(\langle f_1(a), \ldots, f_n(a) \rangle)$ for all $a \in dom(g[f_1, \ldots, f_n])$. Note that if $n = 1$ then $g[f_1] = g \circ f_1$. If $n = 0$ and $g \neq \bot$ then $dom(g[]) = A$ and $g[](a) = g(\langle\rangle)$ for all $a \in A$; if $n = 0$ and $g = \bot$ then $g[] = \bot$.

Natural numbers. By $\leq$, we denote the usual linear order on the set of natural numbers $\mathbb{N}$. Note that $\langle \mathbb{N}, \leq \rangle$ has a least element 0 but no greatest element, not even a maximal one. Sometimes, we extend the natural numbers by attaching ω, which results in the complete lattice $\langle \mathbb{N} \cup \{\omega\}, \leq \rangle$ with least element 0 and greatest element ω.

Words. Given a set Σ, a *word* (over the *alphabet* Σ) is a partial map $w \in \mathbb{N} \rightharpoonup \Sigma$ such that $dom(w)$ is an order ideal in $\langle \mathbb{N}, \leq \rangle$, i. e., $dom(w) = \mathbb{N}$ or $dom(w) = \{0, \ldots, n-1\}$ for some $n \in \mathbb{N}$. We call w *infinite* if $dom(w) = \mathbb{N}$, *finite* otherwise. We call w *empty* if $dom(w) = \emptyset$, i. e., if $w = \bot$; we use the symbol ϵ to denote the empty word $\bot$. By Σ^∞, we denote the set of words, by Σ^ω the set of infinite words, by Σ^* the set of finite words and by Σ^+ the set of finite non-empty words. We write $len(w)$ to denote the length of a word $w \in \Sigma^\infty$, where $len(w) = \omega$ if

$w \in \Sigma^\omega$ and $len(w) = n \in \mathbb{N}$ if $w \in \Sigma^*$ and $dom(w) = \{0, \ldots, n-1\}$.

Given $u, v \in \Sigma^\infty$, we define the *catenation* of u and v as the word $uv \in \Sigma^\infty$ where $len(uv) = len(u) + len(v)$ if $u, v \in \Sigma^*$, otherwise $len(uv) = \omega$, and for all $i < len(uv)$, $(uv)(i) = u(i)$ if $i < len(u)$, otherwise $(uv)(i) = v(i - len(u))$. Catenation is an associative operation, therefore we may write a finite non-empty word $w \in \Sigma^+$ as the sequence of its letters $w(0) \ldots w(n-1)$ where $n = len(w)$; alternatively we may write w_i to denote the letter $w(i)$, so $w = w_0 \ldots w_{n-1}$. We extend catenation to sets of words $U, V \subseteq \Sigma^\infty$ in the canonical way, i. e., $UV = \{uv \mid u \in U, v \in V\}$. Given $w \in \Sigma^\infty$ and $W \subseteq \Sigma^\infty$, we may abbreviate the sets $\{w\}W$ and $W\{w\}$ by wW and Ww, respectively.

By the symbol $\preceq$, we denote the order on Σ^∞ which is induced by the order $\trianglelefteq$ on $\mathbb{N} \rightharpoonup \Sigma$. Note that for all $u, w \in \Sigma^\infty$, $u \preceq w$ iff $uv = w$ for some $v \in \Sigma^\infty$; for this reason, we say that u is a *prefix* of w if and only if $u \preceq w$. Given $L \subseteq \Sigma^\infty$, we define $prf(L)$, the *prefix-closure* of L, by $prf(L) = \bigcup_{w \in L} \preceq(w)$, and we say that L is *prefix-closed* if and only if $L = prf(L)$. Note that the set of all prefix-closed subsets of Σ^∞, ordered by inclusion, forms a complete lattice. Given $w \in \Sigma^\infty$, we may sometimes write $prf(w)$ instead of $prf(\{w\})$, i. e., $prf(w) = \preceq(w)$.

Given $w \in \Sigma^\infty$ and $n \in \mathbb{N}$, by w^n we denote the n-fold catenation of w with itself, which is defined inductively through $w^n = ww^{n-1}$ if $n > 0$, otherwise $w^n = \epsilon$. We use w^* to abbreviate the set $\{w^n \mid n \in \mathbb{N}\}$, and we write w^ω to denote $\bigvee w^*$; note that this join of w^* always exists. Finally, we may write w^+ and w^∞ to abbreviate the sets $w^* \setminus \{\epsilon\}$ and $w^* \cup \{w^\omega\}$, respectively.

Given an order $\sqsubseteq$ on Σ, by $\sqsubseteq^\infty$ we denote the order on Σ^∞ which is induced by the lifting of $\sqsubseteq$ to $\mathbb{N} \rightharpoonup \Sigma$. Note that the empty word ϵ is the least element in $\langle \Sigma^\infty, \preceq \rangle$ and, as $\preceq \subseteq \sqsubseteq^\infty$, also in $\langle \Sigma^\infty, \sqsubseteq^\infty \rangle$.

Trees. Given an alphabet Σ, a *tree* (over Σ) is a partial map $t \in \mathbb{N}^* \rightharpoonup \Sigma$ such that $dom(t)$ is an order ideal in $\langle \mathbb{N}^*, \preceq \rangle$ and for every $w \in \mathbb{N}^*$, $\{n \in \mathbb{N} \mid wn \in dom(t)\}$ is an order ideal in $\langle \mathbb{N}, \leq \rangle$. We call the words in $dom(t)$ the *nodes* of t. We call a node w of t a *leaf* node if w is maximal in $dom(t)$, w is a *non-leaf* node otherwise. Given a node w of t, we call $t(w) \in \Sigma$ the *label* of w in t; alternatively we may write t_w to denote the label $t(w)$. We call a tree t *empty* if $dom(t) = \emptyset$, i. e., if $t = \bot$; we use the symbol λ to denote the empty tree $\bot$. We call a tree *finite* resp. *infinite* if its set of nodes is finite resp. infinite. By $\Sigma^\triangle$, we denote the set of trees and by $\Sigma^\blacktriangle$ the set of finite trees.

Given a tree $t \in \Sigma^\triangle$, we call t *infinitely branching* if $\{n \in \mathbb{N} \mid wn \in dom(t)\} = \mathbb{N}$ for some $w \in \mathbb{N}^*$, t is *finitely branching* otherwise. We say that t is of *infinite depth* if $\prec(w) \subseteq dom(t)$ for some $w \in \mathbb{N}^\omega$, t is of *finite depth* otherwise. Note that by König's tree lemma, $t \in \Sigma^\blacktriangle$ if and only if t is finitely branching and of finite depth.

By the symbol $\preceq$, we denote the order on $\Sigma^\triangle$ which is induced by the order $\trianglelefteq$ on $\mathbb{N}^* \rightharpoonup \Sigma$. Given an order $\sqsubseteq$ on Σ, by $\sqsubseteq^\triangle$ we denote the order on $\Sigma^\triangle$ which

is induced by the lifting of $\sqsubseteq$ to $\mathbb{N}^* \rightharpoonup \Sigma$. Note that the empty tree λ is the least element in $\langle \Sigma^\Delta, \preceq \rangle$ and, as $\preceq \subseteq \sqsubseteq^\Delta$, also in $\langle \Sigma^\Delta, \sqsubseteq^\Delta \rangle$.

Chapter 3

Inferences in Semilattices

This chapter defines and investigates conditional inference rules (i. e., rules which are constrained by a side condition) for inferring inequations in a meet-semilattice. We develop an order on the set of all inference rules so that rules can be compared. Soundness and various notions of completeness of conditional inference rules are introduced. Finally, we present a restricted class of inference rules that is important in verification, the assume-guarantee rules. In Chapter 4, a subclass of these, the circular assume-guarantee rules, will be examined thoroughly for soundness and completeness.

3.1 Terms, Formulas and Relations

Definition 3.1 (Variable, Term).
We fix a countably infinite set $\mathcal{V}$, the set of *variables*. The set of *terms* $\mathcal{T}$ is built inductively from variables in $\mathcal{V}$, the nullary operator $\top$ and the binary operator $\sqcap$. We consider $\top$ neutral w. r. t. $\sqcap$, which is seen as associative, commutative and idempotent. We define a binary relation $\sqsubseteq$ on $\mathcal{T}$ by $t \sqsubseteq t'$ if and only if $t \sqcap t' = t$. We define the function $var : \mathcal{T} \to \mathcal{P}_{\text{fin}}(\mathcal{V})$ by $var(t) = \{x \in \mathcal{V} \mid t \sqsubseteq x\}$. We define the function $size : \mathcal{T} \to \mathbb{N}$ by $size(t) = |var(t)|$, i. e., $size(t)$ is the cardinality of $var(t)$.

Notation. Given a set of terms T, we may write $var(T)$ instead of $\bigcup_{t \in T} var(t)$. We call $\sqsupseteq$, the converse of $\sqsubseteq$, the *subterm* relation on terms as for all $t, t' \in \mathcal{T}$, $t \sqsupseteq t'$ iff there is $s \in \mathcal{T}$ such that $s \sqcap t = t'$.

As $\mathcal{T}$ with $\sqcap$ and $\top$ is an idempotent Abelian monoid, $\boldsymbol{\mathcal{T}} = \langle \mathcal{T}, \sqsubseteq \rangle$ is a topped meet-semilattice with meet $\sqcap$ and greatest element $\top$. Note that every term t can be uniquely represented by its set of variables $var(t)$ as $var : \mathcal{T} \to \mathcal{P}_{\text{fin}}(\mathcal{V})$ is an isomorphism from $\langle \mathcal{T}, \sqsubseteq \rangle$ to $\langle \mathcal{P}_{\text{fin}}(\mathcal{V}), \supseteq \rangle$.

Notation. Given $t, t' \in \mathcal{T}$, we define the term $t - t' \in \mathcal{T}$ such that $var(t - t') = var(t) \setminus var(t')$; note that this operation is well-defined because var is an isomor-

phism. We say that $t - t'$ is the result of *factoring* out t' from t. We assume that $\sqcap$ has higher precedence than $-$, i. e., given terms t, t', t'', t''', the expressions $t \sqcap t' - t'' \sqcap t'''$ and $(t \sqcap t') - (t'' \sqcap t''')$ denote the same term.

Definition 3.2 (Valuation).
Given a topped meet-semilattice $\boldsymbol{S} = \langle S, \leq \rangle$, a *valuation* is a partial map from $\mathcal{V}$ to S. Given a valuation $\alpha \in \mathcal{V} \rightharpoonup S$, we write $\hat{\alpha}$ to denote the extension of α to terms, i. e., $\hat{\alpha} \in \mathcal{T} \rightharpoonup S$, where $dom(\hat{\alpha}) = \{t \in \mathcal{T} \mid var(t) \subseteq dom(\alpha)\}$ and for all $t \in dom(\hat{\alpha})$, $\hat{\alpha}(t) = \bigwedge\{\alpha(x) \mid x \in var(t)\}$. By $V(\boldsymbol{S})$, we denote the set of valuations.

Since $\boldsymbol{\mathcal{T}} = \langle \mathcal{T}, \sqsubseteq \rangle$ is a topped meet-semilattice, there are valuations mapping variables to terms. Given such a valuation σ, $\hat{\sigma} \in \mathcal{T} \rightharpoonup \mathcal{T}$ is the partial function on terms which is defined for a term $t = x_1 \sqcap \ldots \sqcap x_n \in \mathcal{T}$ iff σ is defined for all $x_i \in \mathcal{V}$, in which case $\hat{\sigma}$ replaces every x_i by its image under σ, i. e., $\hat{\sigma}(t) = \sigma(x_1) \sqcap \ldots \sqcap \sigma(x_n)$.

Notation. Given $\sigma \in V(\boldsymbol{\mathcal{T}})$ and $T \subseteq \mathcal{T}$ with $var(T) \subseteq dom(\sigma)$, we may write $\hat{\sigma}(T)$ to abbreviate $\{\hat{\sigma}(t) \mid t \in T\}$. Note that $V(\boldsymbol{\mathcal{T}})$ contains the *identity* id on $\mathcal{V}$, i. e., $dom(\mathrm{id}) = \mathcal{V}$ and $\mathrm{id}(x) = x$ for all $x \in \mathcal{V}$.

Definition 3.3 (Invertible).
Given $\sigma \in V(\boldsymbol{\mathcal{T}})$, we say that σ is *invertible* if and only if there is $\tau \in V(\boldsymbol{\mathcal{T}})$ such that $\hat{\tau} \circ \sigma = \mathrm{id}|_{dom(\sigma)}$.

Proposition 3.4. *Obviously, $\sigma \in V(\boldsymbol{\mathcal{T}})$ is invertible if and only if for all $x, y \in dom(\sigma)$, $var(\sigma(x)) \neq \emptyset$, and $x \neq y$ implies $var(\sigma(x)) \cap var(\sigma(y)) = \emptyset$.*

Notation. If the valuation $\sigma \in V(\boldsymbol{\mathcal{T}})$ is invertible then the set $\{\tau \in V(\boldsymbol{\mathcal{T}}) \mid \hat{\tau} \circ \sigma = \mathrm{id}|_{dom(\sigma)}$ and $rng(\tau) \subseteq \mathcal{V}\}$ has a least element, which we denote by σ^{-1}. We call σ^{-1} the *inverse* of σ.

Proposition 3.5. *Let $\sigma, \tau \in V(\boldsymbol{\mathcal{T}})$ be invertible, and let $\alpha \in V(\boldsymbol{S})$ for some topped meet-semilattice $\boldsymbol{S}$. Then obviously*

1. *$dom(\sigma^{-1}) = var(rng(\sigma))$ and $rng(\sigma^{-1}) = dom(\sigma)$,*
2. *$dom(\alpha \circ \sigma^{-1}) = var(rng(\sigma|_{dom(\alpha)}))$,*
3. *$\hat{\tau} \circ \sigma$ is invertible and $(\hat{\tau} \circ \sigma)^{-1} = \sigma^{-1} \circ \tau^{-1}$, and*
4. *if $\sigma \trianglelefteq \tau$ then $\sigma^{-1} \trianglelefteq \tau^{-1}$.*

Definition 3.6 (Substitution).
We say that $\sigma \in V(\boldsymbol{\mathcal{T}})$ is a *substitution* if and only if σ is invertible and $dom(\sigma)$ is finite. We denote the set of substitutions by $\mathcal{S}$.

Proposition 3.7. *Obviously, for all substitutions σ and all finite sets of variables X there is a substitution τ such that $\sigma \trianglelefteq \tau$ and $X \subseteq dom(\tau)$.*

Definition 3.8 (Formula).
A *formula* is a pair $\langle t, t' \rangle$ of terms, where we refer to t as the *left-* and to t' as the *right-hand side* of the formula. We say that a formula $\langle t, t' \rangle$ is in *normal form* iff $var(t) \cap var(t') = \emptyset$, and $\langle t, t' \rangle$ is in *strong normal form* iff it is normal form and $t' \in \mathcal{V}$. By $\mathcal{F}$ resp. $\mathcal{F}_{\mathrm{n}}$ resp. $\mathcal{F}_{\mathrm{N}}$, we denote the set of formulas resp. formulas in normal form resp. formulas in strong normal form.

Notation. Mostly, we will write a formula $\langle t, t' \rangle$ as $t \sqsubseteq t'$. We will do so only if no confusion can arise, i. e., only if in the current context $t \sqsubseteq t'$ cannot mean the statement that t' is a subterm of t. Given a formula $t \sqsubseteq t'$, we denote its variable set by $var(t \sqsubseteq t')$ and its size by $size(t \sqsubseteq t')$, i. e., $var(t \sqsubseteq t') = var(t) \cup var(t')$ and $size(t \sqsubseteq t') = size(t) + size(t')$. Given a set of formulas Φ, we may write $var(\Phi)$ resp. $size(\Phi)$ to abbreviate $\bigcup_{\varphi \in \Phi} var(\varphi)$ resp. $\sum_{\varphi \in \Phi} size(\varphi)$.

Let $\boldsymbol{S} = \langle S, \leq \rangle$ be a topped meet-semilattice, $n \in \mathbb{N}$, $\Gamma \subseteq S^n$ and $t_1, \ldots, t_n \in \mathcal{T}$. We may view the set Γ as a total function in $S^n \to \{0, 1\}$ by identifying it with its characteristic function. Furthermore, we may view every term t_i as a partial map in $V(\boldsymbol{S}) \rightharpoonup S$ by defining $dom(t_i) = \{\alpha \in V(\boldsymbol{S}) \mid var(t_i) \subseteq dom(\alpha)\}$ and $t_i(\alpha) = \hat{\alpha}(t_i)$ for all $\alpha \in dom(t_i)$. Hence the tuple composition of Γ and $t_1, \ldots, t_n$ is well-defined. More precisely, $\Gamma[t_1, \ldots, t_n] \in V(\boldsymbol{S}) \rightharpoonup \{0, 1\}$ with $dom(\Gamma[t_1, \ldots, t_n]) = \{\alpha \in V(\boldsymbol{S}) \mid var(\{t_1, \ldots, t_n\}) \subseteq dom(\alpha)\}$.

Definition 3.9 (Relation).
Given a topped meet-semilattice $\boldsymbol{S} = \langle S, \leq \rangle$, a *relation* $\Gamma[t_1, \ldots, t_n]$ is the tuple composition of Γ and $t_1, \ldots, t_n$ where $n \in \mathbb{N}$, $\Gamma \subseteq S^n$ and $t_1, \ldots, t_n \in \mathcal{T}$. By $\mathcal{R}(\boldsymbol{S})$, we denote the set of relations.

Notation. Sometimes, we call $\Gamma[t_1, \ldots, t_n]$ a *named* relation to emphasize that $\Gamma[t_1, \ldots, t_n]$ rather relates values of variables than tuple entries. We denote the variable set of $\Gamma[t_1, \ldots, t_n]$ by $var(\Gamma[t_1, \ldots, t_n])$, i. e., $var(\Gamma[t_1, \ldots, t_n]) = var(\{t_1, \ldots, t_n\})$. If $\Gamma = S^0$ then we may write *True* to abbreviate the relation $\Gamma[]$.

Definition 3.10 (Sequent).
Given a topped meet-semilattice $\boldsymbol{S} = \langle S, \leq \rangle$, a *sequent* Λ is a finite non-empty word consisting of sets of formulas and relations, i. e., $\Lambda \in (\mathcal{P}(\mathcal{F}) \cup \mathcal{R}(\boldsymbol{S}))^+$. We say that a sequent Λ is *finite* if and only if $\Lambda \in (\mathcal{P}_{\mathrm{fin}}(\mathcal{F}) \cup \mathcal{R}(\boldsymbol{S}))^+$.

Notation. Let Λ be a sequent and let $n = len(\Lambda)$. We write Λ as a comma-separated sequence $\Lambda_0, \ldots, \Lambda_{n-1}$. We may omit set braces around singletons, i. e., we may write $\Lambda_0, \ldots, \Lambda_{i-1}, \{\psi\}, \Lambda_{i+1}, \ldots, \Lambda_{n-1}$ as $\Lambda_0, \ldots, \Lambda_{i-1}, \psi, \Lambda_{i+1}, \ldots, \Lambda_{n-1}$, where ψ is a formula. We denote the variable set of the sequent Λ by $var(\Lambda)$, i. e., $var(\Lambda) = var(\Lambda_0) \cup \ldots \cup var(\Lambda_{n-1})$.

Notation. Let $\sigma \in V(\boldsymbol{\mathcal{T}})$. Given a formula $t \sqsubseteq t'$ with $var(t \sqsubseteq t') \subseteq dom(\sigma)$, we may write $\hat{\sigma}(t \sqsubseteq t')$ to denote the formula $\hat{\sigma}(t) \sqsubseteq \hat{\sigma}(t')$; note that $t \sqsubseteq t'$ is in normal form if and only if $\hat{\sigma}(t \sqsubseteq t')$ is in normal form. Given a set of formulas Φ with $var(\Phi) \subseteq dom(\sigma)$, we may write $\hat{\sigma}(\Phi)$ to abbreviate $\{\hat{\sigma}(\varphi) \mid \varphi \in \Phi\}$. And given a relation $\Gamma[t_1, \ldots, t_n]$ with $var(\Gamma[t_1, \ldots, t_n]) \subseteq dom(\sigma)$, we may write $\hat{\sigma}(\Gamma[t_1, \ldots, t_n])$ instead of the relation $\Gamma[\hat{\sigma}(t_1), \ldots, \hat{\sigma}(t_n)]$. Finally, if $\Lambda_1, \ldots, \Lambda_n$ is a sequent then we may write $\hat{\sigma}(\Lambda_1, \ldots, \Lambda_n)$ to abbreviate the sequent $\hat{\sigma}(\Lambda_1), \ldots, \hat{\sigma}(\Lambda_n)$.

3.2 Satisfiability and Entailment

In this section, we define satisfiability and entailment of relations and (sets of) formulas w. r. t. valuations, then we prove a technical but indispensable lemma about substitution, and we end the section with an investigation of the complexity of the (unconstrained) entailment problem for formulas. Throughout this section we fix a topped meet-semilattice $\boldsymbol{S} = \langle S, \leq \rangle$, i. e., all relations resp. valuations are implicitly in $\mathcal{R}(\boldsymbol{S})$ resp. $V(\boldsymbol{S})$.

Contrary to the standard in logic, we define satisfiability and entailment relative to a given valuation α. Thus, some variables in the formulas and relations may be constrained to the values given by α whereas others (those not in $dom(\alpha)$) are to be considered free. This view of formulas being partially constrained by a valuation will become important in Section 3.3 when defining and comparing inference rules with side conditions. It is also central to the definitions of soundness (Section 3.4) and completeness (Section 3.5).

Definition 3.11 (Satisfiability of Relations).
Given a relation $\Gamma[t_1, \ldots, t_n]$ and a valuation α, we say that $\Gamma[t_1, \ldots, t_n]$ is *satisfiable* under α, denoted by $\boldsymbol{S}, \alpha \models \Gamma[t_1, \ldots, t_n]$, if and only if there is a valuation β with $\alpha \trianglelefteq \beta$ and $var(\Gamma[t_1, \ldots, t_n]) \subseteq dom(\beta)$ such that $\Gamma[t_1, \ldots, t_n](\beta) = 1$.

Proposition 3.12. *Let $\Gamma[t_1, \ldots, t_n]$ be a relation and let α and β be valuations with $\alpha \trianglelefteq \beta$. Obviously, $\boldsymbol{S}, \beta \models \Gamma[t_1, \ldots, t_n]$ implies $\boldsymbol{S}, \alpha \models \Gamma[t_1, \ldots, t_n]$, and $\boldsymbol{S}, \alpha|_{var(\Gamma[t_1, \ldots, t_n])} \models \Gamma[t_1, \ldots, t_n]$ implies $\boldsymbol{S}, \alpha \models \Gamma[t_1, \ldots, t_n]$.*

Lemma 3.13. *Let $\Gamma[t_1, \ldots, t_n]$ be a relation, let α be a valuation and let σ be a substitution with $var(\Gamma[t_1, \ldots, t_n]) \subseteq dom(\sigma)$. Then $\boldsymbol{S}, \alpha \models \hat{\sigma}(\Gamma[t_1, \ldots, t_n])$ implies $\boldsymbol{S}, \hat{\alpha} \circ \sigma \models \Gamma[t_1, \ldots, t_n]$, and if $dom(\alpha) = var(rng(\sigma|_X))$ for some $X \subseteq \mathcal{V}$ then $\boldsymbol{S}, \hat{\alpha} \circ \sigma \models \Gamma[t_1, \ldots, t_n]$ implies $\boldsymbol{S}, \alpha \models \hat{\sigma}(\Gamma[t_1, \ldots, t_n])$.*

Proof. To prove the first implication, let β be a valuation such that $\alpha \trianglelefteq \beta$ and $var(\hat{\sigma}(\Gamma[t_1, \ldots, t_n])) \subseteq dom(\beta)$ and $\hat{\sigma}(\Gamma[t_1, \ldots, t_n])(\beta) = 1$. Clearly, $\hat{\alpha} \circ \sigma \trianglelefteq \hat{\beta} \circ \sigma$ and $var(\Gamma[t_1, \ldots, t_n]) \subseteq dom(\hat{\beta} \circ \sigma)$. Furthermore, we have $\Gamma[t_1, \ldots, t_n](\hat{\beta} \circ \sigma) = \Gamma[\hat{\sigma}(t_1), \ldots, \hat{\sigma}(t_n)](\beta) = \hat{\sigma}(\Gamma[t_1, \ldots, t_n])(\beta) = 1$. Hence $\boldsymbol{S}, \hat{\alpha} \circ \sigma \models \Gamma[t_1, \ldots, t_n]$.

To prove the second implication, let β be a valuation with $\hat{\alpha} \circ \sigma \trianglelefteq \beta$ and $var(\Gamma[t_1, \ldots, t_n]) \subseteq dom(\beta)$ such that $\Gamma[t_1, \ldots, t_n](\beta) = 1$. Define a valuation $\gamma = \alpha \triangledown (\beta|_{\mathcal{V} \setminus X} \circ \sigma^{-1})$, so we have $\alpha \trianglelefteq \gamma$ and $var(\hat{\sigma}(\Gamma[t_1, \ldots, t_n])) \subseteq dom(\gamma)$. Moreover, $\hat{\gamma} \circ \sigma \trianglelefteq \beta$ and $var(\Gamma[t_1, \ldots, t_n]) \subseteq dom(\hat{\gamma} \circ \sigma)$, so $\Gamma[t_1, \ldots, t_n](\hat{\gamma} \circ \sigma) = 1$. Therefore, we have $\hat{\sigma}(\Gamma[t_1, \ldots, t_n])(\gamma) = \Gamma[\hat{\sigma}(t_1), \ldots, \hat{\sigma}(t_n)](\gamma) = \Gamma[t_1, \ldots, t_n](\hat{\gamma} \circ \sigma) = 1$. Hence $\boldsymbol{S}, \alpha \models \hat{\sigma}(\Gamma[t_1, \ldots, t_n])$. □

Definition 3.14 (Satisfiability of Formulas).
Given a set of formulas Φ and a valuation α, we say that Φ is *satisfiable* under α, denoted by $\boldsymbol{S}, \alpha \models \Phi$, if and only if there is a valuation β with $\alpha \trianglelefteq \beta$ and $var(\Phi) \subseteq dom(\beta)$ such that $\hat{\beta}(t) \leq \hat{\beta}(t')$ for all $\langle t, t' \rangle \in \Phi$.

Observe that $\boldsymbol{S}, \alpha \models \Phi$ holds trivially for every set of formulas Φ and every valuation α with $var(\Phi) \subseteq dom(\alpha)$ such that α maps all $x \in var(\Phi)$ to the greatest element of S. Hence every set of formulas Φ is satisfiable under the least valuation $\bot \in V(\boldsymbol{S})$.

Proposition 3.15. *Let Φ be a set of formulas and let α and β be valuations with $\alpha \trianglelefteq \beta$. Obviously, $\boldsymbol{S}, \beta \models \Phi$ implies $\boldsymbol{S}, \alpha \models \Phi$, and $\boldsymbol{S}, \alpha|_{var(\Phi)} \models \Phi$ implies $\boldsymbol{S}, \alpha \models \Phi$.*

Lemma 3.16. *Let Φ be a set of formulas, let α be a valuation and let σ be a substitution with $var(\Phi) \subseteq dom(\sigma)$. Then $\boldsymbol{S}, \alpha \models \hat{\sigma}(\Phi)$ implies $\boldsymbol{S}, \hat{\alpha} \circ \sigma \models \Phi$, and if $dom(\alpha) = var(rng(\sigma|_X))$ for some $X \subseteq \mathcal{V}$ then $\boldsymbol{S}, \hat{\alpha} \circ \sigma \models \Phi$ implies $\boldsymbol{S}, \alpha \models \hat{\sigma}(\Phi)$.*

Proof. To prove the first implication, let β be a valuation such that $\alpha \trianglelefteq \beta$ and $var(\hat{\sigma}(\Phi)) \subseteq dom(\beta)$ and $\hat{\beta}(t) \leq \hat{\beta}(t')$ for all $\langle t, t' \rangle \in \hat{\sigma}(\Phi)$. Obviously, $\hat{\alpha} \circ \sigma \trianglelefteq \hat{\beta} \circ \sigma$ and $var(\Phi) \subseteq dom(\hat{\beta} \circ \sigma)$. Furthermore for every $\langle t, t' \rangle \in \Phi$, we have $\hat{\beta}(\hat{\sigma}(t)) \leq \hat{\beta}(\hat{\sigma}(t'))$, i. e., $(\hat{\beta} \circ \sigma)(t) \leq (\hat{\beta} \circ \sigma)(t')$. Hence $\boldsymbol{S}, \hat{\alpha} \circ \sigma \models \Phi$.

To prove the second implication, let β be a valuation with $\hat{\alpha} \circ \sigma \trianglelefteq \beta$ and $var(\Phi) \subseteq dom(\beta)$ such that $\hat{\beta}(t) \leq \hat{\beta}(t')$ for all $\langle t, t' \rangle \in \Phi$. Define a valuation $\gamma = \alpha \triangledown (\beta|_{\mathcal{V} \setminus X} \circ \sigma^{-1})$, so we have $\alpha \trianglelefteq \gamma$ and $var(\hat{\sigma}(\Phi)) \subseteq dom(\gamma)$. Moreover, $\hat{\gamma} \circ \sigma \trianglelefteq \beta$ and $var(\Phi) \subseteq dom(\hat{\gamma} \circ \sigma)$, so for all $\langle t, t' \rangle \in \Phi$, we have $(\hat{\gamma} \circ \sigma)(t) \leq (\hat{\gamma} \circ \sigma)(t')$, i. e., $\hat{\gamma}(\hat{\sigma}(t)) \leq \hat{\gamma}(\hat{\sigma}(t'))$. This is equivalent to $\hat{\gamma}(t) \leq \hat{\gamma}(t')$ for all $\langle t, t' \rangle \in \hat{\sigma}(\Phi)$. Hence $\boldsymbol{S}, \alpha \models \hat{\sigma}(\Phi)$. □

Definition 3.17 (Truth and Satisfiability of Sequents).
Given a sequent $\Lambda_1, \ldots, \Lambda_n$ and a valuation α, we say that $\Lambda_1, \ldots, \Lambda_n$ is *true* under α if and only if for all $i \in \{1, \ldots, n\}$, $var(\Lambda_i) \subseteq dom(\alpha)$ and $\boldsymbol{S}, \alpha \models \Lambda_i$. We say that $\Lambda_1, \ldots, \Lambda_n$ is *satisfiable* under α, denoted by $\boldsymbol{S}, \alpha \models \Lambda_1, \ldots, \Lambda_n$, if and only if there is a valuation β with $\alpha \trianglelefteq \beta$ such that $\Lambda_1, \ldots, \Lambda_n$ is true under β.

Proposition 3.18. *Let $\Lambda_1, \ldots, \Lambda_n$ be a sequent and let α and β be valuations with $\alpha \trianglelefteq \beta$. Then obviously, $\boldsymbol{S}, \beta \models \Lambda_1, \ldots, \Lambda_n$ implies $\boldsymbol{S}, \alpha \models \Lambda_1, \ldots, \Lambda_n$, and $\boldsymbol{S}, \alpha|_{var(\Lambda_1, \ldots, \Lambda_n)} \models \Lambda_1, \ldots, \Lambda_n$ implies $\boldsymbol{S}, \alpha \models \Lambda_1, \ldots, \Lambda_n$.*

Lemma 3.19. *Let $\Lambda_1, \ldots, \Lambda_n$ be a sequent, let α be a valuation and let σ be a substitution with $var(\Lambda_1, \ldots, \Lambda_n) \subseteq dom(\sigma)$. Then $\boldsymbol{S}, \alpha \models \hat{\sigma}(\Lambda_1, \ldots, \Lambda_n)$ implies $\boldsymbol{S}, \hat{\alpha} \circ \sigma \models \Lambda_1, \ldots, \Lambda_n$, and if $dom(\alpha) = var(rng(\sigma|_X))$ for some $X \subseteq \mathcal{V}$ then $\boldsymbol{S}, \hat{\alpha} \circ \sigma \models \Lambda_1, \ldots, \Lambda_n$ implies $\boldsymbol{S}, \alpha \models \hat{\sigma}(\Lambda_1, \ldots, \Lambda_n)$.*

Proof. To prove the first implication, let β be a valuation with $\alpha \trianglelefteq \beta$ such that $\hat{\sigma}(\Lambda_1, \ldots, \Lambda_n)$ is true under β, i. e., for every $i \in \{1, \ldots, n\}$, $var(\hat{\sigma}(\Lambda_i)) \subseteq dom(\beta)$ and $\boldsymbol{S}, \beta \models \hat{\sigma}(\Lambda_i)$. Thus, $\hat{\alpha} \circ \sigma \trianglelefteq \hat{\beta} \circ \sigma$ and $var(\Lambda_i) \subseteq dom(\hat{\beta} \circ \sigma)$, and by Lemma 3.13 or 3.16, $\boldsymbol{S}, \hat{\beta} \circ \sigma \models \Lambda_i$. Therefore, $\Lambda_1, \ldots, \Lambda_n$ is true under $\hat{\beta} \circ \sigma$. Hence $\boldsymbol{S}, \hat{\alpha} \circ \sigma \models \Lambda_1, \ldots, \Lambda_n$.

To prove the second implication, let β be a valuation with $\hat{\alpha} \circ \sigma \trianglelefteq \beta$ such that $\Lambda_1, \ldots, \Lambda_n$ is true under β, i. e., $var(\Lambda_1, \ldots, \Lambda_n) \subseteq dom(\beta)$ and $\boldsymbol{S}, \beta \models \Lambda_i$ for all $i \in \{1, \ldots, n\}$. Define $\gamma = \alpha \triangledown (\beta|_{\mathcal{V} \setminus X} \circ \sigma^{-1})$, so we have $\alpha \trianglelefteq \gamma$ and $var(\hat{\sigma}(\Lambda_1, \ldots, \Lambda_n)) \subseteq dom(\gamma)$ — to be precise, $dom(\gamma) = var(rng(\sigma|_{X \cup dom(\beta)}))$. Moreover, $\hat{\gamma} \circ \sigma \trianglelefteq \beta$ and $var(\Lambda_1, \ldots, \Lambda_n) \subseteq dom(\hat{\gamma} \circ \sigma)$, so $\boldsymbol{S}, \hat{\gamma} \circ \sigma \models \Lambda_i$ for every $i \in \{1, \ldots, n\}$. This implies $\boldsymbol{S}, \gamma \models \hat{\sigma}(\Lambda_i)$ by Lemma 3.13 or 3.16, so $\hat{\sigma}(\Lambda_1, \ldots, \Lambda_n)$ is true under γ. Hence $\boldsymbol{S}, \alpha \models \hat{\sigma}(\Lambda_1, \ldots, \Lambda_n)$. □

Definition 3.20 (Entailment and Equivalence).
Given the sequents Λ and Λ' and the valuation α, we say that Λ *entails* Λ' under α (or that Λ' is a *logical consequence* of Λ under α), denoted by $\Lambda \models^{\boldsymbol{S}}_{\alpha} \Lambda'$, if and only if for all valuations β with $\alpha \trianglelefteq \beta$ and $var(\Lambda) \cup var(\Lambda') \subseteq dom(\beta)$, $\boldsymbol{S}, \beta \models \Lambda$ implies $\boldsymbol{S}, \beta \models \Lambda'$. We say that Λ and Λ' are *logically equivalent* under α, denoted by $\Lambda \equiv^{\boldsymbol{S}}_{\alpha} \Lambda'$, if and only if $\Lambda \models^{\boldsymbol{S}}_{\alpha} \Lambda'$ and $\Lambda' \models^{\boldsymbol{S}}_{\alpha} \Lambda$.

Notation. If $\alpha = \bot$ then we may write $\Lambda \models^{\boldsymbol{S}} \Lambda'$ resp. $\Lambda \equiv^{\boldsymbol{S}} \Lambda'$ instead of $\Lambda \models^{\boldsymbol{S}}_{\alpha} \Lambda'$ resp. $\Lambda \equiv^{\boldsymbol{S}}_{\alpha} \Lambda'$.

Proposition 3.21. *Let α be valuation and let Λ and Λ' be two sequents such that $\Lambda = \emptyset$ or $\Lambda = True$. Obviously, $\Lambda \models^{\boldsymbol{S}}_{\alpha} \Lambda'$ implies $\boldsymbol{S}, \alpha \models \Lambda'$, and if $var(\Lambda') \subseteq dom(\alpha)$ then $\boldsymbol{S}, \alpha \models \Lambda'$ implies $\Lambda \models^{\boldsymbol{S}}_{\alpha} \Lambda'$.*

Note that unless $\boldsymbol{S}$ is trivial, we cannot identify $\emptyset \models^{\boldsymbol{S}}_{\alpha} \Lambda'$ with $\boldsymbol{S}, \alpha \models \Lambda'$. The reason is that if $\boldsymbol{S}$ is non-trivial then there exist a sequent Λ' and a valuation α with $var(\Lambda') \not\subseteq dom(\alpha)$ such that $\boldsymbol{S}, \alpha \models \Lambda'$ holds but $\emptyset \models^{\boldsymbol{S}}_{\alpha} \Lambda'$ does not hold. Accordingly, the same applies to $True \models^{\boldsymbol{S}}_{\alpha} \Lambda'$.

Proposition 3.22. *Let Λ and Λ' be two sequents and let α and β be two valuations with $\alpha \trianglelefteq \beta$. Then $\Lambda \models^{\boldsymbol{S}}_{\alpha} \Lambda'$ implies $\Lambda \models^{\boldsymbol{S}}_{\beta} \Lambda'$, and $\Lambda \models^{\boldsymbol{S}}_{\beta} \Lambda'$ implies $\Lambda \models^{\boldsymbol{S}}_{\beta|_{var(\Lambda) \cup var(\Lambda')}} \Lambda'$.*

Proof. The first implication is immediate from the definition. To prove the second implication, let $X = var(\Lambda) \cup var(\Lambda')$ and let γ be a valuation with $\beta|_X \trianglelefteq \gamma$ and $var(\Lambda) \cup var(\Lambda') \subseteq dom(\gamma)$ such that $\boldsymbol{S}, \gamma \models \Lambda$. Define $\delta = \beta \triangledown \gamma|_{\mathcal{V} \setminus dom(\beta)}$. Then $\gamma|_X = \delta|_X$, so we have $\boldsymbol{S}, \delta \models \Lambda$. And as $\beta \trianglelefteq \delta$ and $var(\Lambda) \cup var(\Lambda') \subseteq dom(\delta)$, $\Lambda \models^{\boldsymbol{S}}_{\beta} \Lambda'$ implies $\boldsymbol{S}, \delta \models \Lambda'$, which implies $\boldsymbol{S}, \gamma \models \Lambda'$. □

Lemma 3.23. *Let Λ and Λ' be two sequents, α a valuation and σ a substitution with $var(\Lambda) \cup var(\Lambda') \subseteq dom(\sigma)$. Then $\Lambda \models^{\boldsymbol{S}}_{\hat{\alpha}\circ\sigma} \Lambda'$ implies $\hat{\sigma}(\Lambda) \models^{\boldsymbol{S}}_{\alpha} \hat{\sigma}(\Lambda')$, and if $dom(\alpha) = var(rng(\sigma|_X))$ for some $X \subseteq \mathcal{V}$ then $\hat{\sigma}(\Lambda) \models^{\boldsymbol{S}}_{\alpha} \hat{\sigma}(\Lambda')$ implies $\Lambda \models^{\boldsymbol{S}}_{\hat{\alpha}\circ\sigma} \Lambda'$.*

Proof. Observe that $\hat{\sigma}(\Lambda) \models^{\boldsymbol{S}}_{\alpha} \hat{\sigma}(\Lambda')$ if and only if $\hat{\sigma}(\Lambda) \models^{\boldsymbol{S}}_{\beta} \hat{\sigma}(\Lambda')$, and $\Lambda \models^{\boldsymbol{S}}_{\hat{\alpha}\circ\sigma} \Lambda'$ if and only if $\Lambda \models^{\boldsymbol{S}}_{\hat{\beta}\circ\sigma} \Lambda'$, where $\beta = \alpha|_{var(\hat{\sigma}(\Lambda))\cup var(\hat{\sigma}(\Lambda'))}$. Thus w. l. o. g. we may assume that $dom(\alpha) \subseteq var(\hat{\sigma}(\Lambda)) \cup var(\hat{\sigma}(\Lambda'))$.

To prove the first implication, assume that $\Lambda \models^{\boldsymbol{S}}_{\hat{\alpha}\circ\sigma} \Lambda'$ and let β be a valuation with $\alpha \trianglelefteq \beta$ and $var(\hat{\sigma}(\Lambda)) \cup var(\hat{\sigma}(\Lambda')) \subseteq dom(\beta)$ such that $\boldsymbol{S}, \beta \models \hat{\sigma}(\Lambda)$ — w. l. o. g. we assume that $dom(\beta) = var(\hat{\sigma}(\Lambda)) \cup var(\hat{\sigma}(\Lambda'))$. By Lemma 3.19, we have $\boldsymbol{S}, \hat{\beta} \circ \sigma \models \Lambda$. Obviously, $\hat{\alpha} \circ \sigma \trianglelefteq \hat{\beta} \circ \sigma$ and $var(\Lambda) \cup var(\Lambda') \subseteq dom(\hat{\beta} \circ \sigma)$, so because of our assumption, we get $\boldsymbol{S}, \hat{\beta} \circ \sigma \models \Lambda'$. Again by Lemma 3.19, this implies $\boldsymbol{S}, \beta \models \hat{\sigma}(\Lambda')$. Hence $\hat{\sigma}(\Lambda) \models^{\boldsymbol{S}}_{\alpha} \hat{\sigma}(\Lambda')$.

To prove the second implication, let X be a set of variables with $dom(\alpha) = var(rng(\sigma|_X))$ and assume that $\hat{\sigma}(\Lambda) \models^{\boldsymbol{S}}_{\alpha} \hat{\sigma}(\Lambda')$. Let β be a valuation with $\hat{\alpha}\circ\sigma \trianglelefteq \beta$ and $var(\Lambda) \cup var(\Lambda') \subseteq dom(\beta)$ such that $\boldsymbol{S}, \beta \models \Lambda$. Define a valuation $\gamma = \alpha \triangledown (\beta|_{\mathcal{V}\setminus X} \circ \sigma^{-1})$, so we have $\alpha \trianglelefteq \gamma$ and $var(\hat{\sigma}(\Lambda)) \cup var(\hat{\sigma}(\Lambda')) \subseteq dom(\gamma)$ — to be precise, $dom(\gamma) = var(rng(\sigma|_{X\cup dom(\beta)}))$. Moreover, $\hat{\gamma} \circ \sigma \trianglelefteq \beta$ and $var(\Lambda) \cup var(\Lambda') \subseteq dom(\hat{\gamma} \circ \sigma)$, so we have $\boldsymbol{S}, \hat{\gamma} \circ \sigma \models \Lambda$. By Lemma 3.19, this implies $\boldsymbol{S}, \gamma \models \hat{\sigma}(\Lambda)$, which by our assumption implies $\boldsymbol{S}, \gamma \models \hat{\sigma}(\Lambda')$, which again by Lemma 3.19 implies $\boldsymbol{S}, \hat{\gamma} \circ \sigma \models \Lambda'$. Thus, $\boldsymbol{S}, \beta \models \Lambda'$ follows. Hence $\Lambda \models^{\boldsymbol{S}}_{\hat{\alpha}\circ\sigma} \Lambda'$. □

Notation. Let $\alpha \in V(\boldsymbol{S})$ and $t \in \mathcal{T}$. When writing $\hat{\alpha}(t)$, by convention we may assume that $\hat{\alpha}$ is defined for t, i. e., $var(t) \subseteq dom(\alpha)$. Furthermore, we may refrain from denoting the extension of α to terms explicitly, i. e., depending on the context, α may actually stand for $\hat{\alpha}$.

Employing the above conventions, we can collect the preceding lemmas into the following more readable Substitution Lemma.

Proposition 3.24 (Substitution Lemma).
Let Λ and Λ' be two sequents, let α be a valuation and let σ be a substitution. Then

- $\boldsymbol{S}, \alpha \models \sigma(\Lambda)$ *implies* $\boldsymbol{S}, \alpha \circ \sigma \models \Lambda$, *and*
- $\Lambda \models^{\boldsymbol{S}}_{\alpha\circ\sigma} \Lambda'$ *implies* $\sigma(\Lambda) \models^{\boldsymbol{S}}_{\alpha} \sigma(\Lambda')$.

If $dom(\alpha) = var(rng(\sigma|_X))$ for some $X \subseteq \mathcal{V}$ then

- $\boldsymbol{S}, \alpha \models \sigma(\Lambda)$ *if and only if* $\boldsymbol{S}, \alpha \circ \sigma \models \Lambda$, *and*
- $\Lambda \models^{\boldsymbol{S}}_{\alpha\circ\sigma} \Lambda'$ *if and only if* $\sigma(\Lambda) \models^{\boldsymbol{S}}_{\alpha} \sigma(\Lambda')$.

We can reformulate the Substitution Lemma by replacing the valuation α with $\alpha \circ \sigma^{-1}$. Note that we do not need a constraint on the domain of α any more because $dom(\alpha \circ \sigma^{-1}) = var(rng(\sigma|_{dom(\alpha)}))$.

Corollary 3.25. *Let Λ and Λ' be two sequents, let α be a valuation and let σ be a substitution. Then*

- $\boldsymbol{S}, \alpha \circ \sigma^{-1} \models \sigma(\Lambda)$ *if and only if* $\boldsymbol{S}, \alpha \models \Lambda$, *and*
- $\Lambda \models^{\boldsymbol{S}}_{\alpha} \Lambda'$ *if and only if* $\sigma(\Lambda) \models^{\boldsymbol{S}}_{\alpha \circ \sigma^{-1}} \sigma(\Lambda')$.

Note that for $\alpha = \bot$, the Substitution Lemma for entailment boils down to $\Lambda \models^{\boldsymbol{S}} \Lambda'$ iff $\sigma(\Lambda) \models^{\boldsymbol{S}} \sigma(\Lambda')$. Here, it becomes obvious that substitutions must be invertible for the right-to-left direction to hold.

The special case $\alpha = \bot$ deserves more attention, as entailment of sets of formulas under $\bot$ is equivalent to the uniform word problem in $\boldsymbol{S}$. More precisely, such entailment problems can be translated into uniform word problems and vice versa, because every formula $t \sqsubseteq t'$ corresponds to the equation $t \sqcap t' \doteq t$ and every equation $t \doteq t'$ corresponds to the set of formulas $\{t \sqsubseteq t', t' \sqsubseteq t\}$.

By a representation theorem [Dav93, SS03], every meet-semilattice is isomorphic to a sub-meet-semilattice of a product of copies of the two-element meet-semilattice $\langle \{0,1\}, \leq \rangle$. Thus, in every non-trivial meet-semilattice, uniform word problems yield the same answers as in $\langle \{0,1\}, \leq \rangle$. Consequently for sets of formulas Φ and Φ', the entailment problem $\Phi \models^{\boldsymbol{S}} \Phi'$ does not depend on $\boldsymbol{S}$ unless $\boldsymbol{S}$ is trivial, in which case $\Phi \models^{\boldsymbol{S}} \Phi'$ for all $\Phi, \Phi' \subseteq \mathcal{F}$.

Definition 3.26 (Entailment and Equivalence of Formulas).
Given two sets of formulas Φ and Φ', we say that Φ *entails* Φ', denoted by $\Phi \models \Phi'$, if and only if $\Phi \models^{\langle \{0,1\}, \leq \rangle} \Phi'$. We say that Φ and Φ' are *logically equivalent*, denoted by $\Phi \equiv \Phi'$, if and only if $\Phi \models \Phi'$ and $\Phi' \models \Phi$.

Notation. If $\Phi = \{\varphi_1, \ldots, \varphi_n\}$ is a non-empty finite set, we may adopt sequent notation and write $\varphi_1, \ldots, \varphi_n \models \Phi'$ instead of $\{\varphi_1, \ldots, \varphi_n\} \models \Phi'$. Likewise, we may write the right-hand side Φ' in sequent notation if Φ' is a non-empty finite set. Accordingly, the same conventions apply to logical equivalences.

Proposition 3.27. *If $\boldsymbol{S}$ is non-trivial then $\Phi \models \Phi'$ if and only if $\Phi \models^{\boldsymbol{S}} \Phi'$.*

As satisfiability and entailment problems generally depend on the semilattice $\boldsymbol{S}$, their computational complexity will also depend on $\boldsymbol{S}$. However, the entailment problem for formulas does not depend on $\boldsymbol{S}$, therefore its complexity is expressible in terms of the size of formulas only. Naively, uniform word problems in the two-element meet-semilattice $\langle \{0,1\}, \leq \rangle$ are decidable in Co-NP, so Co-NP is an upper bound on deciding entailment of formulas. However, we can do in quadratic time by reducing entailment in $\langle \{0,1\}, \leq \rangle$ to entailment of propositional Horn clauses.

We sketch a procedure for deciding $\Phi \models \varphi'$ in linear time, where Φ is a finite set of formulas and φ' a formula. We can flatten Φ and φ' such that all right-hand sides are variables. This can be done by repeatedly replacing on the right-hand sides all proper subterms $x \sqcap y$ with $x, y \in \mathcal{V}$ by new variables z (depending on x and y) and adding the equations $z \doteq x \sqcap y$ (each translated into the three formulas $z \sqsubseteq x$, $z \sqsubseteq y$ and $x \sqcap y \sqsubseteq z$) to Φ. Formulas of the form $t \sqsubseteq \top \in \Phi$ are simply deleted.[5] This transformation preserves entailment and the blow up of Φ is only linear in the size of the original formulas. Now, we may view $\langle \{0, 1\}, \leq \rangle$ as the two-element boolean algebra because the meet corresponds to conjunction and (the characteristic function of) the order to implication. Thus, the formula $x_1 \sqcap \ldots \sqcap x_m \sqsubseteq y_1 \sqcap \ldots \sqcap y_n$ may be seen as the propositional formula stating that the conjunction of the x_i implies the conjunction of the y_j. In fact, if Φ and φ' are flattened, then each formula directly corresponds to a propositional Horn clause. Hence, the original entailment problem $\Phi \models \varphi'$ in meet-semilattices is reduced (with a linear blowup due to flattening) to entailment of propositional Horn clauses, which is known to be decidable in linear time [DG84].

Deciding the general case $\Phi \models \Phi'$, where Φ and Φ' are finite sets of formulas, takes quadratic time as we must decide $\Phi \models \varphi'$ for every $\varphi' \in \Phi'$.

3.3 Inference Rules

An inference rule is conveniently viewed as a set of inferences, where an inference usually is a pair consisting of a finite set of formulas, the premises, and a formula, the conclusion. When interpreting such an inference with premises Φ and conclusion ψ in a particular meet-semilattice, truth of Φ under some valuation α may imply truth of ψ under α, whereas for another valuation β, truth of Φ under β may not imply truth of ψ under β. Therefore, it makes sense to enhance the notion of inference in such a way that premises and conclusions can be restricted further. We do so by defining an inference to be a triple consisting of premises, conclusion and a valuation restricting some variables of premises and conclusion to particular values. Of course, the valuation must not render the inference useless, i. e., it must not make the premises unsatisfiable.

Throughout the remainder of this chapter we fix a non-trivial topped meet-semilattice $\boldsymbol{S} = \langle S, \leq \rangle$, i. e., all valuations are implicitly in $V(\boldsymbol{S})$.

Definition 3.28 (Inference).
An *inference* is a triple $\langle \Phi, \psi, \alpha \rangle$ where Φ is a finite set of formulas, ψ is a formula and α is a valuation with $dom(\alpha)$ finite such that $\boldsymbol{S}, \alpha \models \Phi$. The elements of Φ are called *premises* (or Φ is called *the premise*), ψ is called *conclusion* and α is called *restriction*. By $I(\boldsymbol{S})$, we denote the set of all inferences.

[5] We assume that φ' is not of the form $t \sqsubseteq \top$, otherwise the entailment $\Phi \models \varphi'$ is trivial.

Notation. We denote the variable set of an inference $\langle \Phi, \psi, \alpha \rangle$ by $var(\langle \Phi, \psi, \alpha \rangle)$, i. e., $var(\langle \Phi, \psi, \alpha \rangle) = var(\Phi) \cup var(\psi) \cup dom(\alpha)$.

Definition 3.29 (Inference Rule).
An *inference rule* I is a set of inferences which is *closed under substitution and entailment*, i. e., for all inferences $\langle \Phi, \psi, \alpha \rangle$ and $\langle \Phi', \psi', \alpha' \rangle$ and all substitutions σ with $var(\langle \Phi, \psi, \alpha \rangle) \subseteq dom(\sigma)$, if $\Phi' \models \sigma(\Phi)$ and $\sigma(\psi) \models \psi'$ and $\alpha \trianglelefteq \alpha' \circ \sigma$ then $\langle \Phi, \psi, \alpha \rangle \in I$ implies $\langle \Phi', \psi', \alpha' \rangle \in I$. By $R(\boldsymbol{S})$, we denote the set of inference rules.

Definition 3.30 (Order on Inference Rules).
Given two inference rules I and I', we say that I' is *weaker than* I (or conversely, I is *stronger than* I') if and only if $I' \subseteq I$.

Obviously, the weaker-than relation is an order the set of all inference rules. In fact, $\langle R(\boldsymbol{S}), \subseteq \rangle$ is a complete lattice, where join is union and meet is intersection.

Definition 3.31 (Induced Rule).
Given a set of inferences I, by $\bar{I}$ we denote the inference rule which is *induced* by I, i. e., the weakest inference rule containing I; formally $\bar{I} = \bigcap \{ I' \in R(\boldsymbol{S}) \mid I \subseteq I' \}$.

Viewing inference rules as (almost) arbitrary sets of inferences admits an elegant way of ordering rules. On the other hand, this view hides the structure of many rules; e. g., there are rules where the premises and conclusions of all inferences share a common pattern, which is not accessible in the set-based view of inference rules. Therefore, we introduce another presentation of inference rules which reveals the common pattern — called schema — directly. Note, however, that such a schema need not exist, hence only a subclass of inference rules can be presented schematically.

Definition 3.32 (Inference Schema).
An *inference schema* (or just *schema*) R is a triple $\langle \Phi, \psi, \Gamma[t_1, \ldots, t_n] \rangle$ where Φ is a finite set of formulas, ψ is a formula and $\Gamma[t_1, \ldots, t_n]$ is a relation. Similar to inferences, we call Φ the *premises*, ψ the *conclusion* and $\Gamma[t_1, \ldots, t_n]$ the *side condition*. We call R *syntactic* if and only if the side condition $\Gamma[t_1, \ldots, t_n]$ is the relation *True*.

Notation. Sometimes, we will write an inference schema $R = \langle \Phi, \psi, \Gamma[t_1, \ldots, t_n] \rangle$ as $R : \Phi / \psi$ if $\Gamma[t_1, \ldots, t_n]$ or as

$$R : \frac{\varphi_1 \ \cdots \ \varphi_m}{\psi} \text{ if } \Gamma[t_1, \ldots, t_n]$$

when $\Phi = \{\varphi_1, \ldots, \varphi_m\}$. If R is syntactic then we may omit the side condition and write $R : \Phi / \psi$, simply. By $var(R)$, we denote the set of variables of the schema $R : \Phi / \psi$ if $\Gamma[t_1, \ldots, t_n]$ i. e., $var(R) = var(\Phi) \cup var(\psi) \cup var(\Gamma[t_1, \ldots, t_n])$.

Definition 3.33 (Associated Inferences).
Given a schema $R : \Phi/\psi$ if $\Gamma[t_1, \ldots, t_n]$, there is a set of inferences $I(R)$ *associated* with R, where

$$I(R) = \{\langle \Phi, \psi, \alpha \rangle \in I(\boldsymbol{S}) \mid dom(\alpha) = var(R) \text{ and } \boldsymbol{S}, \alpha \models \Gamma[t_1, \ldots, t_n]\}.$$

Definition 3.34 (Empty Schema).
Given a schema $R : \Phi/\psi$ if $\Gamma[t_1, \ldots, t_n]$, we call R *empty* if and only if $I(R) = \emptyset$.

Observe that a schema R cannot be both syntactic and empty at the same time.

Lemma 3.35. *Let $R : \Phi/\psi$ if $\Gamma[t_1, \ldots, t_n]$ be a schema. Then*

$$\overline{I(R)} = \{\langle \Phi', \psi', \alpha' \rangle \in I(\boldsymbol{S}) \mid \exists \langle \Phi, \psi, \alpha \rangle \in I(R), \exists \sigma \in \mathcal{S} \text{ with } dom(\sigma) = var(R) \text{ such that } \Phi' \models \sigma(\Phi), \sigma(\psi) \models \psi' \text{ and } \alpha \trianglelefteq \alpha' \circ \sigma\}.$$

Proof. We denote the right-hand side of the above equation by I_R. First, we prove that I_R is an inference rule, i. e., closed under substitution and entailment. Let $\langle \Phi', \psi', \alpha' \rangle \in I_R$, i. e., there exist an inference $\langle \Phi, \psi, \alpha \rangle \in I(R)$ and a substitution σ with $dom(\sigma) = var(R)$ such that $\Phi' \models \sigma(\Phi)$, $\sigma(\psi) \models \psi'$ and $\alpha \trianglelefteq \alpha' \circ \sigma$. Let ρ be a substitution with $var(\langle \Phi', \psi', \alpha' \rangle) \subseteq dom(\rho)$, let $\langle \Phi'', \psi'', \alpha'' \rangle$ be an inference and assume that $\Phi'' \models \rho(\Phi')$, $\rho(\psi') \models \psi''$ and $\alpha' \trianglelefteq \alpha'' \circ \rho$. We have to show $\langle \Phi'', \psi'', \alpha'' \rangle \in I_R$.

There is a substitution τ such that $\rho \trianglelefteq \tau$ and $var(rng(\sigma)) \subseteq dom(\tau)$, so $dom(\tau \circ \sigma) = dom(\sigma) = var(R)$. Hence $\rho(\Phi') = \tau(\Phi')$, $\rho(\psi') = \tau(\psi')$ and $\alpha'' \circ \rho \trianglelefteq \alpha'' \circ \tau$, so we have $\Phi'' \models \tau(\Phi')$, $\tau(\psi') \models \psi''$ and $\alpha' \trianglelefteq \alpha'' \circ \tau$. The latter implies $\alpha' \circ \sigma \trianglelefteq \alpha'' \circ \tau \circ \sigma$, so $\alpha \trianglelefteq \alpha'' \circ (\tau \circ \sigma)$. And by the Substitution Lemma, $\Phi' \models \sigma(\Phi)$ implies $\tau(\Phi') \models \tau(\sigma(\Phi))$, so $\Phi'' \models (\tau \circ \sigma)(\Phi)$. Likewise, $\sigma(\psi) \models \psi'$ implies $(\tau \circ \sigma)(\psi) \models \psi''$. Thus, the substitution $\tau \circ \sigma$ witnesses $\langle \Phi'', \psi'', \alpha'' \rangle \in I_R$, and hence I_R is an inference rule.

Second, we show that $I_R = \bigcap\{I \in R(\boldsymbol{S}) \mid I(R) \subseteq I\}$, i. e., I_R is the weakest rule containing $I(R)$. The right-to-left inclusion is trivial as obviously $I(R) \subseteq I_R$. To prove the left-to-right inclusion, let I be an inference rule containing $I(R)$ and let $\langle \Phi', \psi', \alpha' \rangle \in I_R$. Then there exist an inference $\langle \Phi, \psi, \alpha \rangle \in I(R)$ and a substitution σ with $dom(\sigma) = var(R)$ such that $\Phi' \models \sigma(\Phi)$ and $\sigma(\psi) \models \psi'$ and $\alpha \trianglelefteq \alpha' \circ \sigma$. As $I(R) \subseteq I$, $\langle \Phi, \psi, \alpha \rangle \in I$. Note that $var(\langle \Phi, \psi, \alpha \rangle) = var(R) = dom(\sigma)$, so because I is closed under substitution and entailment, $\langle \Phi, \psi, \alpha \rangle \in I$ implies $\langle \Phi', \psi', \alpha' \rangle \in I$. Hence, I_R is the weakest rule containing $I(R)$. □

Given a schema $R : \Phi/\psi$ if $\Gamma[t_1, \ldots, t_n]$, there are a finite set of formulas Φ' in strong normal form and a formula ψ' in normal form such that $\Phi' \equiv \Phi$ and $\psi' \equiv \psi$, so by switching from R to the schema $R' : \Phi'/\psi'$ if $\Gamma[t_1, \ldots, t_n]$, we do not change the induced inference rule, i. e., $\overline{I(R')} = \overline{I(R)}$.

Notation. Given a schema $R : \Phi/\psi$ if $\Gamma[t_1, \ldots, t_n]$, we may w. l. o. g. assume that $\Phi \subseteq \mathcal{F}_{\mathrm{N}}$ and $\psi \in \mathcal{F}_{\mathrm{n}}$. Given two schemas R and R', we say that R' is *weaker (stronger) than* R iff $\overline{I(R')}$ is weaker (stronger) than $\overline{I(R)}$. The following corollaries to Lemma 3.35 provide sufficient resp. necessary criteria for establishing this weaker-than relation on schemas.

Corollary 3.36. *Let $R : \Phi/\psi$ if $\Gamma[t_1, \ldots, t_n]$ and $R' : \Phi'/\psi'$ if $\Gamma'[t'_1, \ldots, t'_{n'}]$ be two schemas. Then R' is weaker than R if R' is empty or there exists a substitution σ such that*

1. *$dom(\sigma) = var(R)$ and $var(rng(\sigma)) \subseteq var(R')$,*
2. *$\Phi' \models \sigma(\Phi)$ and $\sigma(\psi) \models \psi'$, and*
3. *$\Phi', \Gamma'[t'_1, \ldots, t'_{n'}] \models^{\boldsymbol{S}} \sigma(\Gamma[t_1, \ldots, t_n])$.*

Proof. If R' is empty then $\overline{I(R')} = \emptyset$, hence R' is trivially weaker than R. So let σ be a substitution satisfying the conditions 1, 2 and 3. As $\overline{I(R')}$ is the weakest rule containing $I(R')$, it suffices to show $I(R') \subseteq \overline{I(R)}$, so let $\langle \Phi', \psi', \alpha' \rangle \in I(R')$, i. e., $dom(\alpha') = var(R')$ and $\boldsymbol{S}, \alpha' \models \Gamma'[t'_1, \ldots, t'_{n'}]$. Furthermore $\boldsymbol{S}, \alpha' \models \Phi'$ as $\langle \Phi', \psi', \alpha' \rangle \in I(\boldsymbol{S})$, so we have $\boldsymbol{S}, \alpha' \models \Phi', \Gamma'[t'_1, \ldots, t'_{n'}]$. By the conditions 2 and 3, this implies $\boldsymbol{S}, \alpha' \models \sigma(\Phi)$ and $\boldsymbol{S}, \alpha' \models \sigma(\Gamma[t_1, \ldots, t_n])$, respectively. By the Substitution Lemma, we get $\boldsymbol{S}, \alpha' \circ \sigma \models \Phi$ and $\boldsymbol{S}, \alpha' \circ \sigma \models \Gamma[t_1, \ldots, t_n]$, whereby the former also implies that $\langle \Phi, \psi, \alpha' \circ \sigma \rangle$ is an inference. As $var(rng(\sigma)) \subseteq dom(\alpha')$ by condition 1, we have $dom(\alpha' \circ \sigma) = dom(\sigma) = var(R)$, so actually $\langle \Phi, \psi, \alpha' \circ \sigma \rangle$ is an inference in $I(R)$. Due to condition 2, this inference and the substitution σ witness $\langle \Phi', \psi', \alpha' \rangle \in \overline{I(R)}$ by Lemma 3.35. □

Corollary 3.37. *Let $R : \Phi/\psi$ if $\Gamma[t_1, \ldots, t_n]$ and $R' : \Phi'/\psi'$ if $\Gamma'[t'_1, \ldots, t'_{n'}]$ be two schemas. If R' is weaker than R then R' is empty or there exists a substitution σ such that*

1. *$dom(\sigma) = var(R)$ and $var(rng(\sigma)) \subseteq var(R')$, and*
2. *$\Phi' \models \sigma(\Phi)$ and $\sigma(\psi) \models \psi'$.*

Proof. Assume that R' is weaker than R but R' is not empty. Then there exists an inference $\langle \Phi', \psi', \alpha' \rangle \in I(R')$; note that $dom(\alpha') = var(R')$. As R' is weaker than R, we have $\langle \Phi', \psi', \alpha' \rangle \in \overline{I(R)}$, so by Lemma 3.35 there exist an inference $\langle \Phi, \psi, \alpha \rangle \in I(R)$ and a substitution σ with $dom(\sigma) = var(R)$ such that $\Phi' \models \sigma(\Phi)$, $\sigma(\psi) \models \psi'$ and $\alpha \trianglelefteq \alpha' \circ \sigma$. Thus, we have $var(R) = dom(\alpha) \subseteq dom(\alpha' \circ \sigma) \subseteq dom(\sigma) = var(R)$, hence $dom(\alpha' \circ \sigma) = dom(\sigma)$, which implies $var(rng(\sigma)) \subseteq dom(\alpha') = var(R')$. □

3.4 Soundness

To be of any use, an inference rule should be sound, i. e., it should not derive false conclusions from true premises. Put positively, all conclusions should be entailed by their premises, at least under the corresponding restriction.

Definition 3.38 (Soundness).
We say that an inference $\langle \Phi, \psi, \alpha \rangle$ is *sound* if and only if $\Phi \models_\alpha^{\boldsymbol{S}} \psi$, and we say that $\langle \Phi, \psi, \alpha \rangle$ is *syntactically sound* if and only if $\Phi \models \psi$. We say that an inference rule I is *(syntactically) sound* if and only if $\langle \Phi, \psi, \alpha \rangle$ is (syntactically) sound for all $\langle \Phi, \psi, \alpha \rangle \in I$. We say that a schema R is *(syntactically) sound* if and only if $\overline{I(R)}$ is (syntactically) sound.

Proposition 3.39. *Let I and I' be inference rules such that I is stronger than I'. Obviously, if I is syntactically sound then I is sound, and if I is (syntactically) sound then I' is (syntactically) sound, too.*

Proposition 3.40. *Let $R : \Phi/\psi$ if $\Gamma[t_1, \ldots, t_n]$ be a schema. Then the following statements are equivalent:*

1. *R is sound.*
2. *$\Phi, \Gamma[t_1, \ldots, t_n] \models^{\boldsymbol{S}} \psi$.*
3. *For all valuations α with $dom(\alpha) = var(\Gamma[t_1, \ldots, t_n])$, $\boldsymbol{S}, \alpha \models \Gamma[t_1, \ldots, t_n]$ implies $\Phi \models_\alpha^{\boldsymbol{S}} \psi$.*

Proof. We will show that 1 implies 3, 3 implies 2, and 2 implies 1.

To prove that 1 implies 3, let α be an arbitrary valuation such that $dom(\alpha) = var(\Gamma[t_1, \ldots, t_n])$ and $\boldsymbol{S}, \alpha \models \Gamma[t_1, \ldots, t_n]$. Let β be a valuation with $\alpha \trianglelefteq \beta$ and $var(\Phi) \cup var(\psi) \subseteq dom(\beta)$ such that $\boldsymbol{S}, \beta \models \Phi$. As $var(R) \subseteq dom(\beta)$, w. l. o. g. we may assume that $dom(\beta) = var(R)$, so $\langle \Phi, \psi, \beta \rangle \in I(R)$ by definition of $I(R)$. As $I(R) \subseteq \overline{I(R)}$, the inference $\langle \Phi, \psi, \beta \rangle$ is sound by 1, which implies $\boldsymbol{S}, \beta \models \psi$ by the definition of soundness. Hence $\Phi \models_\alpha^{\boldsymbol{S}} \psi$.

To prove that 3 implies 2, let β be a valuation with $var(R) \subseteq dom(\beta)$ such that $\boldsymbol{S}, \beta \models \Phi, \Gamma[t_1, \ldots, t_n]$. By 3, $\Phi \models_\alpha^{\boldsymbol{S}} \psi$ follows where $\alpha = \beta|_{var(\Gamma[t_1, \ldots, t_n])}$. So as $\alpha \trianglelefteq \beta$, $\boldsymbol{S}, \beta \models \Phi$ implies $\boldsymbol{S}, \beta \models \psi$.

To prove that 2 implies 1, let $\langle \Phi', \psi', \alpha' \rangle \in \overline{I(R)}$, so by Lemma 3.35 there exist an inference $\langle \Phi, \psi, \alpha \rangle \in I(R)$ and a substitution σ with $dom(\sigma) = var(R)$ such that $\Phi' \models \sigma(\Phi)$, $\sigma(\psi) \models \psi'$ and $\alpha \trianglelefteq \alpha' \circ \sigma$. By the definition of $I(R)$, we know that $dom(\alpha) = var(R)$ and $\boldsymbol{S}, \alpha \models \Phi, \Gamma[t_1, \ldots, t_n]$. This implies $\boldsymbol{S}, \alpha \models \psi$ by 2, so we have $\Phi \models_\alpha^{\boldsymbol{S}} \psi$. As $\alpha \trianglelefteq \alpha' \circ \sigma$, we get $\Phi \models_{\alpha' \circ \sigma}^{\boldsymbol{S}} \psi$, which by the Substitution Lemma implies $\sigma(\Phi) \models_{\alpha'}^{\boldsymbol{S}} \sigma(\psi)$. Hence $\Phi' \models_{\alpha'}^{\boldsymbol{S}} \psi'$ follows from $\Phi' \models \sigma(\Phi)$ and $\sigma(\psi) \models \psi'$, so the inference $\langle \Phi', \psi', \alpha' \rangle$ is sound. □

Proposition 3.41. *Let $R : \Phi/\psi$ if $\Gamma[t_1, \ldots, t_n]$ be a schema. Then the following statements are true:*

1. *If $\Phi \models \psi$ then R is syntactically sound.*
2. *If R is syntactically sound and not empty then $\Phi \models \psi$.*
3. *If R is syntactic and sound then R is syntactically sound and not empty.*

Proof. To prove 1, let $\langle \Phi', \psi', \alpha' \rangle \in \overline{I(R)}$. By Lemma 3.35, there is a substitution σ with $dom(\sigma) = var(R)$ such that $\Phi' \models \sigma(\Phi)$ and $\sigma(\psi) \models \psi'$. By the Substitution Lemma, $\Phi \models \psi$ is equivalent to $\sigma(\Phi) \models \sigma(\psi)$. Hence, $\Phi' \models \psi'$ follows, so R is syntactically sound.

To prove 2, consider some inference $\langle \Phi, \psi, \alpha \rangle \in I(R)$; such an inference exists as R is not empty. Then $\Phi \models \psi$ follows from R being syntactically sound.

To prove 3, observe that since R is syntactic it cannot be empty. By Proposition 3.40, soundness of R implies Φ, *True* $\models^{\boldsymbol{S}} \psi$, which in turn implies $\Phi \models \psi$. Hence R is syntactically sound by 1. □

As a consequence of the above proposition, if a syntactically sound schema R is not syntactic then it is either empty or unnecessarily weak. In particular, if it is not empty then the syntactic schema $R' : \Phi/\psi$ is both sound and stronger than R.

We can measure the extent of soundness of a schema by introducing a notion of soundness for the variables occuring in the schema. When all occuring variables are sound we classify the schema as fully sound.

Definition 3.42 (Sound Variable, Full Soundness).
Given a schema $R : \Phi/t \sqsubseteq t'$ if $\Gamma[t_1, \ldots, t_n]$, we call a variable $x \in var(R)$ *sound* in R if and only if $\Phi, \Gamma[t_1, \ldots, t_n] \models^{\boldsymbol{S}} t \sqsubseteq x$, and we call x *syntactically sound* in R if and only if $\Phi \models t \sqsubseteq x$. We say that R is *fully sound* if and only if every $x \in var(R)$ is sound in R.

Proposition 3.43. *Let $R : \Phi/t \sqsubseteq t'$ if $\Gamma[t_1, \ldots, t_n]$ be a schema. Then the following statements are obviously true:*

1. *Every $x \in var(t)$ is syntactically sound in R.*
2. *R is sound if and only if every $x \in var(t')$ is sound in R.*
3. *R being fully sound implies R being sound.*
4. *If $var(R) = var(t \sqsubseteq t')$ then R being sound implies R being fully sound.*

Example 3.44. Let x, y and z be three distinct variables and consider the syntactic schema $R : \{x \sqsubseteq y, z \sqsubseteq y\}/x \sqsubseteq y$. Obviously, R is sound but z is not sound in R. Hence R witnesses the existence of sound schemas which fail to be fully sound.

We finish this section with a full soundness preserving transformation which strengthens a schema by adding a variable to the right-hand side of the conclusion.

Proposition 3.45. *Let $R : \Phi/t \sqsubseteq t'$ if $\Gamma[t_1, \ldots, t_n]$ be a schema and $x \in var(R)$ with $x \notin var(t \sqsubseteq t')$. Then the schema $R' : \Phi/t \sqsubseteq t' \sqcap x$ if $\Gamma[t_1, \ldots, t_n]$ is stronger than R, and if R is fully sound and then R' is fully sound, too.*

Proof. That R' is stronger than R follows from Corollary 3.36 as the substitution $\mathrm{id}|_{var(R')}$ trivially satisfies the conditions 1, 2 and 3. And from the construction of R', it is also obvious that a variable $y \in var(R') = var(R)$ is sound in R' if and only if it is sound in R. Hence, R being fully sound implies R' being fully sound. □

3.5 Completeness

Entailment may be viewed as a ternary relation relating sets of formulas Φ and Ψ and valuations α. Likewise, every set of inferences may be viewed as a ternary relation relating finite (resp. singleton) sets of formulas Φ and $\{\psi\}$ and valuations α with finite domain. Consequently, any sound inference rule I may be seen as an under-approximation of entailment, as for all inferences $\langle \Phi, \psi, \alpha \rangle$, $\langle \Phi, \psi, \alpha \rangle \in I$ implies $\Phi \models_\alpha^{\boldsymbol{S}} \psi$, i. e., if Φ, $\{\psi\}$ and α are related by the set of inferences I then they are related by entailment, too.

Not all sound inference rules are useful, e. g., the empty set of inferences is a completely useless inference rule. Obviously, an inference rule is the better the closer it approximates entailment; in the best case, which we call (full) completeness, it even over-approximates entailment.

Definition 3.46 (Full Completeness of Inference Rules).
Given an inference rule I, we say that I is *fully complete* if and only if for all inferences $\langle \Phi, \psi, \alpha \rangle$, $\Phi \models_\alpha^{\boldsymbol{S}} \psi$ implies $\langle \Phi, \psi, \alpha \rangle \in I$.

Proposition 3.47. *Let I and I' be inference rules such that I is weaker than I'. If I is fully complete then obviously I' is fully complete, too.*

Proposition 3.48. *Let $I = \{\langle \Phi, \psi, \alpha \rangle \in I(\boldsymbol{S}) \mid \Phi \models_\alpha^{\boldsymbol{S}} \psi\}$. Obviously, I is the strongest sound and the weakest fully complete inference rule; in particular, I is the only sound and fully complete rule.*

Corollary 3.49. *There is no schema R such that $\overline{I(R)}$ is sound and fully complete.*

Proof. Towards a contradiction, we assume a schema $R : \Phi/\psi$ if $\Gamma[t_1, \ldots, t_n]$ such that $\overline{I(R)}$ is sound and fully complete, i. e., $\overline{I(R)} = \{\langle \Phi, \psi, \alpha \rangle \in I(\boldsymbol{S}) \mid \Phi \models_\alpha^{\boldsymbol{S}} \psi\}$. Let x and y be two distinct variables. As $x \sqsubseteq y \models x \sqsubseteq y$, i. e., $\{x \sqsubseteq y\} \models_\bot^{\boldsymbol{S}} x \sqsubseteq y$,

the inference $\langle\{x \sqsubseteq y\}, x \sqsubseteq y, \bot\rangle$ is in $\overline{I(R)}$. By Lemma 3.35, there exist an inference $\langle\Phi, \psi, \alpha\rangle \in I(R)$ and a substitution σ such that $\alpha \trianglelefteq \bot \circ \sigma$, so $\alpha = \bot$, i. e., $\langle\Phi, \psi, \bot\rangle \in \overline{I(R)}$. Hence $\Phi \models_{\bot}^{\boldsymbol{S}} \psi$, i. e., $\Phi \models \psi$, which implies that R is syntactically sound by Proposition 3.41.

Now, let β be a valuation with $y \in dom(\beta)$ and $\beta(y)$ being the greatest element of $\boldsymbol{S}$. Then $\emptyset \models_{\beta}^{\boldsymbol{S}} x \sqsubseteq y$, so the inference $\langle\emptyset, x \sqsubseteq y, \beta\rangle$ is in $\overline{I(R)}$. And as R is syntactically sound, we conclude $\emptyset \models x \sqsubseteq y$, which is a contradiction. □

As full completeness is not useful for classifying the quality of schematically presented rules, we introduce three notions of completeness that are tailored for inference schemas.

Definition 3.50 (Notions of Completeness for Schemas).
Given a schema $R : \Phi/\psi$ if $\Gamma[t_1, \ldots, t_n]$, we say that R is *backward complete* (or just *complete*) if and only if for all valuations α with $dom(\alpha) = var(\psi)$, $\boldsymbol{S}, \alpha \models \psi$ implies $\boldsymbol{S}, \alpha \models \Phi, \Gamma[t_1, \ldots, t_n]$.

We say that R is *forward complete* if and only if for all valuations α with $dom(\alpha) = var(\Phi, \psi)$, $\boldsymbol{S}, \alpha \models \Phi, \psi$ implies $\boldsymbol{S}, \alpha \models \Gamma[t_1, \ldots, t_n]$.

Finally, we say that the side condition $\Gamma[t_1, \ldots, t_n]$ is *complete* in R if and only if for all valuations α with $dom(\alpha) = var(\Gamma[t_1, \ldots, t_n])$, $\Phi \models_{\alpha}^{\boldsymbol{S}} \psi$ implies $\boldsymbol{S}, \alpha \models \Gamma[t_1, \ldots, t_n]$.

Let $R : \Phi/\psi$ if $\Gamma[t_1, \ldots, t_n]$ be a schema. Backward completeness of R enables backward reasoning: Whenever the conclusion ψ is true under some valuation α then α can be extended to a restriction β such that Φ is true under β and $\langle\Phi, \psi, \beta\rangle$ is an inference of R. In other words, backward completeness ensures that all true conclusions are derivable. It is this property, which is most often referred to as completeness.

Forward completeness relates to forward reasoning: Given a valuation α which makes the premises Φ true, we want to infer that the conclusion ψ is true. If R is forward complete then α can be extended to a restriction β such that and $\langle\Phi, \psi, \beta\rangle$ is an inference of R whenever the truth of ψ is consistent with the truth of Φ. Thus, forward completeness ensures that all consistent conclusions are derivable from true premises. Put another way: Forward completeness ensures that the side condition is not too restrictive, i. e., does not prohibit consistent inferences.

Similar to full completeness, completeness of the side condition aims at characterizing entailment. The side condition $\Gamma[t_1, \ldots, t_n]$ is complete in R iff truth of $\Gamma[t_1, \ldots, t_n]$ under a valuation α is necessary for the premises Φ to entail the conclusion ψ under α. Compare this to the statement of Proposition 3.40: R is sound iff truth of $\Gamma[t_1, \ldots, t_n]$ under α is sufficient for Φ to entail ψ under α.

Note that the three notions of completeness are pairwise incomparable, i. e., none entails any of the others. See Chapter 4, which shows that completeness in

the side condition is neither entailed by backward or forward completeness (Section 4.3), nor does it entail any of the two (Section 4.6). The following example demonstrates that neither backward completeness entails forward completeness, nor the other way round.

Example 3.51. Let x and y be two distinct variables. To demonstrate that backward completeness does not entail forward completeness, we consider the schema $R : \{x \sqsubseteq y\}/x \sqsubseteq \top$ if $\Gamma[y]$, where Γ is a strict subset of S containing 1, the greatest element of S. Obviously, R is backward complete as every valuation α with $dom(\alpha) = \{x\}$ can be extended to a valuation β with $dom(\beta) = \{x, y\}$ such that $\beta(y) = 1$. However, as Γ is a strict subset of S there exists a valuation γ with $dom(\gamma) = \{x, y\}$ such that $\gamma(x) = \gamma(y) \notin \Gamma$. Hence $\boldsymbol{S}, \gamma \models x \sqsubseteq y$, $x \sqsubseteq \top$ and $\boldsymbol{S}, \gamma \not\models \Gamma[y]$, i. e., R is not forward complete.

For the other direction, consider the syntactic schema $R' : \{x \sqsubseteq y\}/x \sqcap y \sqsubseteq \top$. Obviously, R' is forward complete, as is any syntactic schema. However, as $\boldsymbol{S}$ is non-trivial there is a valuation α' with $dom(\alpha') = \{x, y\}$ such that $\alpha'(x) \not\leq \alpha'(y)$. Hence $\boldsymbol{S}, \alpha' \models x \sqcap y \sqsubseteq \top$ and $\boldsymbol{S}, \alpha' \not\models x \sqsubseteq y$, i. e., R' is not backward complete.

Although the different notions of completeness have turned out incomparable, forward completeness seems to be the weakest notion, for under certain conditions it is trivially entailed by the others.

Proposition 3.52. *Let $R : \Phi/\psi$ if $\Gamma[t_1, \ldots, t_n]$ be a schema. Then the following statements are obviously true:*

1. *If $var(\Phi) \subseteq var(\psi)$ then R being backward complete implies R being forward complete.*

2. *If $var(\Phi, \psi) \subseteq var(\Gamma[t_1, \ldots, t_n])$ then $\Gamma[t_1, \ldots, t_n]$ being complete in R implies R being forward complete.*

Finally, it should be said that for syntactic schemas, the notions of forward completeness and completeness of the side condition are void, whereas backward completeness remains a meaningful notion. More precisely, given a syntactic schema $R : \Phi/\psi$, R is forward complete and *True* is complete in R, trivially. However, R is backward complete iff for all valuations α with $dom(\alpha) = var(\psi)$, truth of ψ under α implies satisfiability of Φ under α.

3.6 Assume-Guarantee Rules

In verification we often encounter formulas like $z \sqcap x \sqsubseteq y$ where the variables x, y and z denote a system, a property of that system and a property of its environment, respectively. In this context, the formula $z \sqcap x \sqsubseteq y$ may be read as follows: Assuming that its environment satisfies z, the system x guarantees

y. Because of this reading we call z an assumption and y a guarantee. Inference rules whose premises and conclusion may be read in the style above are called assume-guarantee rules.

In this section, we generalize the notion of assume-guarantee rule into a purely syntactical one, in particular, we develop a characterization of assumptions and guarantees that does not require semantic knowledge about whether a variable represents a system or a property.

Definition 3.53 (Assumption, Guarantee).
Given a formula $t \sqsubseteq t'$, we call a variable x an *assumption* of $t \sqsubseteq t'$ if and only if $x \in var(t)$, and we call x a *guarantee* of $t \sqsubseteq t'$ if and only if $x \in var(t')$. Given a schema $R : \Phi/\psi$ if $\Gamma[t_1, \ldots, t_n]$ and a variable $x \in var(R)$, we say that

- x is a *(global) assumption* of R if and only if x is an assumption of ψ,
- x is a *(global) guarantee* of R of R if and only if x is a guarantee of ψ, and
- x is *auxiliary variable* of R if and only if $x \notin var(\psi)$.

Furthermore, we call x a *local assumption* (resp. *local guarantee*) of R if and only if x is an assumption (resp. guarantee) of some $\varphi \in \Phi$.

Observe that every $x \in var(R)$ is either a global assumption or a global guarantee or an auxiliary variable since the conclusion is in normal form. However, a variable may be both local assumption and local guarantee even though the premises are in strong normal form.

Notation. Due to Lemma 3.35, we may henceforth identify schemas with their induced inference rules. I. e., depending on the context, we may view an inference rule R as its schema $R : \Phi/\psi$ if $\Gamma[t_1, \ldots, t_n]$ or as the inference rule (presented as set of inferences) $\overline{I(R)}$ induced by that schema.

Definition 3.54 (Assume-Guarantee Rule).
We call an inference rule $R : \Phi/\psi$ if $\Gamma[t_1, \ldots, t_n]$ an *assume-guarantee rule* (or *A-G rule* for short) if and only if R has global assumptions and for every $x \in var(R)$, either x is a global assumption or x is a local guarantee, but not both. Given an A-G rule $R : \Phi/\psi$ if $\Gamma[t_1, \ldots, t_n]$, we say that R is *circular* if and only if $\Phi \not\models \psi$.

Recall that the premises of an inference rule are in strong normal form, therefore $R : \Phi/t \sqsubseteq t'$ if $\Gamma[t_1, \ldots, t_n]$ is an assume-guarantee rule iff $t \neq \top$ and $var(R) = var(t) \uplus \{x \mid t'' \sqsubseteq x \in \Phi\}$, where $\uplus$ denotes the disjoint union of sets. In particular, if R is an A-G rule then the side condition cannot introduce variables that do not already occur in the premises or the conclusion, i. e., $var(\Gamma[t_1, \ldots, t_n]) \subseteq var(\Phi, \psi)$. The rationale behind the definition of A-G rules is the following:

$$R_1 : \frac{\mathsf{S}_1 \sqsubseteq \mathsf{P}_1 \qquad \mathsf{S}_2 \sqsubseteq \mathsf{P}_2}{\mathsf{S}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2} \qquad R_4 : \frac{\mathsf{P}_2 \sqcap \mathsf{S}_1 \sqsubseteq \mathsf{P}_1 \qquad \mathsf{P}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_2}{\mathsf{S}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2}$$

$$R_2 : \frac{\mathsf{S}_1 \sqsubseteq \mathsf{P}_1 \qquad \mathsf{P}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_2}{\mathsf{S}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2} \qquad R_5 : \frac{\mathsf{P} \sqcap \mathsf{S}_1 \sqsubseteq \mathsf{P}_1 \qquad \mathsf{P}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_2}{\mathsf{S}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2}$$

$$R_3 : \frac{\mathsf{P} \sqcap \mathsf{S}_1 \sqsubseteq \mathsf{P}_1 \qquad \mathsf{P}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_2}{\mathsf{P} \sqcap \mathsf{S}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2} \qquad R_6 : \frac{\mathsf{P}_2 \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_1 \qquad \mathsf{P}_1 \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_2 \qquad \mathsf{S} \sqsubseteq \mathsf{P}}{\mathsf{P} \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2}$$

Figure 3.1: Sample assume-guarantee rules

1. *Global assumptions may be strengthened.* More precisely, truth of the premises should be preserved if a global assumption is strengthened (i. e., in the valuation its value is replaced by a smaller value). This requires that global assumptions are never local guarantees.

2. *Global assumptions must be explicit.* For instance, this forces every auxiliary local assumption (i. e., every auxiliary variable which is a local assumption) to be a local guarantee also, as otherwise the auxiliary local assumption would behave (according to 1) like a global assumption although it is not explicit. Thus, only global assumptions need not be local guarantees.

The requirement that at least one global assumption must exist is a purely technical one. We want to make sure that the left-hand side of the conclusion is different from $\top$, so it can be in the range of substitutions. Note that every inference rule $R : \Phi/\top \sqsubseteq t'$ if $\Gamma[t_1, \ldots, t_n]$ without global assumptions can be transformed into an inference rule $R' : \Phi/x \sqsubseteq t'$ if $\Gamma'[t_1, \ldots, t_n, x]$ with global assumption $x \in \mathcal{V} \setminus var(R)$, where $\Gamma'[t_1, \ldots, t_n, x] \equiv^{\boldsymbol{S}} \top \sqsubseteq x, \Gamma[t_1, \ldots, t_n]$. Obviously, this transformation preserves the partitioning of variables into global assumptions and local guarantees. Furthermore, it preserves soundness and full soundness.

The following example shows that the above definition of A-G rules and circular A-G rules still encompasses all the rules that one usually (i. e., using the semantic system/property distinction) terms 'assume-guarantee rule'.

Example 3.55. As the definition of A-G rules resp. circular A-G rules does not involve the side condition, we may illustrate these notions by syntactic rules only. Let S, S_1, S_2, P, P_1, P_2 be six distinct variables. Figure 3.1 presents six inference rules.

The rules R_1 to R_4 are A-G rules but R_5 and R_6 are not. More precisely, R_5 fails to be an A-G rule because the auxiliary variable P acts as an implicit global assumption, whereas R_6 fails because the global assumption P also is a local guarantee. Note that R_5 can be modified into a weaker A-G rule in two

ways, namely by removing P from the first premise (resulting in R_2) or by adding it to the left-hand side of the conclusion (resulting in R_3). On the other hand, R_6 can be modified into a stronger A-G rule in two ways, namely by removing the last premise or by removing P from the conclusion.

The rules R_1 to R_4 show patterns that are common in the literature about assume-guarantee reasoning. For the sake of explanation, we assume that the variables P, P_1, P_2 resp. S, S_1, S_2 denote properties resp. systems. Note that in all rules the system variables function as global assumptions. Therefore in this system/property distinguishing setting, by convention the term 'assumption' refers to a property variable that functions as assumption.

Rule R_1 states that if system S_i guarantees property P_i for all $i \in \{1, 2\}$ then the composition of both systems $\mathsf{S}_1 \sqcap \mathsf{S}_2$ guarantees the conjunction of both properties $\mathsf{P}_1 \sqcap \mathsf{P}_2$. As no assumptions are involved, R_1 is usually called 'compositional' rather than 'assume-guarantee' rule. However, one may view such rules as special cases of assume-guarantee reasoning.

Rule R_2 says that if S_1 guarantees P_1 and assuming P_1, S_2 guarantees P_2 then $\mathsf{S}_1 \sqcap \mathsf{S}_2$ guarantees $\mathsf{P}_1 \sqcap \mathsf{P}_2$. This is called a 'hierarchical' assume-guarantee rule as the properties form a hierarchy according to which property is guaranteed assuming which other property; here, P_2 is guaranteed assuming P_1, which is guaranteed without assumptions.

The rule R_2 arose from R_1 by introducing the assumption P_1 into the second premise. R_3 arises from R_2 by introducing the assumption P into the first premise and into the conclusion. Thus, R_3 states that if assuming P, S_1 guarantees P_1 and assuming P_1, S_2 guarantees P_2 then assuming P, $\mathsf{S}_1 \sqcap \mathsf{S}_2$ guarantees $\mathsf{P}_1 \sqcap \mathsf{P}_2$. Obviously, R_3 is still a hierarchical rule.

Finally R_4, which also arises from R_2 by introducing an assumption into the first premise, says that if assuming P_2, S_1 guarantees P_1 and assuming P_1, S_2 guarantees P_2 then $\mathsf{S}_1 \sqcap \mathsf{S}_2$ guarantees $\mathsf{P}_1 \sqcap \mathsf{P}_2$. Here, each of the properties is guaranteed assuming the other, i. e., the dependency between them is not hierarchical but cyclic, hence the name 'circular' assume-guarantee rule. Note that this example coincides with our above definition of circular A-G rule; in fact, out of the four A-G rules R_1 to R_4, only R_4 is circular.

From its definition, the circularity of an A-G rule seems to be related to a lack of syntactical soundness, which is made precise in the following proposition.

Proposition 3.56. *Let $R : \Phi/t \sqsubseteq t'$ if $\Gamma[t_1, \ldots, t_n]$ be an A-G rule. Obviously, R is circular if and only if some $x \in var(t')$ is not syntactically sound in R.*

The link between circularity of an A-G rule and cyclic dependencies of variables, which has been suggested intuitively by the above example, will be established by Proposition 3.60 below. For that, we need a formal definition of dependencies.

Definition 3.57 (Dependency Relation).
Given a set of formulas Φ and two variables x and y, we say that x *depends* on

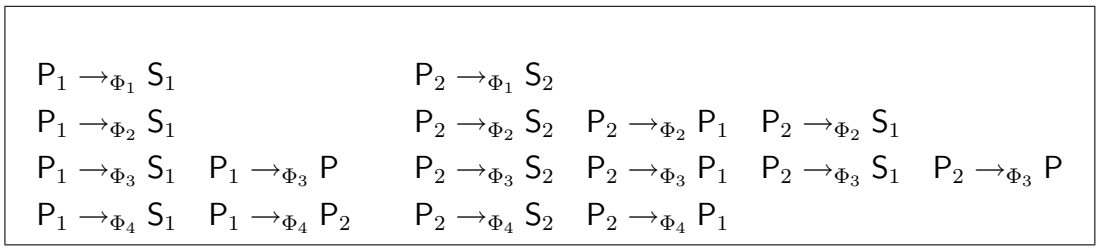

Figure 3.2: Dependency relations of the premises of the A-G rules R_1 to R_4

y in Φ, denoted by $x \to_\Phi y$, if and only if $x \neq y$ and there is a term t such that $\Phi \models y \sqcap t \sqsubseteq x$ and $\Phi \not\models t \sqsubseteq x$.

Notation. By $\to_\Phi^+$ we denote the *transitive closure* of $\to_\Phi$, i. e., $\to_\Phi^+$ is the least binary relation on $\mathcal{V}$ which is transitive and contains $\to_\Phi$. We say that the dependency relation $\to_\Phi$ is *cyclic* if and only if $x \to_\Phi^+ x$ for some variable x.

Example 3.58. Consider the A-G rules R_1 to R_4 from the example above (see figure 3.1) and denote their premises by Φ_1 to Φ_4, respectively. Figure 3.2 shows the corresponding dependency relations. Only $\to_{\Phi_4}$ is cyclic ($\mathsf{P}_1 \to_{\Phi_4} \mathsf{P}_2 \to_{\Phi_4} \mathsf{P}_1$ is a cycle of length 2), which corresponds to the fact that R_4 is the only circular rule.

Lemma 3.59. *Let $R : \Phi / t \sqsubseteq t'$ if $\Gamma[t_1, \ldots, t_n]$ be an A-G rule and let NSS be the set of variables of R which are not syntactically sound in R, i. e., $NSS = \{x \in var(R) \mid \Phi \not\models t \sqsubseteq x\}$. Then for every $x \in NSS$ there is $y \in NSS$ such that $x \to_\Phi y$.*

Proof. Let $x \in NSS$. Then $x \notin var(t)$, i. e., x is not a global assumption, so x must be a local guarantee since R is an A-G rule, i. e., x is a guarantee of some $\varphi \in \Phi$. As φ is in strong normal form, φ equals $t_\varphi \sqsubseteq x$ for some term t_φ with $x \notin var(t_\varphi)$. The term t_φ can be separated into two terms such that one contains exactly the syntactically sound variables, i. e., there are terms t'_φ and t''_φ with $var(t'_\varphi) \subseteq NSS$ and $var(t''_\varphi) \cap NSS = \emptyset$ such that $t_\varphi = t'_\varphi \sqcap t''_\varphi$.

Towards a contradiction, assume that $x \not\to_\Phi y$ for all $y \in NSS$, i. e., for all $y \in NSS$ and all $t \in \mathcal{T}$, if $x \neq y$ and $\Phi \models y \sqcap t \sqsubseteq x$ then $\Phi \models t \sqsubseteq x$. We know that $\Phi \models t'_\varphi \sqcap t''_\varphi \sqsubseteq x$ where $var(t'_\varphi) \subseteq NSS \setminus \{x\}$, so by induction on $size(t'_\varphi)$, we can eliminate t'_φ and get $\Phi \models t''_\varphi \sqsubseteq x$. As all variables in t''_φ are syntactically sound in R, we have $\Phi \models t \sqsubseteq t''_\varphi$, which implies $\Phi \models t \sqsubseteq x$. This contradicts $x \in NSS$. □

Proposition 3.60. *Let $R : \Phi / \psi$ if $\Gamma[t_1, \ldots, t_n]$ be an A-G rule. If R is circular then the dependency relation $\to_\Phi$ is cyclic.*

Proof. Let $NSS = \{x \in var(R) \mid \Phi \not\models t \sqsubseteq x\}$. If R is circular then $\Phi \not\models \psi$, so $NSS \neq \emptyset$. By Lemma 3.59, for every $x \in NSS$ there is $y \in NSS$ such that $x \to_\Phi y$. By finiteness of NSS, there must be $x \in NSS$ such that $x \to_\Phi^+ x$. □

As a consequence, a circular A-G rule R must have at least two distinct local guarantees, otherwise the dependency relation cannot be cyclic. By definition of A-G rules, R must also have at least one global assumption, so the corollary below follows.

Corollary 3.61. *In a circular A-G rule at least three distinct variables must occur.*

Example 3.62. The above proposition says that for circular A-G rules the dependency relation is necessarily cyclic. In general, the converse is not true, i. e., a cyclic dependency relation does not necessarily imply that the A-G rule is circular. Take for instance the syntactic A-G rule

$$R : \frac{\mathsf{P}_3 \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_1 \qquad \mathsf{P}_1 \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_2 \qquad \mathsf{P}_2 \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_3}{\mathsf{S} \sqsubseteq \top}$$

where S, P_1, P_2, P_3 are variables. This rule is syntactically sound in a trivial way, hence it cannot be circular. However, the dependency relation of its premises (which we denote by Φ) is cyclic, since we have $\mathsf{P}_i \to_\Phi \mathsf{P}_j$ for all $i, j \in \{1, 2, 3\}$ with $i \neq j$.

Chapter 4

The Framework

In this chapter, we investigate a number of circular assume-guarantee rules for topped meet-semilattices. The main challenge is to establish soundness of these circular rules. This is done inductively, which requires a notion of approximation on semilattice elements, and the approximants must be ordered in a well-founded way. We obtain the required notion of approximation by embedding the semilattice into a complete lattice, which is always possible, e. g., via completion with order ideals, so we work with complete lattices instead of topped meet-semilattices. We identify a join-dense subset of the lattice, so we can approximate all elements by joins over elements of that subset. Then soundness of a circular assume-guarantee rule is proven by induction on that join-dense subset, provided that the order on that subset is well-founded.

Our plan is as follows. First, we formally introduce well-founded approximations of a complete lattice, the key step to enable induction. Second, we define non-blockingness, a constraint that forms the side condition of our lattice-theoretic circular A-G rules. Then, we prove soundness and completeness of the basic circular A-G rule, which involves only three variables. We extend these results to circular A-G rules involving more variables, and we prove that our rules are actually the strongest fully sound A-G rules. Next, we observe that the rules presented so far are not compositional. We modify the rules in order to achieve compositionality and investigate whether soundness and completeness are preserved. Finally, we briefly examine whether the existence of well-founded approximations, which is fundamental to our proofs of soundness, is necessary for sound circular assume-guarantee reasoning.

Throughout this chapter we assume that the variable set $\mathcal{V}$ contains the distinct variables S_i, P_j, A_k and G_l for all $i, j, k, l \in \mathbb{N}$. We may omit the suffix 0 and write S (resp. P, A, G) to abbreviate S_0 (resp. P_0, A_0, G_0).

4.1 Well-founded Approximations

As already mentioned, the proofs of soundness of circular assume-guarantee rules will be by induction so we need a well-founded domain over which to induct. We could require that our semilattice is well-founded already but that is more than needed. In fact, it suffices to embed the semilattice into a complete lattice $\boldsymbol{\mathcal{L}}$ and identify a suborder $\boldsymbol{\mathcal{A}}$ of that lattice which is well-founded yet still allows to approximate all elements of $\boldsymbol{\mathcal{L}}$ as joins over elements of $\boldsymbol{\mathcal{A}}$.

Definition 4.1 (Well-founded Approximation).
Given a complete lattice $\boldsymbol{\mathcal{L}}$, a *well-founded approximation* (of $\boldsymbol{\mathcal{L}}$) is a suborder $\boldsymbol{\mathcal{A}}$ of $\boldsymbol{\mathcal{L}}$ such that $\mathcal{A}$ is join-dense in $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{\mathcal{A}}$ satisfies DCC. We say that $\boldsymbol{\mathcal{L}}$ is *well-approximable* if and only if there is a well-founded approximation of $\boldsymbol{\mathcal{L}}$.

Notation. Throughout this chapter, ordered sets will be denoted by calligraphic letters; in particular, $\boldsymbol{\mathcal{L}}$ will always denote a lattice. Elements of $\boldsymbol{\mathcal{L}}$ will be denoted by lower-case letters, where the letters a, b, c shall be reserved for approximants, i. e., for the elements of a join-dense suborder of $\boldsymbol{\mathcal{L}}$. Sets of elements of $\boldsymbol{\mathcal{L}}$ will be denoted by upper-case letters.

Obviously, every complete lattice satisfying DCC (and in particular every finite lattice) is well-approximable. The following example presents a well-approximable lattice which does not satisfy DCC.

Example 4.2. Let $\Sigma = \{0, 1\}$ be an alphabet and consider the lattice $\boldsymbol{\mathcal{L}} = \langle \mathcal{L}, \subseteq \rangle$ with $\mathcal{L} = \mathcal{O}(\langle \Sigma^*, \preceq \rangle)$, i. e., $\boldsymbol{\mathcal{L}}$ is the complete lattice of prefix-closed subsets of Σ^*, ordered by inclusion. Note that $\boldsymbol{\mathcal{L}}$ does not satisfy DCC, since for instance $\Sigma^* \supseteq 0\Sigma^* \supseteq 00\Sigma^* \supseteq \ldots$ is an infinite descending chain in $\boldsymbol{\mathcal{L}}$ which does not stabilize. However, the set of principal ideals $\{\preceq(w) \mid w \in \Sigma^*\}$ is join-dense in $\boldsymbol{\mathcal{L}}$ because for every $x \in \mathcal{O}(\langle \Sigma^*, \preceq \rangle)$, $x = \bigcup_{w \in x} \preceq(w)$. And as $\preceq(w)$ is finite for every $w \in \Sigma^*$, every infinite descending chain of principal ideals must stabilize eventually. Hence, the suborder of $\boldsymbol{\mathcal{L}}$ which is generated by the set of principal ideals is a well-founded approximation of $\boldsymbol{\mathcal{L}}$.

While every complete lattice $\boldsymbol{\mathcal{L}}$ has join-dense subsets — in fact, $\mathcal{L}$ itself is join-dense in $\boldsymbol{\mathcal{L}}$ — well-founded approximations need not exist.

Example 4.3. Consider $\langle \mathbb{Z}^\top_\bot, \leq \rangle$, the complete lattice which consists of the linearly ordered integers plus top and bottom elements. In $\langle \mathbb{Z}^\top_\bot, \leq \rangle$, every $z \in \mathbb{Z}$ is completely join-irreducible, hence any join-dense $\mathcal{A} \subseteq \mathbb{Z}^\top_\bot$ contains $\mathbb{Z}$ and thus the suborder $\langle \mathcal{A}, \preceq \rangle$ cannot satisfy DCC. Therefore, $\langle \mathbb{Z}^\top_\bot, \leq \rangle$ is not well-approximable.

However, it turns out that if a lattice has well-founded approximations then there is a canonical one; more precisely, in that case the completely join-irreducible elements generate the least well-founded approximation.

Lemma 4.4. *Let $\boldsymbol{\mathcal{L}} = \langle \mathcal{L}, \leq \rangle$ be a complete lattice and let $\boldsymbol{\mathcal{A}} = \langle \mathcal{A}, \preceq \rangle$ be a well-founded approximation of $\boldsymbol{\mathcal{L}}$. Then for all $a \in \mathcal{A}$ exists $J \subseteq \mathcal{J}(\boldsymbol{\mathcal{L}})$ such that $a = \bigvee J$.*

Proof. By well-founded induction over $\boldsymbol{\mathcal{A}}$. Let $a \in \mathcal{A}$ and assume that for every $b \in \prec(a)$ there is $J_b \subseteq \mathcal{J}(\boldsymbol{\mathcal{L}})$ such that $b = \bigvee J_b$. We have to show that $a = \bigvee J$ for some $J \subseteq \mathcal{J}(\boldsymbol{\mathcal{L}})$.

If $a \in \mathcal{J}(\boldsymbol{\mathcal{L}})$ then $a = \bigvee\{a\}$ and we are done, so assume that $a \notin \mathcal{J}(\boldsymbol{\mathcal{L}})$, hence by definition of complete join-irreducibility, there is $S \subseteq \mathcal{L}$ such that $a = \bigvee S$ but $a \notin S$, which implies $S \subseteq {<}(a)$. As $\mathcal{A}$ is join-dense, for every $x \in S$ there is $A_x \subseteq \mathcal{A}$ such that $x = \bigvee A_x$. Let $A = \bigcup_{x \in S} A_x$. Then $\bigvee A = \bigvee S$. Furthermore, $A \subseteq {<}(a)$ since for every $b \in A$ there is $x \in S$ such that $b \in A_x$, which implies $b \leq \bigvee A_x = x \in {<}(a)$. In fact, $A \subseteq \prec(a)$ as $A \subseteq \mathcal{A}$ and the order $\preceq$ on $\mathcal{A}$ is induced by $\leq$.

By induction hypothesis, for every $b \in A$ there is $J_b \subseteq \mathcal{J}(\boldsymbol{\mathcal{L}})$ such that $b = \bigvee J_b$. Let $J = \bigcup_{b \in A} J_b$, so $J \subseteq \mathcal{J}(\boldsymbol{\mathcal{L}})$. Furthermore $\bigvee J = \bigvee A$, hence we have $a = \bigvee S = \bigvee A = \bigvee J$. □

Proposition 4.5. *Let $\boldsymbol{\mathcal{L}}$ be a complete lattice and let $\boldsymbol{\mathcal{J}}$ be the suborder of $\boldsymbol{\mathcal{L}}$ which is generated by $\mathcal{J}(\boldsymbol{\mathcal{L}})$. The lattice $\boldsymbol{\mathcal{L}}$ is well-approximable if and only if $\boldsymbol{\mathcal{J}}$ is a well-founded approximation of $\boldsymbol{\mathcal{L}}$, and if $\boldsymbol{\mathcal{L}}$ is well-approximable then $\boldsymbol{\mathcal{J}}$ is the least well-founded approximation of $\boldsymbol{\mathcal{L}}$.*

Proof. The right-to-left direction of the first claim is trivial. To prove the other direction, let $\boldsymbol{\mathcal{L}} = \langle \mathcal{L}, \leq \rangle$ and let $\boldsymbol{\mathcal{A}}$ be a well-founded approximation of $\boldsymbol{\mathcal{L}}$. As $\mathcal{A}$ is join-dense in $\boldsymbol{\mathcal{L}}$, $\mathcal{J}(\boldsymbol{\mathcal{L}}) \subseteq \mathcal{A}$ follows by definition of complete join-irreducibility. This implies that $\boldsymbol{\mathcal{J}}$ satisfies DCC because $\boldsymbol{\mathcal{A}}$ does so. It remains to show that $\mathcal{J}(\boldsymbol{\mathcal{L}})$ is join-dense in $\boldsymbol{\mathcal{L}}$, so let $x \in \mathcal{L}$. As $\mathcal{A}$ is join-dense, there is $A \subseteq \mathcal{A}$ such that $x = \bigvee A$. By Lemma 4.4, for every $a \in A$ there exists $J_a \subseteq \mathcal{J}(\boldsymbol{\mathcal{L}})$ such that $a = \bigvee J_a$. Let $J = \bigcup_{a \in A} J_a$. Then $\bigvee J = \bigvee A$, so $x = \bigvee J$, and we are done because $J \subseteq \mathcal{J}(\boldsymbol{\mathcal{L}})$.

The second claim holds because by definition of complete join-irreducibility, $\mathcal{J}(\boldsymbol{\mathcal{L}})$ is contained in every subset of $\mathcal{L}$ which is join-dense in $\boldsymbol{\mathcal{L}}$, hence $\boldsymbol{\mathcal{J}}$ is a suborder of every well-founded approximation of $\boldsymbol{\mathcal{L}}$. □

As a consequence of the above proposition, a lattice may fail to be well-approximable because the completely join-irreducible elements are not join-dense, or because they generate a suborder that does not satisfy DCC. The following example shows three such scenarios.

Example 4.6. First, consider $\langle \mathbb{R}^{\top}_{\bot}, \leq \rangle$, the complete lattice which consists of the linearly ordered reals plus top and bottom elements. This lattice does not have any completely join-irreducible elements, and $\emptyset$ is not join-dense in $\langle \mathbb{R}^{\top}_{\bot}, \leq \rangle$ but does satisfy DCC trivially.

Second, take a look at $\langle \mathbb{Z}^\top_\bot, \leq \rangle$, the complete lattice of the linearly ordered integers with top and bottom. The set of completely join-irreducible elements in $\langle \mathbb{Z}^\top_\bot, \leq \rangle$ is $\mathbb{Z}$, which is join-dense in $\langle \mathbb{Z}^\top_\bot, \leq \rangle$ but does not satisfy DCC.

Third, construct a new lattice $\boldsymbol{\mathcal{L}}$ by placing $\langle \mathbb{R}^\top_\bot, \leq \rangle$ above $\langle \mathbb{Z}^\top_\bot, \leq \rangle$ and fusing the bottom of $\langle \mathbb{R}^\top_\bot, \leq \rangle$ with the top of $\langle \mathbb{Z}^\top_\bot, \leq \rangle$. Then $\boldsymbol{\mathcal{L}}$ is a linearly ordered complete lattice whose set of completely join-irreducible elements is $\mathbb{Z}$, which is neither join-dense in $\boldsymbol{\mathcal{L}}$ nor does it satisfy DCC.

4.2 Non-Blockingness

The key step in the inductive proofs of soundness will always require that certain elements of the lattice are not tied to each other too closely. We are going to specify what we mean by 'not being tied too closely' by defining the opposite, which we call 'blocking each other'.

Throughout the remainder of this chapter we fix a non-trivial complete lattice $\boldsymbol{\mathcal{L}} = \langle \mathcal{L}, \leq \rangle$ with least element 0 and greatest element 1.

Definition 4.7 (Non-Blockingness).
Given a suborder $\boldsymbol{\mathcal{A}} = \langle \mathcal{A}, \preceq \rangle$ of $\boldsymbol{\mathcal{L}}$, we say that two elements $p_1, p_2 \in \mathcal{L}$ *block* each other *relative* to an element $s \in \mathcal{L}$ (w. r. t. $\boldsymbol{\mathcal{A}}$) if and only if there exists $a \in \mathcal{A}$ such that $a \leq s$ and $p_1 \wedge p_2$ is an upper bound of $\prec(a)$ but $a \not\leq p_1$ and $a \not\leq p_2$. We define the ternary *non-blockingness* relation $\mathrm{NB}_{\boldsymbol{\mathcal{A}}}$ on $\mathcal{L}$ (w. r. t. $\boldsymbol{\mathcal{A}}$) by $\langle s, p_1, p_2 \rangle \in \mathrm{NB}_{\boldsymbol{\mathcal{A}}}$ if and only if p_1 and p_2 do not block each other relative to s (w. r. t. $\boldsymbol{\mathcal{A}}$).

Notation. We say that two elements $p_1, p_2 \in \mathcal{L}$ *block* each other (w. r. t. $\boldsymbol{\mathcal{A}}$) if and only if there exists $s \in \mathcal{L}$ such that p_1 and p_2 block each other relative to s. Note that p_1 and p_2 block each other if and only if they block each other relative to 1.

Observe that p_1 and p_2 block each other relative to s if there is a minimal $a \in \mathcal{A}$ such that $a \leq s$ but $a \not\leq p_1$ and $a \not\leq p_2$. The following proposition yields a characterization of non-blockingness.

Proposition 4.8. *Let $\boldsymbol{\mathcal{A}} = \langle \mathcal{A}, \preceq \rangle$ be a suborder of $\boldsymbol{\mathcal{L}}$ and let $s, p_1, p_2 \in \mathcal{L}$. Obviously, the following statements are equivalent:*

1. $\langle s, p_1, p_2 \rangle \in \mathrm{NB}_{\boldsymbol{\mathcal{A}}}$.
2. *For all $a \in \mathcal{A} \cap {\leq}(s)$, if $p_1 \wedge p_2$ is an upper bound of $\prec(a)$ then $a \leq p_1$ or $a \leq p_2$.*
3. *For all $a \in \mathcal{A}$, if $a \leq s$ and for all $b \in \mathcal{A}$, $b \prec a$ implies $b \leq p_1 \wedge p_2$ then $a \leq p_1$ or $a \leq p_2$.*

Example 4.9. Let Σ be a finite alphabet and let $\boldsymbol{\mathcal{L}}$ be the ideal completion of the prefix-ordered finite words over Σ, i. e., $\boldsymbol{\mathcal{L}}$ is the lattice of prefix-closed subsets of Σ^*, ordered by inclusion. As approximation suborder $\boldsymbol{\mathcal{A}}$, we choose the set of all prefix-closures of words in Σ^*, again ordered by inclusion. Let $s, p_1, p_2 \in \mathcal{L}$ be three regular languages, i. e., we may think of s, p_1 and p_2 as finite automata. If it is the case that p_1 and p_2 block each other relative to s then there exists a word $w \in \Sigma^*$ such that w is accepted by s and all proper prefixes of w are accepted by both p_1 and p_2 but neither p_1 nor p_2 accept w itself, i. e., p_1 and p_2 block after reading a proper prefix of w.

On the other hand, if $\langle s, p_1, p_2\rangle \in \mathrm{NB}_{\boldsymbol{\mathcal{A}}}$, i. e., p_1 and p_2 do not block each other relative to s, then every $w \in \Sigma^*$ which is accepted by s is also accepted by either p_1 or p_2 whenever all of its proper prefixes are accepted by both. Put another way: Given a word $uav \in s$ with $a \in \Sigma$ and $u, v \in \Sigma^*$, if u is accepted by both p_1 and p_2 then they cannot block both after reading u because one of them must accept ua. If both happen to accept ua and $v = bw$ with $b \in \Sigma$ and $w \in \Sigma^*$ then the game starts over again, i. e., either p_1 or p_2 has to accept uab, and so on.

By definition, non-blockingness depends on the choice of the approximation suborder $\boldsymbol{\mathcal{A}}$. We show, that there is a canonical approximation suborder if the completely join-irreducible elements are join-dense in $\boldsymbol{\mathcal{L}}$.

Lemma 4.10. *Let $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{J}}$ be suborders of $\boldsymbol{\mathcal{L}}$. If $\mathcal{J} \subseteq \mathcal{A}$ and $\mathcal{J}$ is join-dense in $\boldsymbol{\mathcal{L}}$ then $\mathrm{NB}_{\boldsymbol{\mathcal{A}}} \subseteq \mathrm{NB}_{\boldsymbol{\mathcal{J}}}$.*

Proof. Let $\boldsymbol{\mathcal{A}} = \langle \mathcal{A}, \preceq\rangle$ and $\boldsymbol{\mathcal{J}} = \langle \mathcal{J}, \trianglelefteq\rangle$ and assume $\mathcal{J} \subseteq \mathcal{A}$ and $\mathcal{J}$ being join-dense. Let $\langle s, p_1, p_2\rangle \in \mathrm{NB}_{\boldsymbol{\mathcal{A}}}$, and let $a \in \mathcal{J} \cap {\leq}(s)$ such that $p_1 \wedge p_2$ is an upper bound of $\lhd(a)$. We have to show $a \leq p_1$ or $a \leq p_2$.

As $\mathcal{J} \subseteq \mathcal{A}$, we have $a \in \mathcal{A} \cap {\leq}(s)$. And as $\mathcal{J}$ is join-dense, for every $b \in {\prec}(a)$ there is $J_b \subseteq \mathcal{J}$ such that $b = \bigvee J_b$. Let $J = \bigcup_{b \in \prec(a)} J_b$. Then $\bigvee J = \bigvee {\prec}(a)$. Furthermore, $J \subseteq {<}(a)$ because for every $b \in {\prec}(a)$, we have $\bigvee J_b = b \prec a$, which implies $J_b \subseteq {\prec}(a) \subseteq {<}(a)$. Actually, $J \subseteq \lhd(a)$ because $a \in \mathcal{J}$ and $J \subseteq \mathcal{J}$ and the order $\trianglelefteq$ on $\mathcal{J}$ is induced by $\leq$. Therefore, we have $\bigvee {\prec}(a) = \bigvee J \leq \bigvee \lhd(a)$, so $p_1 \wedge p_2$, which is an upper bound of $\lhd(a)$, must also be an upper bound of ${\prec}(a)$. By Proposition 4.8, we conclude $a \leq p_1$ or $a \leq p_2$ from $\langle s, p_1, p_2\rangle \in \mathrm{NB}_{\boldsymbol{\mathcal{A}}}$. □

Lemma 4.11. *Let $\boldsymbol{\mathcal{A}}$ be a suborder of $\boldsymbol{\mathcal{L}}$ and let $\boldsymbol{\mathcal{J}}$ be the suborder of $\boldsymbol{\mathcal{L}}$ which is generated by $\mathcal{J}(\boldsymbol{\mathcal{L}})$. If $\mathcal{A}$ is join-dense in $\boldsymbol{\mathcal{L}}$ then $\mathrm{NB}_{\boldsymbol{\mathcal{J}}} \subseteq \mathrm{NB}_{\boldsymbol{\mathcal{A}}}$.*

Proof. Let $\boldsymbol{\mathcal{A}} = \langle \mathcal{A}, \preceq\rangle$ and $\boldsymbol{\mathcal{J}} = \langle \mathcal{J}, \trianglelefteq\rangle$ where $\mathcal{J} = \mathcal{J}(\boldsymbol{\mathcal{L}})$ is the set of completely join-irreducible elements in $\boldsymbol{\mathcal{L}}$. Assume that $\mathcal{A}$ is join-dense and note that this implies $\mathcal{J} \subseteq \mathcal{A}$. Let $\langle s, p_1, p_2\rangle \in \mathrm{NB}_{\boldsymbol{\mathcal{J}}}$, and let $a \in \mathcal{A} \cap {\leq}(s)$ such that $p_1 \wedge p_2$ is an upper bound of ${\prec}(a)$. We have to show $a \leq p_1$ or $a \leq p_2$. We distinguish two cases.

First assume $a \in \mathcal{J}$. Then $a \in \mathcal{J} \cap {\leq}(s)$. Furthermore, $\mathcal{J} \subseteq \mathcal{A}$ implies ${\lhd}(a) \subseteq {\prec}(a)$, so $p_1 \wedge p_2$, which is an upper bound of ${\prec}(a)$, must also be an upper bound of ${\lhd}(a)$. Hence by Proposition 4.8, we conclude $a \leq p_1$ or $a \leq p_2$ from $\langle s, p_1, p_2 \rangle \in \mathrm{NB}_{\boldsymbol{\mathcal{J}}}$.

Otherwise assume $a \notin \mathcal{J}$. By definition of complete join-irreducibility, there is $S \subseteq \mathcal{L}$ such that $a = \bigvee S$ but $a \notin S$, which implies $S \subseteq {<}(a)$. As $\mathcal{A}$ is join-dense, for every $x \in S$ there is $A_x \subseteq \mathcal{A}$ such that $x = \bigvee A_x$. Let $A = \bigcup_{x \in S} A_x$. Then $\bigvee A = \bigvee S$. Furthermore, for every $x \in S$, $A_x \subseteq {<}(a)$ because $\bigvee A_x = x < a$, and therefore $A \subseteq {<}(a)$. Actually, $A \subseteq {\prec}(a)$ because $a \in \mathcal{A}$ and $A \subseteq \mathcal{A}$ and the order $\preceq$ on $\mathcal{A}$ is induced by $\leq$. So finally, $a = \bigvee S = \bigvee A \leq \bigvee {\prec}(a) \leq a$, which implies $a = \bigvee {\prec}(a)$. Since $p_1 \wedge p_2$ is an upper bound of ${\prec}(a)$, we have $a \leq p_1 \wedge p_2$, which implies $a \leq p_1$. □

Notation. Let $\boldsymbol{\mathcal{J}}$ be the suborder of $\boldsymbol{\mathcal{L}}$ which is generated by $\mathcal{J}(\boldsymbol{\mathcal{L}})$. We may omit the subscript on the non-blockingness relation w. r. t. $\boldsymbol{\mathcal{J}}$, i. e., instead of $\mathrm{NB}_{\boldsymbol{\mathcal{J}}}$ we may write NB.

Proposition 4.12. *Let $\mathcal{J}(\boldsymbol{\mathcal{L}})$ be join-dense in $\boldsymbol{\mathcal{L}}$ and let $\boldsymbol{\mathcal{A}}$ be a suborder of $\boldsymbol{\mathcal{L}}$ such that $\mathcal{A}$ is join-dense in $\boldsymbol{\mathcal{L}}$. Then* $\mathrm{NB} = \mathrm{NB}_{\boldsymbol{\mathcal{A}}}$.

Proof. Let $\boldsymbol{\mathcal{J}}$ be the suborder of $\boldsymbol{\mathcal{L}}$ which is generated by $\mathcal{J}(\boldsymbol{\mathcal{L}})$, i. e., $\mathcal{J} = \mathcal{J}(\boldsymbol{\mathcal{L}})$. By Lemma 4.11, we have $\mathrm{NB}_{\boldsymbol{\mathcal{J}}} \subseteq \mathrm{NB}_{\boldsymbol{\mathcal{A}}}$. And as $\mathcal{A}$ being join-dense implies $\mathcal{J} \subseteq \mathcal{A}$ by definition of complete join-irreducibility, Lemma 4.10 yields $\mathrm{NB}_{\boldsymbol{\mathcal{A}}} \subseteq \mathrm{NB}_{\boldsymbol{\mathcal{J}}}$. Hence $\mathrm{NB} = \mathrm{NB}_{\boldsymbol{\mathcal{J}}} = \mathrm{NB}_{\boldsymbol{\mathcal{A}}}$. □

Following are a couple of monotonicity results, which relate non-blockingness to the order of the lattice $\boldsymbol{\mathcal{L}}$.

Proposition 4.13. *Let $s, s', p_1, p_2 \in \mathcal{L}$ such that $s' \leq s$. Then $\langle s, p_1, p_2 \rangle \in \mathrm{NB}$ implies $\langle s', p_1, p_2 \rangle \in \mathrm{NB}$.*

Proof. This follows immediately from Proposition 4.8 since ${\leq}(s') \subseteq {\leq}(s)$. □

Proposition 4.14. *Let $s, p, p_1, \ldots, p_n \in \mathcal{L}$. If $\{\langle s, p_1, p \rangle, \ldots, \langle s, p_n, p \rangle\} \subseteq \mathrm{NB}$ then $\langle s, p_1 \wedge \ldots \wedge p_n, p \rangle \in \mathrm{NB}$.*

Proof. Let $\preceq$ be the order on $\mathcal{J}(\boldsymbol{\mathcal{L}})$ which is induced by $\leq$. The proof proceeds by induction on n.

- $n = 0$. Then $p_1 \wedge \ldots \wedge p_n = 1$, so we have $\langle s, p_1 \wedge \ldots \wedge p_n, p \rangle \in \mathrm{NB}$ trivially by Proposition 4.8.

- $n > 0$. Assume $\{\langle s, p_1, p \rangle, \ldots, \langle s, p_n, p \rangle\} \subseteq \mathrm{NB}$ and let $a \in \mathcal{J}(\boldsymbol{\mathcal{L}}) \cap {\leq}(s)$ such that $(p_1 \wedge \ldots \wedge p_n) \wedge p$ is an upper bound of ${\prec}(a)$. If $a \leq p$ then we are done, so assume $a \not\leq p$. As $\{\langle s, p_1, p \rangle, \ldots, \langle s, p_{n-1}, p \rangle\} \subseteq \mathrm{NB}$ implies $\langle s, p_1 \wedge \ldots \wedge p_{n-1}, p \rangle \in \mathrm{NB}$ by induction hypothesis, $a \leq p_1 \wedge \ldots \wedge p_{n-1}$ follows from $a \not\leq p$. And due to $\langle s, p_n, p \rangle \in \mathrm{NB}$, $a \leq p_n$ also follows from $a \not\leq p$. Hence $a \leq p_1 \wedge \ldots \wedge p_n$ and we are done. □

Proposition 4.15. *Let $s, p_1, p'_1, \ldots, p_n, p'_n \in \mathcal{L}$. If for every $i \in \{1, \ldots, n\}$ we have $p_i \wedge s \leq p'_i$ and $\langle s, p_i, p'_i \rangle \in \mathrm{NB}$ then $\langle s, p_1 \wedge \ldots \wedge p_n, p'_1 \wedge \ldots \wedge p'_n \rangle \in \mathrm{NB}$.*

Proof. Let $\preceq$ be the order on $\mathcal{J}(\boldsymbol{\mathcal{L}})$ which is induced by $\leq$. The proof proceeds by induction on n.

- $n = 0$. Then $p_1 \wedge \ldots \wedge p_n = p'_1 \wedge \ldots \wedge p'_n = 1$, so by Proposition 4.8 we have $\langle s, p_1 \wedge \ldots \wedge p_n, p'_1 \wedge \ldots \wedge p'_n \rangle \in \mathrm{NB}$ trivially.
- $n > 0$. Assume $p_i \wedge s \leq p'_i$ and $\langle s, p_i, p'_i \rangle \in \mathrm{NB}$ for every $i \in \{1, \ldots, n\}$. Let $a \in \mathcal{J}(\boldsymbol{\mathcal{L}}) \cap {\leq}(s)$ such that $(p_1 \wedge \ldots \wedge p_n) \wedge (p'_1 \wedge \ldots \wedge p'_n)$ is an upper bound of ${\prec}(a)$. By induction hypothesis, $\langle s, p_1 \wedge \ldots \wedge p_{n-1}, p'_1 \wedge \ldots \wedge p'_{n-1} \rangle \in \mathrm{NB}$, so $a \leq p_1 \wedge \ldots \wedge p_{n-1}$ or $a \leq p'_1 \wedge \ldots \wedge p'_{n-1}$ In fact, $a \leq p'_1 \wedge \ldots \wedge p'_{n-1}$ as $a \leq s$ and $(p_1 \wedge \ldots \wedge p_{n-1}) \wedge s \leq p'_1 \wedge \ldots \wedge p'_{n-1}$ follows from the assumptions. Similarly, $\langle s, p_n, p'_n \rangle \in \mathrm{NB}$ implies $a \leq p_n$ or $a \leq p'_n$, which in turn implies $a \leq p'_n$ due to the assumption $p_n \wedge s \leq p'_n$. Hence $a \leq p'_1 \wedge \ldots \wedge p'_n$ and we are done. □

As a final remark, the following example shows that non-blockingness is undecidable in general, even in very simple well-approximable lattices.

Example 4.16. Let $\boldsymbol{\mathcal{L}} = \langle \mathcal{P}(\mathbb{N}), \subseteq \rangle$ be the power set lattice of the natural numbers. Obviously, $\boldsymbol{\mathcal{L}}$ is well-approximable as $\mathcal{J}(\boldsymbol{\mathcal{L}}) = \big\{\{n\} \mid n \in \mathbb{N}\big\}$, i. e., exactly the singleton sets are completely join-irreducible in $\boldsymbol{\mathcal{L}}$. According to Proposition 4.8, as all elements in $\mathcal{J}(\boldsymbol{\mathcal{L}})$ are minimal we can simplify the definition of the ternary relation NB to for all $s, p_1, p_2 \in \mathcal{P}(\mathbb{N})$, $\langle s, p_1, p_2 \rangle \in \mathrm{NB}$ iff $s \subseteq p_1 \cup p_2$. Still, NB is undecidable. Otherwise, we would have an algorithm which, given a (representation of a) recursive set $p \in \mathcal{P}(\mathbb{N})$, decides whether $\langle \mathbb{N}, p, p \rangle \in \mathrm{NB}$, i. e., whether $\mathbb{N} = p$. It is well-known that such an algorithm cannot exist, see for instance [Rog87].

4.3 The Basic Circular Assume-Guarantee Rule

In this section, we present a simple circular assume-guarantee rule for the complete lattice $\boldsymbol{\mathcal{L}}$. The rule is proven sound (provided that $\boldsymbol{\mathcal{L}}$ is well-approximable) and complete. Why we consider this rule to be *the* basic circular assume-guarantee rule will be revealed at the end of Section 4.4.

Definition 4.17 (Circular Assume-Guarantee Rule AG_2).

$$\mathrm{AG}_2 : \frac{\mathsf{P}_2 \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_1 \qquad \mathsf{P}_1 \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_2}{\mathsf{S} \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2} \text{ if } \mathrm{NB}[\mathsf{S}, \mathsf{P}_1, \mathsf{P}_2]$$

To attribute meaning to the inference rule AG_2, one should think of it as a proof rule used in verification. The variable S represents the system to be verified

and P_1 and P_2 represent two properties. In this context, AG_2 states that the system guarantees both properties (conclusion $\mathsf{S} \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2$) if it guarantees either assuming the other (premises $\mathsf{P}_2 \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_1$ and $\mathsf{P}_1 \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_2$), provided that the properties do not block each other relative to the system (side condition $\mathrm{NB}[\mathsf{S}, \mathsf{P}_1, \mathsf{P}_2]$).

It is rather obvious that — as the above definition claims — AG_2 is a circular assume-guarantee rule. However, it is not obvious that AG_2 is sound. Our proof of soundness, whose main part is the following inductive lemma, requires the lattice $\boldsymbol{\mathcal{L}}$ to be well-approximable. This seems to be a crucial requirement; in fact, Section 4.7 suggests that AG_2 is very likely to become unsound when $\boldsymbol{\mathcal{L}}$ fails to be well-approximable.

Lemma 4.18. *Let $\boldsymbol{\mathcal{A}}$ be a well-founded approximation of $\boldsymbol{\mathcal{L}}$, let $\langle s, p_1, p_2\rangle \in \mathrm{NB}_{\boldsymbol{\mathcal{A}}}$ such that $p_2 \wedge s \leq p_1$ and $p_1 \wedge s \leq p_2$. Then for all $a \in \mathcal{A}$, $a \leq s$ implies $a \leq p_1 \wedge p_2$.*

Proof. By well-founded induction over $\boldsymbol{\mathcal{A}} = \langle \mathcal{A}, \preceq\rangle$. Let $a \in \mathcal{A}$ and assume that for every $b \in \prec(a)$, $b \leq s$ implies $b \leq p_1 \wedge p_2$. We have to show that $a \leq s$ implies $a \leq p_1 \wedge p_2$.

Assume $a \leq s$. Then $b \leq s$ for every $b \in \prec(a)$ because the order $\preceq$ on $\mathcal{A}$ is induced by $\leq$. Thus by induction hypothesis, $b \leq p_1 \wedge p_2$ for every $b \in \prec(a)$, i. e., $p_1 \wedge p_2$ is an upper bound of $\prec(a)$. Because of $\langle s, p_1, p_2\rangle \in \mathrm{NB}_{\boldsymbol{\mathcal{A}}}$, we obtain $a \leq p_1$ or $a \leq p_2$ by Proposition 4.8. Together with the assumption $a \leq s$, we get $a \leq p_1 \wedge s$ or $a \leq p_2 \wedge s$. Case distinction:

- If $a \leq p_1 \wedge s$ then $a \leq p_2$ follows from $p_1 \wedge s \leq p_2$.
- If $a \leq p_2 \wedge s$ then $a \leq p_1$ follows from $p_2 \wedge s \leq p_1$.

In any case, we are done because we have $a \leq p_1 \wedge p_2$. □

Theorem 4.19. *If $\boldsymbol{\mathcal{L}}$ is well-approximable then AG_2 is sound.*

Proof. We use Proposition 3.40 to show soundness. Let α be a valuation with $dom(\alpha) = \{\mathsf{S}, \mathsf{P}_1, \mathsf{P}_2\}$ such that $\boldsymbol{\mathcal{L}}, \alpha \models \mathsf{P}_2 \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_1, \mathsf{P}_1 \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_2, \mathrm{NB}[\mathsf{S}, \mathsf{P}_1, \mathsf{P}_2]$. We have to show $\boldsymbol{\mathcal{L}}, \alpha \models \mathsf{S} \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2$.

Assume that $\boldsymbol{\mathcal{L}}$ is well-approximable, so by Proposition 4.5 $\boldsymbol{\mathcal{J}}$, the suborder of $\boldsymbol{\mathcal{L}}$ which is generated by the completely join-irreducible elements $\mathcal{J}(\boldsymbol{\mathcal{L}})$, is a well-founded approximation of $\boldsymbol{\mathcal{L}}$. Therefore, there is a set of approximants $A \subseteq \mathcal{J}(\boldsymbol{\mathcal{L}})$ such that $\alpha(\mathsf{S}) = \bigvee A$. Furthermore, by choice of the valuation α and with the definition of the non-blockingness relation NB, we have $\alpha(\mathsf{P}_2) \wedge \alpha(\mathsf{S}) \leq \alpha(\mathsf{P}_1)$ and $\alpha(\mathsf{P}_1) \wedge \alpha(\mathsf{S}) \leq \alpha(\mathsf{P}_2)$ and $\langle \alpha(\mathsf{S}), \alpha(\mathsf{P}_1), \alpha(\mathsf{P}_2)\rangle \in \mathrm{NB}_{\boldsymbol{\mathcal{J}}}$. So, Lemma 4.18 yields $a \leq \alpha(\mathsf{P}_1) \wedge \alpha(\mathsf{P}_2)$ for all $a \in A$, i. e., $\alpha(\mathsf{P}_1) \wedge \alpha(\mathsf{P}_2)$ is an upper bound of A. Thus $\alpha(\mathsf{S}) \leq \alpha(\mathsf{P}_1) \wedge \alpha(\mathsf{P}_2)$ since $\alpha(\mathsf{S})$ is the least upper bound of A. Hence we have $\boldsymbol{\mathcal{L}}, \alpha \models \mathsf{S} \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2$. □

It turns out that AG_2 is not only sound but it is complete, too. Surprisingly (and unlike the proof of soundness), the proof of completeness is rather trivial. Also, it does not demand extra requirements of the lattice $\boldsymbol{\mathcal{L}}$; in particular, $\boldsymbol{\mathcal{L}}$ need not be well-approximable.

Theorem 4.20. AG_2 *is both backward and forward complete.*

Proof. To show backward completeness of AG_2, choose a valuation α such that $dom(\alpha) = \{\mathsf{S}, \mathsf{P}_1, \mathsf{P}_2\}$ and $\boldsymbol{\mathcal{L}}, \alpha \models \mathsf{S} \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2$. From this, we have to infer $\boldsymbol{\mathcal{L}}, \alpha \models \mathsf{P}_2 \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_1$, $\mathsf{P}_1 \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_2$, $\mathrm{NB}[\mathsf{S}, \mathsf{P}_1, \mathsf{P}_2]$.

Let $a \in \mathcal{J}(\boldsymbol{\mathcal{L}})$. As $\boldsymbol{\mathcal{L}}, \alpha \models \mathsf{S} \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2$, $a \leq \alpha(\mathsf{S})$ implies $a \leq \alpha(\mathsf{P}_1) \wedge \alpha(\mathsf{P}_2)$, so in particular $a \leq \alpha(\mathsf{S})$ implies $a \leq \alpha(\mathsf{P}_1)$. By Proposition 4.8 and the definition of NB, we get $\langle \alpha(\mathsf{S}), \alpha(\mathsf{P}_1), \alpha(\mathsf{P}_2) \rangle \in \mathrm{NB}$, i. e., $\boldsymbol{\mathcal{L}}, \alpha \models \mathrm{NB}[\mathsf{S}, \mathsf{P}_1, \mathsf{P}_2]$. The rest follows since $\{\mathsf{S} \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2\} \models \{\mathsf{P}_2 \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_1, \mathsf{P}_1 \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_2\}$.

Forward completeness trivially follows by Proposition 3.52. □

Corollary 4.21. *If* $\boldsymbol{\mathcal{L}}$ *is well-approximable then*

$$\mathsf{P}_2 \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_1,\ \mathsf{P}_1 \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_2,\ \mathrm{NB}[\mathsf{S}, \mathsf{P}_1, \mathsf{P}_2] \equiv^{\boldsymbol{\mathcal{L}}} \mathsf{S} \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2.$$

Proof. Follows from soundness and (backward) completeness of AG_2. □

Example 4.22. To show that the side condition $\mathrm{NB}[\mathsf{S}, \mathsf{P}_1, \mathsf{P}_2]$ is not complete in AG_2, consider the finite (and thus complete and well-approximable) lattice $\boldsymbol{\mathcal{L}} = \langle \mathcal{P}(\{1, 2, 3, 4\}), \subseteq \rangle$. We define the valuation α with $dom(\alpha) = \{\mathsf{S}, \mathsf{P}_1, \mathsf{P}_2\}$ by $\alpha(\mathsf{S}) = \{1, 3, 4\}$, $\alpha(\mathsf{P}_1) = \{1, 2\}$ and $\alpha(\mathsf{P}_2) = \{2, 3\}$. As the premises of AG_2 are not true under alpha, we get $\mathsf{P}_2 \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_1$, $\mathsf{P}_1 \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_2 \models^{\boldsymbol{\mathcal{L}}}_{\alpha} \mathsf{S} \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2$ trivially. On the other hand, we have $\boldsymbol{\mathcal{L}}, \alpha \not\models \mathrm{NB}[\mathsf{S}, \mathsf{P}_1, \mathsf{P}_2]$ i. e., $\{1, 2\}$ and $\{2, 3\}$ block each other relative to $\{1, 3, 4\}$. This is so because $\mathcal{J}(\boldsymbol{\mathcal{L}}) = \big\{\{1\}, \{2\}, \{3\}, \{4\}\big\}$ and for $\{4\} \in \mathcal{J}(\boldsymbol{\mathcal{L}}) \cap {\subseteq}(\{1, 3, 4\})$, we have $\{4\} \not\subseteq \{1, 2\}$ and $\{4\} \not\subseteq \{2, 3\}$ although $\{4\}$ is minimal in $\mathcal{J}(\boldsymbol{\mathcal{L}})$.

Together with Theorem 4.20, the above example confirms a statement from Section 3.5, namely that completeness in the side condition is entailed neither by backward nor forward completeness.

4.4 Extension To More Than Two Properties

When verifying (through backward reasoning) that a given system guarantees a given property, the inference rule AG_2 enables us to reduce the goal into two subgoals, provided that the given property can be cast as a conjunction of two other properties that do not block each other relative to the given system. There is hope that each of the two subgoals, which are of the form *assuming one property, the system guarantees the other*, are easier to establish than the original goal

since the properties to be guaranteed are simpler. In general, it seems desirable to cast the original property as a conjunction of as many properties as possible, as with increasing number of conjuncts the individual properties should become all the simpler. Hence the n subgoals, which generally take the form *assuming all but the i-th property, the system guarantees the i-th property*, should become all the easier to establish. With the inference rule AG_2, however, we are limited to decompositions into exactly two conjuncts.

In this section, we generalize the circular A-G rule AG_2 to a family of A-G rules $\{\mathrm{AG}_n \mid n \in \mathbb{N}\}$, where AG_n is suitable for verification if the given property can be cast as a conjunction of n other properties such that these properties do not block each other pairwise relative to the given system. We prove soundness (provided that $\boldsymbol{\mathcal{L}}$ is well-approximable) and completeness of every such A-G rule. Furthermore, we show that the family $\{\mathrm{AG}_n \mid n \in \mathbb{N}\}$ can be characterized as the set of strongest rules in an important class of sound A-G rules.

Notation. Given $m, n \in \mathbb{N}$, we may abbreviate the term $\mathsf{P}_m \sqcap \mathsf{P}_{m+1} \sqcap \ldots \sqcap \mathsf{P}_{n-1} \sqcap \mathsf{P}_n$ by $\bigsqcap_i^{m,n} \mathsf{P}_i$. If the superscript m is omitted, we assume 1 as the lower bound, i. e., $\bigsqcap_i^{n} \mathsf{P}_i$ stands for $\bigsqcap_i^{1,n} \mathsf{P}_i$. Note that $\bigsqcap_i^{m,n} \mathsf{P}_i$ actually denotes $\top$ when $n < m$. The index variable i may additionally be constrained, i. e., given $k \in \mathbb{N}$, we may write $\bigsqcap_{i \neq k}^{m,n} \mathsf{P}_i$ to denote the term $\bigsqcap_i^{m,n} \mathsf{P}_i - \mathsf{P}_k$, i. e., $\mathsf{P}_m \sqcap \ldots \sqcap \mathsf{P}_{k-1} \sqcap \mathsf{P}_{k+1} \sqcap \ldots \sqcap \mathsf{P}_n$ if $m \leq k \leq n$, and $\mathsf{P}_m \sqcap \ldots \sqcap \mathsf{P}_n$ otherwise. Accordingly, by the same conventions we define the abbreviations $\bigsqcap_i^{m,n} \mathsf{S}_i$, $\bigsqcap_i^{m,n} \mathsf{A}_i$ and $\bigsqcap_i^{m,n} \mathsf{G}_i$. Also, we may use j instead of i as index variable.

Definition 4.23 (Assume-Guarantee Rule AG_n).
Given $n \in \mathbb{N}$, we define the inference rule

$$\mathrm{AG}_n : \frac{\bigsqcap_{j \neq 1}^{n} \mathsf{P}_j \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_1 \quad \ldots \quad \bigsqcap_{j \neq n}^{n} \mathsf{P}_j \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_n}{\mathsf{S} \sqsubseteq \bigsqcap_i^{n} \mathsf{P}_i} \text{ if } \bigwedge_{1 \leq i < j \leq n} \mathrm{NB}[\mathsf{S}, \mathsf{P}_i, \mathsf{P}_j].$$

Obviously for all $n \in \mathbb{N}$, the inference rule AG_n is an assume-guarantee rule. Note that for $n = 2$, AG_n coincides with AG_2 as defined in Section 4.3. In general for $n \in \mathbb{N}$, the side condition of AG_n is a conjunction of $n(n-1)/2$ non-blockingness relations, which implies that AG_n is syntactic if $n \leq 1$. More precisely, for $n = 0$ resp. $n = 1$ the above definition yields the syntactic A-G rules $\mathrm{AG}_0 : \emptyset / \mathsf{S} \sqsubseteq \top$ resp. $\mathrm{AG}_1 : \{\mathsf{S} \sqsubseteq \mathsf{P}_1\} / \mathsf{S} \sqsubseteq \mathsf{P}_1$. Obviously, AG_0 and AG_1 are syntactically sound.

Proposition 4.24. *The A-G rule AG_n is circular if and only if $n \geq 2$.*

Proof. Obviously, AG_0 and AG_1 are not circular, so assume $n \geq 2$. We define the valuation α with $dom(\alpha) = \{\mathsf{S}, \mathsf{P}_1, \ldots, \mathsf{P}_n\}$, where $\alpha(\mathsf{S}) = 1$ and $\alpha(\mathsf{P}_1) = \cdots = \alpha(\mathsf{P}_n) = 0$. Then we have $\boldsymbol{\mathcal{L}}, \alpha \models \bigsqcap_{j \neq i}^{n} \mathsf{P}_j \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_i$ for all $i \in \{1, \ldots, n\}$, but $\boldsymbol{\mathcal{L}}, \alpha \not\models \mathsf{S} \sqsubseteq \bigsqcap_i^{n} \mathsf{P}_i$. Therefore, the premises of AG_n do not entail the conclusion, i. e., AG_n is circular. □

Soundness of a circular A-G rule AG_n with $n > 2$ can be reduced to soundness of AG_2. We sketch how for $n = 3$. Let α be a valuation with $dom(\alpha) = var(\mathrm{AG}_3)$ such that the premises and the side condition of AG_3 are true under α. Note that truth of the side condition in particular implies $\boldsymbol{\mathcal{L}}, \alpha \models \mathrm{NB}[\mathsf{P}_3 \sqcap \mathsf{S}, \mathsf{P}_1, \mathsf{P}_2]$ and $\boldsymbol{\mathcal{L}}, \alpha \models \mathrm{NB}[\mathsf{S}, \mathsf{P}_1 \sqcap \mathsf{P}_2, \mathsf{P}_3]$, cf. the propositions 4.13 and 4.14. Therefore, we have $\boldsymbol{\mathcal{L}}, \alpha \models \mathsf{P}_2 \sqcap (\mathsf{P}_3 \sqcap \mathsf{S}) \sqsubseteq \mathsf{P}_1$, $\mathsf{P}_1 \sqcap (\mathsf{P}_3 \sqcap \mathsf{S}) \sqsubseteq \mathsf{P}_2$, $\mathrm{NB}[\mathsf{P}_3 \sqcap \mathsf{S}, \mathsf{P}_1, \mathsf{P}_2]$, which implies $\boldsymbol{\mathcal{L}}, \alpha \models \mathsf{P}_3 \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2$ by soundness of AG_2. This leaves us in the situation that $\boldsymbol{\mathcal{L}}, \alpha \models \mathsf{P}_3 \sqcap \mathsf{S} \sqsubseteq (\mathsf{P}_1 \sqcap \mathsf{P}_2)$, $(\mathsf{P}_1 \sqcap \mathsf{P}_2) \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_3$, $\mathrm{NB}[\mathsf{S}, \mathsf{P}_1 \sqcap \mathsf{P}_2, \mathsf{P}_3]$, which, again by soundness of AG_2, implies $\boldsymbol{\mathcal{L}}, \alpha \models \mathsf{S} \sqsubseteq (\mathsf{P}_1 \sqcap \mathsf{P}_2) \sqcap \mathsf{P}_3$. Thus, applying AG_2 twice proves soundness of AG_n for $n = 3$. The proof of the following theorem generalizes this idea to arbitrary n.

Theorem 4.25. *If $\boldsymbol{\mathcal{L}}$ is well-approximable then AG_n is sound for all $n \in \mathbb{N}$.*

Proof. Assume that $\boldsymbol{\mathcal{L}}$ is well-approximable. The proof is by induction on n.

- $n \leq 1$. Then AG_n is syntactically sound, trivially.

- $n > 1$. We assume soundness of AG_{n-1} and want to show soundness of AG_n using Proposition 3.40. Let α be a valuation with $dom(\alpha) = \{\mathsf{S}, \mathsf{P}_1, \ldots, \mathsf{P}_n\}$ such that for all $i \in \{1, \ldots, n\}$, we have $\boldsymbol{\mathcal{L}}, \alpha \models \bigsqcap_{j \neq i}^{n} \mathsf{P}_j \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_i$, and for all $i, j \in \{1, \ldots, n\}$ with $i < j$, we have $\boldsymbol{\mathcal{L}}, \alpha \models \mathrm{NB}[\mathsf{S}, \mathsf{P}_i, \mathsf{P}_j]$. We have to show $\boldsymbol{\mathcal{L}}, \alpha \models \mathsf{S} \sqsubseteq \bigsqcap_i^n \mathsf{P}_i$.

  We define the valuation β with $dom(\beta) = \{\mathsf{S}, \mathsf{P}_1, \ldots, \mathsf{P}_{n-1}\}$ such that $\beta(\mathsf{P}_i) = \alpha(\mathsf{P}_i)$ for all $i \in \{1, \ldots, n-1\}$ and $\beta(\mathsf{S}) = \alpha(\mathsf{P}_n \sqcap \mathsf{S})$. Obviously, we have $\boldsymbol{\mathcal{L}}, \beta \models \bigsqcap_{j \neq i}^{n-1} \mathsf{P}_j \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_i$ for all $i \in \{1, \ldots, n-1\}$. And as $\beta(\mathsf{S}) \leq \alpha(\mathsf{S})$, for all $i, j \in \{1, \ldots, n\}$ with $i < j$, we get $\boldsymbol{\mathcal{L}}, \beta \models \mathrm{NB}[\mathsf{S}, \mathsf{P}_i, \mathsf{P}_j]$ by Proposition 4.13. By induction hypothesis, the rule AG_{n-1} is sound, so we infer $\boldsymbol{\mathcal{L}}, \beta \models \mathsf{S} \sqsubseteq \bigsqcap_i^{n-1} \mathsf{P}_i$, i. e., $\boldsymbol{\mathcal{L}}, \alpha \models \mathsf{P}_n \sqcap \mathsf{S} \sqsubseteq \bigsqcap_i^{n-1} \mathsf{P}_i$.

  Next, we define the valuation γ with $dom(\gamma) = \{\mathsf{S}, \mathsf{P}_1, \mathsf{P}_2\}$ such that $\gamma(\mathsf{S}) = \alpha(\mathsf{S})$ and $\gamma(\mathsf{P}_1) = \alpha(\bigsqcap_i^{n-1} \mathsf{P}_i)$ and $\gamma(\mathsf{P}_2) = \alpha(\mathsf{P}_n)$. Then we have $\boldsymbol{\mathcal{L}}, \gamma \models \mathsf{P}_2 \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_1$ because of the above application of the induction hypothesis, and $\boldsymbol{\mathcal{L}}, \gamma \models \mathsf{P}_1 \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_2$ because of the truth of the last premise of AG_n under α. Furthermore, we have $\boldsymbol{\mathcal{L}}, \gamma \models \mathrm{NB}[\mathsf{S}, \mathsf{P}_1, \mathsf{P}_2]$, because by Proposition 4.14, $\boldsymbol{\mathcal{L}}, \alpha \models \mathrm{NB}[\mathsf{S}, \bigsqcap_i^{n-1} \mathsf{P}_i, \mathsf{P}_n]$ follows from the fact that $\boldsymbol{\mathcal{L}}, \alpha \models \mathrm{NB}[\mathsf{S}, \mathsf{P}_i, \mathsf{P}_n]$ for all $i \in \{1, \ldots, n-1\}$. By Theorem 4.19, the rule AG_2 is sound, so we infer $\boldsymbol{\mathcal{L}}, \gamma \models \mathsf{S} \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2$, and therefore $\boldsymbol{\mathcal{L}}, \alpha \models \mathsf{S} \sqsubseteq \bigsqcap_i^{n-1} \mathsf{P}_i \sqcap \mathsf{P}_n$. □

The completeness proof for all AG_n rules is a straightforward generalization of the respective proof for AG_2.

Theorem 4.26. *AG_n is both backward and forward complete for all $n \in \mathbb{N}$.*

Proof. Let $n \in \mathbb{N}$. To show backward completeness of AG_n, choose a valuation α with $dom(\alpha) = \{\mathsf{S}, \mathsf{P}_1, \ldots, \mathsf{P}_n\}$ such that $\boldsymbol{\mathcal{L}}, \alpha \models \mathsf{S} \sqsubseteq \bigsqcap_i^n \mathsf{P}_i$. We have to show $\boldsymbol{\mathcal{L}}, \alpha \models \bigsqcap_{j \neq i}^n \mathsf{P}_j \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_i$, for all $i \in \{1, \ldots, n\}$ and $\boldsymbol{\mathcal{L}}, \alpha \models \mathrm{NB}[\mathsf{S}, \mathsf{P}_i, \mathsf{P}_j]$ for all $i, j \in \{1, \ldots, n\}$ with $i < j$.

The proof proceeds in a similar way than the proof of Theorem 4.20. Let $i, j \in \{1, \ldots, n\}$ with $i < j$. Since $\boldsymbol{\mathcal{L}}, \alpha \models \mathsf{S} \sqsubseteq \bigsqcap_i^n \mathsf{P}_i$, we have $\alpha(\mathsf{S}) \leq \alpha(\mathsf{P}_i)$, so for every $a \in \mathcal{J}(\boldsymbol{\mathcal{L}})$, $a \leq \alpha(\mathsf{S})$ implies $a \leq \alpha(\mathsf{P}_i)$. By Proposition 4.8, we get $\langle \alpha(\mathsf{S}), \alpha(\mathsf{P}_i), \alpha(\mathsf{P}_j) \rangle \in \mathrm{NB}$, i. e., $\boldsymbol{\mathcal{L}}, \alpha \models \mathrm{NB}[\mathsf{S}, \mathsf{P}_i, \mathsf{P}_j]$. The rest follows because all premises of AG_n are entailed by the conclusion $\mathsf{S} \sqsubseteq \bigsqcap_i^n \mathsf{P}_i$.

Forward completeness trivially follows by Proposition 3.52. □

The AG_n rules are not just arbitrary (sound and complete) inferences rules — they turn out to be the strongest fully sound assume-guarantee rules. This claim is made more precise by Theorem 4.30 below, whose proof rests on the lemmas 4.27 and 4.29.

Lemma 4.27. *Let $R : \Phi / t \sqsubseteq t'$ if $\Gamma[t_1, \ldots, t_n]$ be an A-G rule without auxiliary variables, i. e., $var(R) = var(t \sqsubseteq t')$. If R is sound then R is weaker than AG_m, where $m = size(t')$.*

Proof. Assume that R is sound and let σ be a substitution with $dom(\sigma) = \{\mathsf{S}, \mathsf{P}_1, \ldots, \mathsf{P}_m\}$ such that $\sigma(\mathsf{S}) = t$ and $\sigma(\mathsf{P}_i) \in var(t')$ for all $i \in \{1, \ldots, m\}$. Note that such a substitution exists because $t \sqsubseteq t'$ is in normal form, $t \neq \top$ and t' consists of m distinct variables. We prove the claim that R is weaker than AG_m by Corollary 3.36, so we check its conditions:

1. Obvious.

2. As $\sigma(\mathsf{S} \sqsubseteq \bigsqcap_i^m \mathsf{P}_i)$ is $t \sqsubseteq t'$, the conclusion of R is trivially entailed by the conclusion of AG_m under σ. We still have to show for all $i \in \{1, \ldots, m\}$ that $\Phi \models \sigma(\bigsqcap_{j \neq i}^m \mathsf{P}_j \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_i)$, so let $i \in \{1, \ldots, m\}$, let $x = \sigma(\mathsf{P}_i)$ and $t'_x = \sigma(\bigsqcap_{j \neq i}^m \mathsf{P}_j)$. As $x \in var(t')$, x is a global guarantee. Therefore, x cannot be a global assumption, so by definition of assume-guarantee rules, it must be a local guarantee. Thus, there exists a premise $t_x \sqsubseteq x \in \Phi$, and hence $\Phi \models t_x \sqsubseteq x$. As $t_x \sqsubseteq x$ is in normal form, we have $var(t_x) \subseteq var(R) \setminus \{x\} = var(t \sqsubseteq t') \setminus \{x\} = var(t'_x \sqcap t)$. Therefore, $\Phi \models t_x \sqsubseteq x$ implies $\Phi \models t'_x \sqcap t \sqsubseteq x$, i. e., $\Phi \models \sigma(\bigsqcap_{j \neq i}^m \mathsf{P}_j \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_i)$.

3. For brevity, we denote the side condition of AG_m by $\mathrm{NB}[\mathsf{S}, \mathsf{P}_1, \ldots, \mathsf{P}_m]$. We have to show $\Phi, \Gamma[t_1, \ldots, t_n] \models^{\boldsymbol{\mathcal{L}}} \sigma(\mathrm{NB}[\mathsf{S}, \mathsf{P}_1, \ldots, \mathsf{P}_m])$, so let α be a valuation with $dom(\alpha) = var(\Phi, \Gamma[t_1, \ldots, t_n]) \cup var(\sigma(\mathrm{NB}[\mathsf{S}, \mathsf{P}_1, \ldots, \mathsf{P}_m]))$ such that $\boldsymbol{\mathcal{L}}, \alpha \models \Phi, \Gamma[t_1, \ldots, t_n]$. As R is sound, we infer $\boldsymbol{\mathcal{L}}, \alpha \models t \sqsubseteq t'$ using Proposition 3.40, i. e., we have $\boldsymbol{\mathcal{L}}, \alpha \models \sigma(\mathsf{S} \sqsubseteq \bigsqcap_i^m \mathsf{P}_i)$. By the Substitution Lemma, this implies $\boldsymbol{\mathcal{L}}, \alpha \circ \sigma \models \mathsf{S} \sqsubseteq \bigsqcap_i^m \mathsf{P}_i$. Note that actually $dom(\alpha) = var(rng(\sigma))$ since $var(\Phi, \Gamma[t_1, \ldots, t_n]) \subseteq var(R) = var(t \sqsubseteq t') =$

$var(rng(\sigma))$, so $dom(\alpha \circ \sigma) = dom(\sigma) = var(\mathsf{S} \sqsubseteq \bigsqcap_i^m \mathsf{P}_i)$. Therefore, we get $\boldsymbol{\mathcal{L}}, \alpha \circ \sigma \models \mathrm{NB}[\mathsf{S}, \mathsf{P}_1, \ldots, \mathsf{P}_m]$ by backward completeness of AG_m, see Theorem 4.26. Finally, $\boldsymbol{\mathcal{L}}, \alpha \models \sigma(\mathrm{NB}[\mathsf{S}, \mathsf{P}_1, \ldots, \mathsf{P}_m])$ follows by the Substitution Lemma; note that this last step required that $dom(\alpha) = var(rng(\sigma))$. □

Lemma 4.28. AG_n *is non-empty for all* $n \in \mathbb{N}$.

Proof. Let $n \in \mathbb{N}$ and denote the premises resp. conclusion of AG_n by Φ_n resp. ψ_n. We define the valuation α with $dom(\alpha) = var(\mathrm{AG}_n)$, where $\alpha(\mathsf{S}) = \alpha(\mathsf{P}_1) = \cdots = \alpha(\mathsf{P}_n) = 1$. Obviously, $\boldsymbol{\mathcal{L}}, \alpha \models \bigsqcap_{j \neq i}^n \mathsf{P}_j \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_i$ for all $i \in \{1, \ldots, n\}$, so we have $\boldsymbol{\mathcal{L}}, \alpha \models \Phi_n$, hence $\langle \Phi_n, \psi_n, \alpha \rangle$ is an inference. As $\boldsymbol{\mathcal{L}}, \alpha \models \mathrm{NB}[\mathsf{S}, \mathsf{P}_i, \mathsf{P}_j]$ for all $i, j \in \{1, \ldots, n\}$ with $i < j$ by Proposition 4.8, $\langle \Phi_n, \psi_n, \alpha \rangle$ is an inference of AG_n. □

Lemma 4.29. *For all* $m, n \in \mathbb{N}$, *if* AG_m *is weaker than* AG_n *then* $m = n$.

Proof. Let $m, n \in \mathbb{N}$ and assume that AG_m is weaker than AG_n. For brevity, we denote the premises and conclusion of AG_k by Φ_k and ψ_k, respectively, where $k \in \{m, n\}$. By Lemma 4.28, AG_m is non-empty, so by Corollary 3.37, there exists a substitution σ with $dom(\sigma) = var(\mathrm{AG}_n)$, $var(rng(\sigma)) \subseteq var(\mathrm{AG}_m)$, $\Phi_m \models \sigma(\Phi_n)$ and $\sigma(\psi_n) \models \psi_m$.

We claim that $\sigma(\mathsf{P}_i) \in \mathcal{V}$ for all $i \in \{1, \ldots, n\}$. To prove this claim by contradiction, we assume $\sigma(\mathsf{P}_i) \notin \mathcal{V}$ for some $i \in \{1, \ldots, n\}$. Since $\Phi_m \models \sigma(\Phi_n)$, we know that $\Phi_m \models \sigma(\bigsqcap_{j \neq i}^n \mathsf{P}_j \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_i)$. We define the valuation α with $dom(\alpha) = var(\mathrm{AG}_m)$ such that for all $x \in var(\mathrm{AG}_m)$, $\alpha(x) = 0$ if $x \in var(\sigma(\mathsf{P}_i))$, otherwise $\alpha(x) = 1$. Then $\alpha(\sigma(\bigsqcap_{j \neq i}^n \mathsf{P}_j \sqcap \mathsf{S})) = 1$ and $\alpha(\sigma(\mathsf{P}_i)) = 0$, so we have $\boldsymbol{\mathcal{L}}, \alpha \not\models \sigma(\bigsqcap_{j \neq i}^n \mathsf{P}_j \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_i)$. On the other hand, fix an arbitrary $k \in \{1, \ldots, m\}$. As $var(\bigsqcap_{j \neq k}^m \mathsf{P}_j \sqcap \mathsf{S}) = var(\mathrm{AG}_m) \setminus \{\mathsf{P}_k\}$ and $\big(var(\mathrm{AG}_m) \setminus \{\mathsf{P}_k\}\big) \cap var(\sigma(\mathsf{P}_i)) \neq \emptyset$, we have $\alpha(\bigsqcap_{j \neq k}^m \mathsf{P}_j \sqcap \mathsf{S}) = 0$, which implies $\boldsymbol{\mathcal{L}}, \alpha \models \bigsqcap_{j \neq k}^m \mathsf{P}_j \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_k$. Hence, α witnesses that $\Phi_m \not\models \sigma(\bigsqcap_{j \neq i}^n \mathsf{P}_j \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_i)$, which contradicts $\Phi_m \models \sigma(\Phi_n)$.

From $\sigma(\psi_n) \models \psi_m$, i. e., $\sigma(\mathsf{S}) \sqsubseteq \sigma(\bigsqcap_i^n \mathsf{P}_i) \models \mathsf{S} \sqsubseteq \bigsqcap_i^m \mathsf{P}_i$, it follows that $\bigsqcap_i^m \mathsf{P}_i$ is a subterm of $\sigma(\bigsqcap_i^n \mathsf{P}_i)$. And as $\sigma(\mathsf{P}_i) \in \mathcal{V}$ for all $i \in \{1, \ldots, n\}$, this implies that $m = size(\bigsqcap_i^m \mathsf{P}_i) \leq size(\sigma(\bigsqcap_i^n \mathsf{P}_i)) = n$. On the other hand, $n \leq m$ follows from $dom(\sigma) = var(\mathrm{AG}_n)$ and $var(rng(\sigma)) \subseteq var(\mathrm{AG}_m)$. Hence $m = n$. □

Theorem 4.30. *If* $\boldsymbol{\mathcal{L}}$ *is well-approximable then in* $\boldsymbol{R(\mathcal{L})}$, *the lattice of inference rules ordered by the weaker-than relation, the family of assume-guarantee rules* $\{\mathrm{AG}_n \mid n \in \mathbb{N}\}$ *forms a* maximals' cover *of the set of fully sound assume-guarantee rules — that means*

1. *every fully sound A-G rule is weaker than* AG_n *for some* $n \in \mathbb{N}$, *and*

2. *for every* $n \in \mathbb{N}$, AG_n *is maximal in the set of fully sound A-G rules.*

Proof. To prove 1, let $R : \Phi/t \sqsubseteq t'$ if $\Gamma[t_1, \ldots, t_m]$ be a fully sound A-G rule and let t'' be the term with $var(t'') = var(R) \setminus var(t \sqsubseteq t')$, i. e., $var(t'')$ is the set of auxiliary variables of R. Then $R'' : \Phi/t \sqsubseteq t' \sqcap t''$ if $\Gamma[t_1, \ldots, t_m]$ is an A-G rule without auxiliary variables. Furthermore, using Proposition 3.45 an induction on $size(t'')$ yields that R is weaker than R'' and R'' is fully sound, hence sound. By Lemma 4.27, R'' is weaker than AG_n where $n = size(t' \sqcap t'')$. Thus, R is weaker than AG_n.

To prove 2, assume that $\boldsymbol{\mathcal{L}}$ is well-approximable and let $n \in \mathbb{N}$. By Theorem 4.25, the A-G rule AG_n is sound. Actually, AG_n is fully sound by Proposition 3.43 because AG_n has no auxiliary variables. To show maximality, let R be a fully sound A-G rule which is stronger than AG_n. By 1, R is weaker than AG_m for some $m \in \mathbb{N}$. Thus, AG_n is weaker than AG_m, which implies $n = m$ by Lemma 4.29, which implies $R = \mathrm{AG}_n$ in turn. □

Note that an inspection of the above proof lets us state condition 1 of the theorem more precisely: Every fully sound A-G rule R is weaker than AG_n, where n is the number of local guarantees in R.

Corollary 4.31. AG_2 *is the strongest sound circular A-G rule with* 3 *variables.*

Proof. Let $R : \Phi/\psi$ if $\Gamma[t_1, \ldots, t_n]$ be a sound circular A-G rule with $var(R) = \{x, y, z\}$, where x, y and z are three distinct variables. Because of circularity, R must have one global assumption and two local guarantees; w. l. o. g. we assume that x is the global assumption and y and z are local guarantees. As the dependency relation must be cyclic, we know that $y \rightarrow_\Phi z$ and $z \rightarrow_\Phi y$. Furthermore, circularity implies that the right-hand side of ψ must be different from $\top$, so we know that $\psi \models x \sqsubseteq y$ or $\psi \models x \sqsubseteq z$; w. l. o. g. we assume that $\psi \models x \sqsubseteq y$. Then soundness of R yields $\Phi, \Gamma[t_1, \ldots, t_n] \models^{\boldsymbol{\mathcal{L}}} x \sqsubseteq y$. And as $z \rightarrow_\Phi y$ implies $\Phi \models y \sqcap x \sqsubseteq z$, we also get $\Phi, \Gamma[t_1, \ldots, t_n] \models^{\boldsymbol{\mathcal{L}}} x \sqsubseteq z$. Hence R is fully sound, so by Theorem 4.30 it is weaker than AG_n, where n is the number of local guarantees in R, i. e., R is weaker than AG_2. □

To end this section, we will justify the claim of Section 4.3 that AG_2 is *the* basic circular assume-guarantee rule. The most important fact supporting this claim is that the proof of soundness of the generalization AG_n uses Theorem 4.19, i. e., soundness of AG_2. The same applies to the proofs of soundness for all other circular A-G rules that are presented further below in this chapter. Another fact is that AG_2 is among the simplest circular A-G rules, where we measure simplicity by the number of variables occuring in a rule. This follows from Corollary 3.61, which states that there cannot exist any circular A-G rule with less than three variables. Moreover by Corollary 4.31, AG_2 is the strongest sound rule among these circular A-G rules with three variables.

4.5 Another Extension

In this section, we present an extension of the circular A-G rule AG_2 to a another family of A-G rules $\{\mathrm{AG}'_n \mid n \in \mathbb{N}\}$. Again, we prove soundness (provided that $\boldsymbol{\mathcal{L}}$ is well-approximable) and completeness.

Definition 4.32 (Assume-Guarantee Rule AG'_n).
Given $n \in \mathbb{N}$, we define the inference rule

$$\mathrm{AG}'_n : \frac{\mathsf{A}_1 \sqcap \mathsf{S} \sqsubseteq \mathsf{G}_1 \quad \ldots \quad \mathsf{A}_n \sqcap \mathsf{S} \sqsubseteq \mathsf{G}_n \quad \prod_i^n \mathsf{G}_i \sqcap \mathsf{S} \sqsubseteq \prod_i^n \mathsf{A}_i}{\mathsf{S} \sqsubseteq \prod_i^n \mathsf{G}_i} \text{ if } \bigwedge_{1 \leq i \leq n} \mathrm{NB}[\mathsf{S}, \mathsf{A}_i, \mathsf{G}_i].$$

An inference rule AG'_n is best understood when we think of the variable S as representing a system whereas the variables A_i and G_j represent properties that function as assumptions and guarantees, respectively. Assumptions and guarantees are paired to form n assume-guarantee specifications (A-G specs, for short), and the i-th premise $\mathsf{A}_i \sqcap \mathsf{S} \sqsubseteq \mathsf{G}_i$ expresses that the system satisfies the i-th A-G spec, i. e., assuming the i-th assumption, the system guarantees the i-th guarantee. As the conclusion expresses that the system guarantees the conjunction of all n guarantees, the purpose of rule AG'_n is to combine the n A-G specs and discharge their assumptions. According to the rule, this is possible if there is a 'feedback' from the guarantees to the assumptions (cf. premise $\prod_i^n \mathsf{G}_i \sqcap \mathsf{S} \sqsubseteq \prod_i^n \mathsf{A}_i$) and assumptions and guarantees do not block each other relative to the system (cf. side condition).

Note that strictly speaking, AG'_n is not an A-G rule because its last premise is not in strong normal form. However, it can be transformed into an A-G rule by splitting $\prod_i^n \mathsf{G}_i \sqcap \mathsf{S} \sqsubseteq \prod_i^n \mathsf{A}_i$ into the n premises $\prod_i^n \mathsf{G}_i \sqcap \mathsf{S} \sqsubseteq \mathsf{A}_1$, $\ldots$, $\prod_i^n \mathsf{G}_i \sqcap \mathsf{S} \sqsubseteq \mathsf{A}_n$. For the sake of readability we have decided not to do so.

Proposition 4.33. *The A-G rule AG'_n is circular if and only if $n \geq 1$.*

Proof. The syntactic rule $\mathrm{AG}'_0 : \{\mathsf{S} \sqsubseteq \top\}/\mathsf{S} \sqsubseteq \top$ is clearly non-circular, so assume $n \geq 1$. We define the valuation α with $dom(\alpha) = \{\mathsf{S}, \mathsf{A}_1, \ldots, \mathsf{A}_n, \mathsf{G}_1, \ldots, \mathsf{G}_n\}$, where $\alpha(\mathsf{S}) = 1$ and $\alpha(\mathsf{A}_i) = \alpha(\mathsf{G}_i) = 0$ for all $i \in \{1, \ldots, n\}$. Then all premises of AG'_n are true under α but the conclusion is not, i. e., AG'_n is circular. □

We prove AG'_n sound by the strength of AG_2, i. e., we show that AG_2 is stronger than AG'_n, so soundness of the latter follows from soundness of the former.

Theorem 4.34. *If $\boldsymbol{\mathcal{L}}$ is well-approximable then AG'_n is fully sound for all $n \in \mathbb{N}$.*

Proof. Let $n \in \mathbb{N}$. We are going to show that AG'_n is weaker than AG_2. Define the substitution σ with $dom(\sigma) = \{\mathsf{S}, \mathsf{P}_1, \mathsf{P}_2\}$, where $\sigma(\mathsf{S}) = \mathsf{S}$ and $\sigma(\mathsf{P}_1) = \prod_i^n \mathsf{A}_i$ and $\sigma(\mathsf{P}_2) = \prod_i^n \mathsf{G}_i$. We have to check the conditions of Corollary 3.36:

1. Obvious.

2. As $\bigsqcap_i^n \mathsf{G}_i \sqcap \mathsf{S} \sqsubseteq \bigsqcap_i^n \mathsf{A}_i$ is $\sigma(\mathsf{P}_2 \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_1)$ and $\mathsf{A}_1 \sqcap \mathsf{S} \sqsubseteq \mathsf{G}_1, \ldots, \mathsf{A}_n \sqcap \mathsf{S} \sqsubseteq \mathsf{G}_n \models \sigma(\mathsf{P}_1 \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_2)$, the premises of AG'_n entail the premises of AG_2 under σ. Furthermore, $\sigma(\mathsf{S} \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2)$ obviously entails $\mathsf{S} \sqsubseteq \bigsqcap_i^n \mathsf{G}_i$.

3. The fact $\mathsf{A}_1 \sqcap \mathsf{S} \sqsubseteq \mathsf{G}_1, \ldots, \mathsf{A}_n \sqcap \mathsf{S} \sqsubseteq \mathsf{G}_n$, $\mathrm{NB}[\mathsf{S}, \mathsf{A}_1, \mathsf{G}_1], \ldots, \mathrm{NB}[\mathsf{S}, \mathsf{A}_n, \mathsf{G}_n] \models^{\boldsymbol{\mathcal{L}}} \mathrm{NB}[\mathsf{S}, \bigsqcap_i^n \mathsf{A}_i, \bigsqcap_i^n \mathsf{G}_i]$ is implied by Proposition 4.15. Therefore, the premises and the side condition of AG'_n together entail $\sigma(\mathrm{NB}[\mathsf{S}, \mathsf{P}_1, \mathsf{P}_2])$.

By Proposition 3.39, soundness of AG'_n follows from Theorem 4.19, so we know that the variables $\mathsf{S}, \mathsf{G}_1, \ldots, \mathsf{G}_n$ all are sound in AG'_n by Proposition 3.43. Thanks to the premise $\bigsqcap_i^n \mathsf{G}_i \sqcap \mathsf{S} \sqsubseteq \bigsqcap_i^n \mathsf{A}_i$, the auxiliary variables $\mathsf{A}_1, \ldots, \mathsf{A}_n$ are also sound in AG'_n, hence AG'_n is fully sound. □

In principle, the proof of completeness is similar to the proof of Theorem 4.20, completeness of AG_2. However in AG'_n, the variables $\mathsf{A}_1, \ldots, \mathsf{A}_n$ are auxiliary variables, so forward completeness no longer follows from backward completeness.

Theorem 4.35. AG'_n *is both backward and forward complete for all* $n \in \mathbb{N}$.

Proof. Let $n \in \mathbb{N}$. For brevity, we denote the premises resp. conclusion of AG'_n by Φ_n resp. ψ_n. To show forward completeness of AG'_n, choose a valuation α with $dom(\alpha) = var(\mathrm{AG}'_n)$ such that $\boldsymbol{\mathcal{L}}, \alpha \models \Phi_n, \psi_n$. Fix an arbitrary $i \in \{1, \ldots, n\}$. Since $\boldsymbol{\mathcal{L}}, \alpha \models \psi_n$, we have $\alpha(\mathsf{S}) \leq \alpha(\mathsf{G}_i)$, so for every $a \in \mathcal{J}(\boldsymbol{\mathcal{L}})$, $a \leq \alpha(\mathsf{S})$ implies $a \leq \alpha(\mathsf{G}_i)$. By Proposition 4.8, we conclude that $\langle \alpha(\mathsf{S}), \alpha(\mathsf{A}_i), \alpha(\mathsf{G}_i) \rangle \in \mathrm{NB}$, i. e., $\boldsymbol{\mathcal{L}}, \alpha \models \mathrm{NB}[\mathsf{S}, \mathsf{A}_i, \mathsf{G}_i]$. As i is arbitrary, the side condition is true under α, hence AG'_n is forward complete.

To show backward completeness of AG'_n, choose a valuation α with $dom(\alpha) = \{\mathsf{S}, \mathsf{G}_1, \ldots, \mathsf{G}_n\}$ such that $\boldsymbol{\mathcal{L}}, \alpha \models \psi_n$. Extend the valuation α to β such that $\alpha \trianglelefteq \beta$, $dom(\beta) = var(\mathrm{AG}'_n)$ and $\beta(\mathsf{A}_i) = 1$ for all $i \in \{1, \ldots, n\}$. Obviously, we have $\boldsymbol{\mathcal{L}}, \beta \models \psi_n$. Moreover, we get $\boldsymbol{\mathcal{L}}, \beta \models \bigsqcap_i^n \mathsf{G}_i \sqcap \mathsf{S} \sqsubseteq \bigsqcap_i^n \mathsf{A}_i$, and for all $i \in \{1, \ldots, n\}$, $\boldsymbol{\mathcal{L}}, \beta \models \psi_n$ implies $\boldsymbol{\mathcal{L}}, \beta \models \mathsf{S} \sqsubseteq \mathsf{G}_i$, hence $\boldsymbol{\mathcal{L}}, \beta \models \mathsf{A}_i \sqcap \mathsf{S} \sqsubseteq \mathsf{G}_i$. Thus, we have $\boldsymbol{\mathcal{L}}, \beta \models \Phi_n, \psi_n$, so the side condition of AG'_n is true under β by forward completeness. Therefore, premises and side condition are satisfiable under α, hence AG'_n is backward complete. □

4.6 Compositional Assume-Guarantee Rules

In the introduction to Section 4.4 we claimed that assume-guarantee reasoning is useful in verification because it reduces a goal to a number of subgoals. These subgoals are of the form *assuming some properties, the system guarantees some other property.* If the system is not monolithic but compound of subsystems then there is a chance that in each subgoal, assuming the to-be-assumed properties, the to-be-guaranteed property is guaranteed by some subsystem already. This potential decomposition of the system in the subgoals is the reason why there is

hope that the subgoals are easier to verify than the original goal. However, the A-G rules presented so far do not explicitly force system decomposition, which is why we term them non-compositional.

To make system decomposition in an A-G rule $R : \Phi/\psi$ if $\Gamma[t_1, \ldots, t_n]$ explicit, we must explicitly represent the system to be verified as a composition of subsystems, i. e., the left-hand side of the conclusion ψ must be a term of the form $\bigsqcap_i^m \mathsf{S}_i$, where the variables S_1 to S_m represent the subsystems. Certainly, R cannot be termed compositional if $\bigsqcap_i^m \mathsf{S}_i$ occurs as a subterm in some premise. We call R *compositional in the premises* iff only proper subterms of $\bigsqcap_i^m \mathsf{S}_i$ occur in the premises, i. e., iff $var(\bigsqcap_i^m \mathsf{S}_i) \not\subseteq var(\varphi)$ for all $\varphi \in \Phi$. In this case, the system is decomposed in all subgoals arising from premises.

However, not only the premises generate subgoals; establishing truth of the side condition must also be seen as a subgoal. Therefore, R cannot be termed compositional if the side condition $\Gamma[t_1, \ldots, t_n]$ involves the composition of all subsystems, which would be the case, for instance, if $\bigsqcap_i^m \mathsf{S}_i$ were a subterm of some term t_i. We call R *compositional in the side condition* iff the side condition does not involve all subsystems or it is a boolean combination of conditions none of which involves all subsystems. Formally, R is compositional in the side condition iff $var(\bigsqcap_i^m \mathsf{S}_i) \not\subseteq var(\Gamma[t_1, \ldots, t_n])$ or there is a finite set of named relations C such that $\Gamma[t_1, \ldots, t_n]$ is expressible as a boolean combination of relations in C and $var(\bigsqcap_i^m \mathsf{S}_i) \not\subseteq var(\Gamma'[t'_1, \ldots, t'_{n'}])$ for all $\Gamma'[t'_1, \ldots, t'_{n'}] \in C$. In these cases, the system is decomposed in all subgoals arising from the side condition. Finally, we call R *compositional* iff it is compositional in the premises and in the side condition.

For simplicity, we restrict the number of subsystems to two in this section. Following, we present a compositional circular A-G rule for two properties, i. e., a variant of AG_2 which is compositional in the premises and in the side condition. Its soundness follows directly from soundness of AG_2.

Definition 4.36 (Compositional Circular A-G Rule $\mathrm{AG}_2^{\mathrm{c}}$).

$$\mathrm{AG}_2^{\mathrm{c}} : \frac{\mathsf{P}_2 \sqcap \mathsf{S}_1 \sqsubseteq \mathsf{P}_1 \qquad \mathsf{P}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_2}{\mathsf{S}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2} \text{ if } \mathrm{NB}[\top, \mathsf{P}_1, \mathsf{P}_2]$$

Theorem 4.37. *If $\boldsymbol{\mathcal{L}}$ is well-approximable then $\mathrm{AG}_2^{\mathrm{c}}$ is sound.*

Proof. By Proposition 3.39, it suffices to show that $\mathrm{AG}_2^{\mathrm{c}}$ is weaker than AG_2, then soundness follows from Theorem 4.19. We define the substitution σ with $dom(\sigma) = \{\mathsf{S}, \mathsf{P}_1, \mathsf{P}_2\}$ by $\sigma(\mathsf{S}) = \mathsf{S}_1 \sqcap \mathsf{S}_2$, $\sigma(\mathsf{P}_1) = \mathsf{P}_1$ and $\sigma(\mathsf{P}_2) = \mathsf{P}_2$. Obviously, condition 1 of Corollary 3.36 holds. Condition 2 holds because $\mathsf{P}_2 \sqcap \mathsf{S}_1 \sqsubseteq \mathsf{P}_1 \models \sigma(\mathsf{P}_2 \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_1)$ and $\mathsf{P}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_2 \models \sigma(\mathsf{P}_1 \sqcap \mathsf{S} \sqsubseteq \mathsf{P}_2)$. And condition 3 is satisfied since by Proposition 4.13, $\mathrm{NB}[\top, \mathsf{P}_1, \mathsf{P}_2] \models^{\boldsymbol{\mathcal{L}}} \sigma(\mathrm{NB}[\mathsf{S}, \mathsf{P}_1, \mathsf{P}_2])$. □

Note that $\mathrm{AG}_2^{\mathrm{c}}$ is strictly weaker than AG_2 because of condition 1 of Corollary 3.37. Moreover, the side condition of $\mathrm{AG}_2^{\mathrm{c}}$ is more restrictive, which is why

$\mathrm{AG}_2^{\mathrm{c}}$ is neither forward nor backward complete. However, $\mathrm{AG}_2^{\mathrm{c}}$ is complete in another sense, namely its side condition characterizes exactly when the premises entail the conclusion.

Theorem 4.38. *If $\mathcal{J}(\boldsymbol{\mathcal{L}})$ is join-dense in $\boldsymbol{\mathcal{L}}$ then the side condition $\mathrm{NB}[\top, \mathsf{P}_1, \mathsf{P}_2]$ is complete in $\mathrm{AG}_2^{\mathrm{c}}$.*

Proof. Let $\boldsymbol{\mathcal{J}} = \langle \mathcal{J}, \preceq \rangle$ be the suborder of $\boldsymbol{\mathcal{L}}$ which is generated by $\mathcal{J}(\boldsymbol{\mathcal{L}})$, i. e., $\mathcal{J} = \mathcal{J}(\boldsymbol{\mathcal{L}})$. Assume that $\mathcal{J}$ is join-dense in $\boldsymbol{\mathcal{L}}$ and choose a valuation α with $dom(\alpha) = \{\mathsf{P}_1, \mathsf{P}_2\}$ such that $\mathsf{P}_2 \sqcap \mathsf{S}_1 \sqsubseteq \mathsf{P}_1$, $\mathsf{P}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_2 \models_\alpha^{\boldsymbol{\mathcal{L}}} \mathsf{S}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2$. We have to show that $\boldsymbol{\mathcal{L}}, \alpha \models \mathrm{NB}[\top, \mathsf{P}_1, \mathsf{P}_2]$. Towards a contradiction, we assume that $\alpha(\mathsf{P}_1)$ and $\alpha(\mathsf{P}_2)$ block each other, i. e., there exists $a \in \mathcal{J}$ such that $\alpha(\mathsf{P}_1) \wedge \alpha(\mathsf{P}_2)$ is an upper bound of $\prec(a)$ but $a \not\leq \alpha(\mathsf{P}_1)$ and $a \not\leq \alpha(\mathsf{P}_2)$.

As $\mathcal{J}$ is join-dense, there is $J_1 \subseteq \mathcal{J}$ such that $\alpha(\mathsf{P}_1) \wedge a = \bigvee J_1$. From $a \not\leq \alpha(\mathsf{P}_1)$ follows $\alpha(\mathsf{P}_1) \wedge a < a$, so $\bigvee J_1 < a$, i. e., $J_1 \subseteq {<}(a)$. Actually, $J_1 \subseteq \prec(a)$ because $a \in \mathcal{J}$ and $J_1 \subseteq \mathcal{J}$ and the order $\preceq$ on $\mathcal{J}$ is induced by $\leq$. So $\alpha(\mathsf{P}_1) \wedge \alpha(\mathsf{P}_2)$, which is an upper bound of $\prec(a)$, must also be an upper bound of J_1, which in turn implies that $\alpha(\mathsf{P}_1) \wedge a = \bigvee J_1 \leq \alpha(\mathsf{P}_1) \wedge \alpha(\mathsf{P}_2) \leq \alpha(\mathsf{P}_2)$.

Likewise, by join-density of $\mathcal{J}$ there is $J_2 \subseteq \mathcal{J}$ such that $\alpha(\mathsf{P}_2) \wedge a = \bigvee J_2$. By similar arguments than above, $a \not\leq \alpha(\mathsf{P}_2)$ implies $J_2 \subseteq \prec(a)$, hence we obtain that $\alpha(\mathsf{P}_2) \wedge a = \bigvee J_2 \leq \alpha(\mathsf{P}_1) \wedge \alpha(\mathsf{P}_2) \leq \alpha(\mathsf{P}_1)$.

Define β to be the extension of α which maps S_1 and S_2 to a, i. e., $dom(\beta) = \{\mathsf{S}_1, \mathsf{S}_2, \mathsf{P}_1, \mathsf{P}_2\}$ and $\alpha \trianglelefteq \beta$ and $\beta(\mathsf{S}_1) = \beta(\mathsf{S}_2) = a$. The above arguments yield $\boldsymbol{\mathcal{L}}, \beta \models \mathsf{P}_2 \sqcap \mathsf{S}_1 \sqsubseteq \mathsf{P}_1$, $\mathsf{P}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_2$, which implies $\boldsymbol{\mathcal{L}}, \beta \models \mathsf{S}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2$ by choice of α. Hence we have $a \wedge a \leq \alpha(\mathsf{P}_1) \wedge \alpha(\mathsf{P}_2)$, which contradicts $a \not\leq \alpha(\mathsf{P}_1)$ and $a \not\leq \alpha(\mathsf{P}_2)$. □

Corollary 4.39. *Let α be a valuation with $dom(\alpha) = \{\mathsf{P}_1, \mathsf{P}_2\}$. If $\boldsymbol{\mathcal{L}}$ is well-approximable then the following statements are equivalent:*

1. $\boldsymbol{\mathcal{L}}, \alpha \models \mathrm{NB}[\top, \mathsf{P}_1, \mathsf{P}_2]$.
2. $\mathsf{P}_2 \sqcap \mathsf{S}_1 \sqsubseteq \mathsf{P}_1$, $\mathsf{P}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_2 \models_\alpha^{\boldsymbol{\mathcal{L}}} \mathsf{S}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2$.

Proof. Follows from soundness and completeness of $\mathrm{AG}_2^{\mathrm{c}}$ — recall that by Proposition 4.5, $\mathcal{J}(\boldsymbol{\mathcal{L}})$ is join-dense $\boldsymbol{\mathcal{L}}$ whenever $\boldsymbol{\mathcal{L}}$ is well-approximable. □

Example 4.40. To show that $\mathrm{AG}_2^{\mathrm{c}}$ is not forward complete, consider any well-approximable lattice $\boldsymbol{\mathcal{L}}$. Define the valuation α with $dom(\alpha) = \{\mathsf{S}_1, \mathsf{S}_2, \mathsf{P}_1, \mathsf{P}_2\}$ such that $\alpha(\mathsf{S}_1) = \alpha(\mathsf{S}_2) = \alpha(\mathsf{P}_1) = \alpha(\mathsf{P}_2) = 0$. Then premises and conclusion of $\mathrm{AG}_2^{\mathrm{c}}$ are true under α, i. e., $\boldsymbol{\mathcal{L}}, \alpha \models \mathsf{P}_2 \sqcap \mathsf{S}_1 \sqsubseteq \mathsf{P}_1$, $\mathsf{P}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_2$, $\mathsf{S}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2$. Yet with $\beta = \alpha|_{\{\mathsf{P}_1, \mathsf{P}_2\}}$, we get $\mathsf{P}_2 \sqcap \mathsf{S}_1 \sqsubseteq \mathsf{P}_1$, $\mathsf{P}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_2 \not\models_\beta^{\boldsymbol{\mathcal{L}}} \mathsf{S}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2$, which implies $\boldsymbol{\mathcal{L}}, \beta \not\models \mathrm{NB}[\top, \mathsf{P}_1, \mathsf{P}_2]$ by Corollary 4.39. Therefore $\boldsymbol{\mathcal{L}}, \alpha \not\models \mathrm{NB}[\top, \mathsf{P}_1, \mathsf{P}_2]$, i. e., the side condition of $\mathrm{AG}_2^{\mathrm{c}}$ is not true under α.

By Proposition 3.52, $\mathrm{AG}_2^{\mathrm{c}}$ not being forward complete implies that $\mathrm{AG}_2^{\mathrm{c}}$ cannot be backward complete.

Together with Theorem 4.38, the above example confirms a statement from Section 3.5, namely that completeness in the side condition entails neither backward nor forward completeness.

Example 4.41. We present a variant of $\mathrm{AG}_2^{\mathrm{c}}$, a compositional circular A-G rule for two properties whose side condition is a boolean combination of two relations, each of which involves only one subsystem.

$$\mathrm{AG}_2^{\mathrm{c}\prime} : \frac{\mathsf{P}_2 \sqcap \mathsf{S}_1 \sqsubseteq \mathsf{P}_1 \qquad \mathsf{P}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_2}{\mathsf{S}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2} \text{ if } \mathrm{NB}[\mathsf{S}_1, \mathsf{P}_1, \mathsf{P}_2] \vee \mathrm{NB}[\mathsf{S}_2, \mathsf{P}_1, \mathsf{P}_2]$$

Soundness of $\mathrm{AG}_2^{\mathrm{c}\prime}$ (provided that $\boldsymbol{\mathcal{L}}$ is well-approximable) is proven in essentially the same way than soundness of $\mathrm{AG}_2^{\mathrm{c}}$. More precisely, the substitution σ from the proof of Theorem 4.37 witnesses that $\mathrm{AG}_2^{\mathrm{c}\prime}$ is weaker than AG_2; note that condition 3 of Corollary 3.36 is satisfied as for all $i \in \{1, 2\}$, $\mathrm{NB}[\mathsf{S}_i, \mathsf{P}_1, \mathsf{P}_2] \models^{\boldsymbol{\mathcal{L}}} \sigma(\mathrm{NB}[\mathsf{S}, \mathsf{P}_1, \mathsf{P}_2])$ by Proposition 4.13.

Obviously, the side condition of $\mathrm{AG}_2^{\mathrm{c}\prime}$ is less restrictive than the side condition of $\mathrm{AG}_2^{\mathrm{c}}$, so on the one hand, $\mathrm{AG}_2^{\mathrm{c}\prime}$ is strictly stronger than $\mathrm{AG}_2^{\mathrm{c}}$. On the other hand, $\mathrm{AG}_2^{\mathrm{c}\prime}$ is less complete, i. e., it is not all complete. Because there are no auxiliary variables and all variables occur in the side condition, it suffices to show that $\mathrm{AG}_2^{\mathrm{c}\prime}$ is not forward complete; backward incompleteness and incompleteness of the side condition then follow by Proposition 3.52.

To show that $\mathrm{AG}_2^{\mathrm{c}\prime}$ is not forward complete, consider the finite (hence well-approximable) lattice $\boldsymbol{\mathcal{L}} = \langle \mathcal{P}(\{1, 2, 3, 4\}), \subseteq \rangle$, see also Example 4.22. We define the valuation α with $dom(\alpha) = \{\mathsf{S}_1, \mathsf{S}_2, \mathsf{P}_1, \mathsf{P}_2\}$ by $\alpha(\mathsf{S}_1) = \{1\}$, $\alpha(\mathsf{S}_2) = \{2\}$, $\alpha(\mathsf{P}_1) = \{3\}$ and $\alpha(\mathsf{P}_2) = \{4\}$. Obviously, premises and conclusion of $\mathrm{AG}_2^{\mathrm{c}\prime}$ are true under α. However, $\{3\}$ and $\{4\}$ block each other relative to $\{1\}$ and relative to $\{2\}$, therefore we have $\boldsymbol{\mathcal{L}}, \alpha \not\models \mathrm{NB}[\mathsf{S}_i, \mathsf{P}_1, \mathsf{P}_2]$ for all $i \in \{1, 2\}$. Hence the side condition of $\mathrm{AG}_2^{\mathrm{c}\prime}$ is not true under α.

Note that in [Mai03], we prove that no sound compositional circular A-G rule can ever be forward complete. Sound and backward complete compositional circular A-G rules may exist, however, as a consequence of Proposition 3.52, they must necessarily employ auxiliary variables.

4.7 Beyond Well-founded Approximations

The proofs of soundness of all circular A-G rules that have been presented in this chapter required the lattice $\boldsymbol{\mathcal{L}}$ to be well-approximable. But is this requirement really necessary for soundness of circular assume-guarantee reasoning? This section can not provide a definite answer but we will argue that at least the A-G rules from this chapter tend to become unsound when $\boldsymbol{\mathcal{L}}$ is not well-approximable. Recall from Section 4.1 that $\boldsymbol{\mathcal{L}}$ can fail to be well-approximable for two reasons:

1. Because $\mathcal{J}(\boldsymbol{\mathcal{L}})$ is not join-dense in $\boldsymbol{\mathcal{L}}$, or
2. because the suborder of $\boldsymbol{\mathcal{L}}$ which is generated by $\mathcal{J}(\boldsymbol{\mathcal{L}})$ does not satisfy DCC.

We show that the first case inadvertently leads to unsoundness of the circular A-G rule $\mathrm{AG}_2^{\mathrm{c}\prime}$ from Example 4.41. Addressing the second case, we are able to show unsoundness only if $\boldsymbol{\mathcal{L}}$ satisfies additional requirements.

Proposition 4.42. *If $\mathcal{J}(\boldsymbol{\mathcal{L}})$ is not join-dense in $\boldsymbol{\mathcal{L}}$ then $\mathrm{AG}_2^{\mathrm{c}\prime}$ is unsound.*

Proof. Assume that $\mathcal{J}(\boldsymbol{\mathcal{L}})$ is not join-dense in $\boldsymbol{\mathcal{L}}$. Then there is $s \in \mathcal{L}$ such that $\bigvee X \neq s$ for all $X \subseteq \mathcal{J}(\boldsymbol{\mathcal{L}})$. Let $X = \{a \in \mathcal{J}(\boldsymbol{\mathcal{L}}) \mid a \leq s\}$ and $p = \bigvee X$ and note that $p < s$. Define the valuation α with $dom(\alpha) = \{\mathsf{S}_1, \mathsf{S}_2, \mathsf{P}_1, \mathsf{P}_2\}$ by $\alpha(\mathsf{S}_1) = \alpha(\mathsf{S}_2) = s$ and $\alpha(\mathsf{P}_1) = \alpha(\mathsf{P}_2) = p$. Then $\boldsymbol{\mathcal{L}}, \alpha \models \mathsf{P}_2 \sqcap \mathsf{S}_1 \sqsubseteq \mathsf{P}_1$, $\mathsf{P}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_2$ because $\alpha(\mathsf{P}_1) = \alpha(\mathsf{P}_2)$. But because $\alpha(\mathsf{P}_1 \sqcap \mathsf{P}_2) = p < s = \alpha(\mathsf{S}_1 \sqcap \mathsf{S}_2)$, we also have $\boldsymbol{\mathcal{L}}, \alpha \not\models \mathsf{S}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2$, so the premises of $\mathrm{AG}_2^{\mathrm{c}\prime}$ do not entail the conclusion under α. By Proposition 3.40, unsoundness follows provided that the side condition of $\mathrm{AG}_2^{\mathrm{c}\prime}$ is true under α.

We have to show $\boldsymbol{\mathcal{L}}, \alpha \models \mathrm{NB}[\mathsf{S}_1, \mathsf{P}_1, \mathsf{P}_2]$ or $\boldsymbol{\mathcal{L}}, \alpha \models \mathrm{NB}[\mathsf{S}_2, \mathsf{P}_1, \mathsf{P}_2]$, which amounts to showing that $\langle s, p, p\rangle \in \mathrm{NB}$. Let $a \in \mathcal{J}(\boldsymbol{\mathcal{L}}) \cap {\leq}(s)$. Then $a \in X$, so $a \leq \bigvee X = p$, hence $\langle s, p, p\rangle \in \mathrm{NB}$ by Proposition 4.8. □

If $\mathcal{J}(\boldsymbol{\mathcal{L}})$ is not join-dense in $\boldsymbol{\mathcal{L}}$ then by Proposition 3.39, all stronger rules than $\mathrm{AG}_2^{\mathrm{c}\prime}$, in particular AG_2, must be unsound, too. Note that the non-blockingness relation NB loses its restrictiveness if $\mathcal{J}(\boldsymbol{\mathcal{L}})$ is not join-dense in $\boldsymbol{\mathcal{L}}$. In the extreme case when $\mathcal{J}(\boldsymbol{\mathcal{L}}) = \emptyset$, e. g., when $\boldsymbol{\mathcal{L}} = \langle \mathbb{R}_\perp^\top, \leq\rangle$ (see Example 4.6), we have $\mathrm{NB} = \mathcal{L}^3$. In this case, even $\mathrm{AG}_2^{\mathrm{c}}$ is unsound although it is weaker than $\mathrm{AG}_2^{\mathrm{c}\prime}$.

Proposition 4.43. *Let $\boldsymbol{\mathcal{J}}$ be the suborder of $\boldsymbol{\mathcal{L}}$ which is generated by $\mathcal{J}(\boldsymbol{\mathcal{L}})$. If $\boldsymbol{\mathcal{J}}$ is forest-like and does not satisfy DCC then $\mathrm{AG}_2^{\mathrm{c}\prime}$ is unsound.*

Proof. Let $\boldsymbol{\mathcal{J}} = \langle \mathcal{J}, \preceq\rangle$, i. e., $\mathcal{J} = \mathcal{J}(\boldsymbol{\mathcal{L}})$, and assume that $\boldsymbol{\mathcal{J}}$ is forest-like and does not satisfy DCC. This implies that there is an infinite descending sequence $a_0 \succ a_1 \succ a_2 \succ \ldots$ of elements in $\mathcal{J}$. Let $s = a_0$ and $p = \bigwedge\{a_m \mid m \in \mathbb{N}\}$ and note that $p < s$. Define the valuation α with $dom(\alpha) = \{\mathsf{S}_1, \mathsf{S}_2, \mathsf{P}_2, \mathsf{P}_2\}$ by $\alpha(\mathsf{S}_1) = \alpha(\mathsf{S}_2) = s$ and $\alpha(\mathsf{P}_1) = \alpha(\mathsf{P}_2) = p$. Like in the proof of Proposition 4.42, the premises of $\mathrm{AG}_2^{\mathrm{c}\prime}$ do not entail the conclusion under α, so unsoundness of $\mathrm{AG}_2^{\mathrm{c}\prime}$ follows by Proposition 3.40 again, provided that the side condition of $\mathrm{AG}_2^{\mathrm{c}\prime}$ is true under α.

Again, we have to show that $\langle s, p, p\rangle \in \mathrm{NB}$, so let $a \in \mathcal{J} \cap {\leq}(s)$ and assume that $p \wedge p$ is an upper bound of ${\prec}(a)$. Choose an arbitrary $i \in \mathbb{N}$. As $s = a_0 \in \mathcal{J}$, we have $a_i \preceq s$ and $a \preceq s$. And as $\boldsymbol{\mathcal{J}}$ is forest-like, this implies $a \preceq a_i$ or $a_i \preceq a$. The latter would imply $a_{i+1} \in {\prec}(a)$, which would contradict p being an upper bound of ${\prec}(a)$ because $p = \bigwedge\{a_m \mid m \in \mathbb{N}\} < a_{i+1}$. So we have $a \preceq a_i$, and as i was chosen arbitrarily, a is a lower bound of $\{a_m \mid m \in \mathbb{N}\}$. This implies $a \leq \bigwedge\{a_m \mid m \in \mathbb{N}\} = p$, hence $\langle s, p, p\rangle \in \mathrm{NB}$ by Proposition 4.8. □

Proposition 4.44. *If $\mathcal{J}(\boldsymbol{\mathcal{L}})$ does not have minimal elements then* $\mathrm{AG}_2^{\mathrm{c}}$ *is unsound.*

Proof. Let $\preceq$ be the order on $\mathcal{J}(\boldsymbol{\mathcal{L}})$ which is induced by $\leq$. Assume that there are no minimal elements in $\mathcal{J}(\boldsymbol{\mathcal{L}})$ and note that $\prec(a) \neq \emptyset$ for all $a \in \mathcal{J}(\boldsymbol{\mathcal{L}})$. Define the valuation α with $dom(\alpha) = \{\mathsf{S}_1, \mathsf{S}_2, \mathsf{P}_2, \mathsf{P}_2\}$ by $\alpha(\mathsf{S}_1) = \alpha(\mathsf{S}_2) = 1$ and $\alpha(\mathsf{P}_1) = \alpha(\mathsf{P}_2) = 0$. Obviously, the premises of $\mathrm{AG}_2^{\mathrm{c}}$ do not entail the conclusion under α, so by Proposition 3.40 again, unsoundness follows provided that the side condition $\mathrm{NB}[\top, \mathsf{P}_1, \mathsf{P}_2]$ is true under α. By Proposition 4.8 this is the case, i. e., $\langle 1, 0, 0\rangle \in \mathrm{NB}$, because there is no $a \in \mathcal{J}(\boldsymbol{\mathcal{L}})$ such that $0 \wedge 0$ is an upper bound of $\prec(a)$; otherwise $\prec(a)$ would have to be empty for some $a \in \mathcal{J}(\boldsymbol{\mathcal{L}})$. □

Again, unsoundness is inherited to stronger rules. I. e., if the suborder which is generated by $\mathcal{J}(\boldsymbol{\mathcal{L}})$ is forest-like and does not satisfy DCC then AG_2 is unsound. And if $\mathcal{J}(\boldsymbol{\mathcal{L}})$ does not have minimal elements then both AG_2 and $\mathrm{AG}_2^{\mathrm{c}\prime}$ are unsound. Note that the lack of minimal elements in $\mathcal{J}(\boldsymbol{\mathcal{L}})$ implies that the suborder of $\boldsymbol{\mathcal{L}}$ which is generated by $\mathcal{J}(\boldsymbol{\mathcal{L}})$ does not satisfy DCC.

To demonstrate that the additional requirements of the above two propositions are not unrealistic, consider the lattice $\boldsymbol{\mathcal{L}} = \langle \mathbb{Z}_\perp^\top, \leq\rangle$ from Example 4.6. Then the set of completely join-irreducible elements, $\mathbb{Z}$, does not have a minimal element, and the suborder generated by $\mathbb{Z}$ is trivially forest-like and does not satisfy DCC. Thus, $\langle \mathbb{Z}_\perp^\top, \leq\rangle$ satisfies the requirements of the propositions 4.43 and 4.44.

Of course, the results of this section do not out-rule the possibility that there are circular assume-guarantee rules that are sound in non-well-approximable lattices. Firstly, it remains open whether any of the presented A-G rules remain sound in lattices that do not meet the requirements of the propositions 4.43 and 4.44 although they fail to be well-approximable because the suborder generated by the completely join-irreducible elements does not satisfy DCC. Secondly, there may be A-G rules with other side conditions, which may ensure soundness in certain non-well-approximable lattices. For instance, A-G rules whose side condition fixes the global guarantees to 1 will always be sound, regardless of the properties of the lattice. It remains open whether there are more interesting examples of circular A-G rules which are sound in non-well-approximable lattices.

Chapter 5

Instantiations

The assume-guarantee rules from Chapter 4 are formulated for a very abstract setting. The systems and properties to be verified are elements of a complete well-approximable lattice, which do not have internal structure but must satisfy certain non-blockingness conditions. In real-life verification, problems are different. Usually, systems and properties do not form a complete well-approximable lattice, and even if they do, it is unclear how to prove non-blockingness conditions efficiently — they may even be undecidable, see Example 4.16. This prevents application of the assume-guarantee rules from Chapter 4 in practice.

As we have outlined in the introduction, it is natural to assume that systems and properties form a meet-semilattice. It is well-known that every semilattice can be embedded into various complete lattices. Thus, it may be possible to embed the semilattice of systems and properties into a complete and well-approximable lattice. If this is the case then we will be able to apply the A-G rules from Chapter 4 to the given systems and properties provided that we can establish the truth of their side conditions, which boils down to establishing non-blockingness. In reality, systems and properties are not black boxes but do have internal structure, and under certain conditions this structure may greatly simplify (or even trivialize) establishing non-blockingness. Thus, we may specialize the abstract A-G rules from Chapter 4 to the concrete setting by replacing the non-blockingness side conditions with other conditions which are (hopefully) decidable and which imply non-blockingness (using the internal structure of systems and properties). Note that because of this implication, the specialized A-G rule will be weaker than the original rule, so it inherits soundness.

The above paragraph sketches the two main ideas how to obtain sound circular assume-guarantee rules for concrete settings by instantiating/specializing the abstract assume-guarantee framework from Chapter 4. To summarize, such an instantiation comprises

- a complete and well-approximable lattice into which the meet-semilattice of systems and properties is embedded, and

- an A-G rule for that lattice which is weaker than one of the A-G rules from Chapter 4.

Circular assume-guarantee rules for various formalisms have been published for more than 20 years now. Common to all of them is an inductive argument to resolve the apparent circularity and establish soundness, similar to the one we used to establish soundness of AG_2 (cf. Lemma 4.18). Together with the abstractness of our framework, this fact suggests that many known assume-guarantee rules might be special instances of the framework. In this chapter, we support this claim by instantiating the framework into two directions:

- system-oriented (see Section 5.1), where the goal is to prove that some composite transition system refines another composite transition system, and

- specification-oriented (see Section 5.2) where one has to prove that a conjunction of assume-guarantee specifications entails another assume-guarantee specification.

In both directions, we derive rules that are known from the literature as well as variations or generalizations of these, which were not known before. It is important to note that proving a rule to be an instance of our framework does not require a circularity-breaking induction any more. Thus for the existing rules, we have reduced their original (and sometimes obscure) soundness proofs to the rather intelligible soundness proof of our rule AG_2.

5.1 Rules For Moore Machines

Behavioral systems, e. g., hardware circuits, device driver software or graphical user interfaces, are often modeled as transition systems. Their specifications may also be given as transition systems, in which case proving conformance of the system w. r. t. the specification amounts to checking that the system refines the specification. What we mean by 'the system refines the specification' depends on the notion of refinement, of which there exist many, for instance trace containment, trace tree containment and simulation.

In the realm of concurrent systems, the complex transition systems, which form the system and specification to be verified, are composed of simpler ones running in parallel. Often, each specification component is thought of as an abstraction of some system component resp. each system component is thought of as an implementation of some specification component. If each implementation refines its abstraction then we can conclude that the system as a whole refines the specification (provided that the notion of refinement is a reasonable, i. e., compositional one). However, as the system components are designed to interact with each other, in isolation they will usually fail to refine their abstractions. Yet, when composed with abstractions of the other components, each system component may refine its abstraction. This is a circular situation, so it is not immediately clear whether we can conclude that the system as a whole refines the specification — only a circular assume-guarantee rule can yield the desired conclusion. Such rules are known for different notions of refinement, e. g., trace containment [AH99] and simulation [HQRT02], and for various presentations of transition systems, e. g., Mealy machines [McM97] and timed systems [TAKB96].

In this section, we are going to establish soundness of a circular assume-guarantee rule for another presentation of transition systems, Moore machines, with trace containment as refinement. We do so by showing it to be an instance of the assume-guarantee framework from Chapter 4. To this end, we define a special semantics of Moore machines, the partial trace languages, which form a complete and well-approximable lattice. For this lattice, we derive an A-G rule from the rule $\mathrm{AG}_2^{\mathrm{c}}$, see Section 4.6. In exactly the same way, we then derive a second A-G rule for Moore machines, which is based on a different semantics, the partial trace tree languages. Thus, we obtain a second circular assume-guarantee rule for Moore machines, with trace tree containment as refinement.

5.1.1 Moore Machines

A Moore machine (see for instance [HQRT02]) is a state transition system with input and output ports. Transitions depend on the current state and the current values of the input ports. The transition relation must be non-terminating (i. e., for each state and all possible input values there is at least one successor state) but need not be deterministic (i. e., exactly one successor state). The value of

each output port only depends on the current state, but is independent of the current input values.

Definition 5.1 (Port, Value, Assignment, Partial Assignment).
We fix a finite non-empty set $\mathcal{X}$, the set of *ports*, and a (possibly infinite) non-empty set $\mathcal{D}$, the domain of *values*. An *assignment* is a total map from $\mathcal{X}$ to $\mathcal{D}$ and a *partial assignment* is a partial map from $\mathcal{X}$ to $\mathcal{D}$. We denote the set of assignments by Σ and the set of partial assignments by $\hat{\Sigma}$.

Formally, $\Sigma = \mathcal{X} \to \mathcal{D}$ and $\hat{\Sigma} = \mathcal{X} \rightharpoonup \mathcal{D}$. Recall that $\Sigma \subseteq \hat{\Sigma}$, that $\trianglelefteq$ is an order on $\hat{\Sigma}$, that $\bot$ is the least element in $\langle \hat{\Sigma}, \trianglelefteq \rangle$ and that Σ forms the set of maximal elements.

Definition 5.2 (Moore machine).
A *Moore machine* M is a six-tuple $\langle I, O, S, \iota, \delta, \rho \rangle$, where

- $I \subseteq \mathcal{X}$ is the set of *input ports*,
- $O \subseteq \mathcal{X}$ is the set of *output ports*,
- S is the (possibly infinite) *state space*,
- $\iota \in \mathcal{P}(S) \setminus \{\emptyset\}$ is the set of *initial states*,
- $\delta : S \times \Sigma|_I \to \mathcal{P}(S) \setminus \{\emptyset\}$ is the *transition function*, and
- $\rho : S \to \Sigma|_O$ is the *output function*,

such that $I \cap O = \emptyset$, i. e., the sets of input and output ports are disjoint.

Note that the state space of a Moore machine is always a non-empty set as it contains a non-empty subset of initial states. Contrary to [HQRT02], we do not demand finite non-determinism, i. e., the set of initial states may be infinite and for all states $s \in S$ and all inputs $a \in \Sigma|_I$, the set of successor states $\delta(s, a)$ may be infinite.

When composing two Moore machines in parallel, inputs and outputs may be 'cross-wired', i. e., input ports of one machine may (but need not) be output ports of the other, and vice versa. In order to avoid conflicts when two machines in parallel share the same output port, we prohibit sharing of output ports.

Definition 5.3 (Compatibility, Parallel Composition).
Given Moore machines $M_1 = \langle I_1, O_1, S_1, \iota_1, \delta_1, \rho_1 \rangle$ and $M_2 = \langle I_2, O_2, S_2, \iota_2, \delta_2, \rho_2 \rangle$, we call M_1 and M_2 *compatible* if and only if $O_1 \cap O_2 = \emptyset$. If M_1 and M_2 are compatible then the *parallel composition* (or just *composition*) of M_1 and M_2, denoted by $M_1 \parallel M_2$, exists and is defined as the six-tuple $\langle I, O, S, \iota, \delta, \rho \rangle$, where

- $O = O_1 \cup O_2 \subseteq \mathcal{X}$,

- $I = (I_1 \cup I_2) \setminus O \subseteq \mathcal{X}$,
- $S = S_1 \times S_2$,
- $\iota = \iota_1 \times \iota_2 \in \mathcal{P}(S) \setminus \{\emptyset\}$,
- $\delta : S \times \Sigma|_I \to \mathcal{P}(S) \setminus \{\emptyset\}$ such that for all $\langle s_1, s_2 \rangle \in S$ and all $a \in \Sigma|_I$, $\delta(\langle s_1, s_2 \rangle, a) = \delta_1\big(s_1, \big(a \triangledown \rho_2(s_2)\big)|_{I_1}\big) \times \delta_2\big(s_2, \big(a \triangledown \rho_1(s_1)\big)|_{I_2}\big)$, and
- $\rho : S \to \Sigma|_O$ such that for all $\langle s_1, s_2 \rangle \in S$, $\rho(\langle s_1, s_2 \rangle) = \rho_1(s_1) \triangledown \rho_2(s_2)$.

Observe that $M_1 \parallel M_2$ is well-defined due to the following facts:

- $\iota \neq \emptyset$ because $\iota_1 \neq \emptyset$ and $\iota_2 \neq \emptyset$.
- The join $a \triangledown \rho_1(s_1)$ exists because $dom(a) \cap dom(\rho_1(s_1)) = I \cap O_1 = \emptyset$. Similarly, $a \triangledown \rho_2(s_2)$ exists.
- $\big(a \triangledown \rho_2(s_2)\big)|_{I_1} \in \Sigma|_{I_1}$ because $I_1 \subseteq (I_1 \setminus O_2) \cup O_2 = (I_1 \setminus O) \cup O_2 \subseteq I \cup O_2 = dom(a \triangledown \rho_2(s_2))$. Similarly, $\big(a \triangledown \rho_1(s_1)\big)|_{I_2} \in \Sigma|_{I_2}$.
- The product $\delta(\langle s_1, s_2 \rangle, a)$ is non-empty because $\delta_1\big(s_1, \big(a \triangledown \rho_2(s_2)\big)|_{I_1}\big)$ and $\delta_2\big(s_2, \big(a \triangledown \rho_1(s_1)\big)|_{I_2}\big)$ are non-empty.
- The join $\rho_1(s_1) \triangledown \rho_2(s_2)$ exists as $dom(\rho_1(s_1)) \cap dom(\rho_2(s_2)) = O_1 \cap O_2 = \emptyset$.
- $\rho_1(s_1) \triangledown \rho_2(s_2) \in \Sigma|_O$ because $O = O_1 \cup O_2 = dom(\rho_1(s_1) \triangledown \rho_2(s_2))$.

Furthermore $I \cap O = \emptyset$, which proves the following proposition.

Proposition 5.4. *Let M_1 and M_2 be compatible Moore machines. Then $M_1 \parallel M_2$ is a Moore machine.*

5.1.2 Linear-Time Semantics

In this section, we will provide a linear-time semantics for Moore machines, i. e., we will define the language of a Moore machine as the set of finite words which it generates (or accepts). It will turn out, that this semantics is too coarse to instantiate our assume-guarantee framework. Therefore, we will develop a more fine-grained semantics by defining an ordered set of partial words, which can be completed into a well-approximable lattice. For this semantics, we will prove sound an instance of the A-G rule $\mathrm{AG}_2^{\mathrm{c}}$ from Section 4.6. Finally, we will show how to transfer this new A-G rule to the coarser (but more natural) semantics.

Definition 5.5 (Word).
A *word* is an element of Σ^*. We denote the set of all words by W.

Definition 5.6 (Run, Trace, Trace Language).
Given a Moore machine $M = \langle I, O, S, \iota, \delta, \rho \rangle$ and a word $w \in \mathrm{W}$, we call $r \in S^*$ a *run* of w in M if and only if

- $len(r) = len(w) = 0$ or
- $len(r) = len(w) > 0$ and
  - $r_0 \in \iota$,
  - $r_{i+1} \in \delta(r_i, w_i|_I)$ for all $i \in \{0, \ldots, len(r) - 2\}$, and
  - $w_i|_O \trianglelefteq \rho(r_i)$ for all $i \in \{0, \ldots, len(r) - 1\}$.

We call w a *trace* of M if and only if there is $r \in S^*$ such that r is a run of w in M. We define the *trace language* (or just *language*) of M, denoted by $\mathrm{W}(M)$, as the set of all traces of M.

Note that $w_i|_O = \rho(r_i)$ for all positions i of a run r of w in M. The use of the order $\trianglelefteq$ instead of equality in the above definition will become handy later when we define partial traces.

The language of a Moore machine is never empty, as the empty word ϵ is a run of the empty word ϵ.

Proposition 5.7. *Let M be a Moore machine. Obviously, $\epsilon \in \mathrm{W}(M)$.*

Example 5.8. Let $x_1, x_2 \in \mathcal{X}$ be two distinct ports and assume that the distinct values 0, 1, 2 are contained in the domain of values $\mathcal{D}$. Let M_1 and M_2 be Moore machines with $M_i = \langle \emptyset, \{x_i\}, \{\bullet\}, \{\bullet\}, \delta_i, \rho_i \rangle$ and $\delta_i(\bullet, \bot) = \{\bullet\}$ and $\rho_i(\bullet)(x_i) = i$ for all $i \in \{1, 2\}$. Then $\mathrm{W}(M_i) = \{w \in \mathrm{W} \mid w_j(x_i) = i \text{ for all } j < len(w)\}$ for all $i \in \{1, 2\}$.

Soon, we will see that the language of a Moore machine is prefix-closed (this will follow from Lemma 5.16). Now, the prefix-closed subsets of W form a complete and well-approximable lattice (as the completely join-irreducible elements are the prefix-closures of words in W). Thus, this lattice suggests itself for instantiating the framework from Chapter 4. However, $\mathrm{W}(M_1)$ and $\mathrm{W}(M_2)$ block each other as ϵ is contained in both but no word w with $len(w) = 1$ and $w_0(x_1) = w_0(x_2) = 0$ is contained in either language. This shows that we cannot establish our central notion of non-blockingness, not even for the languages of the simplest systems (no input ports, just one output port and one state) and not even when they are compatible (as M_1 and M_2 are) and completely unrelated.

To summarize the above example, the trace language of a Moore machine is too coarse to enable non-blockingness because for each trace, the indivisible step of extending the trace (by appending a new letter) requires to assign values simultaneously to all ports. The obvious way out is to break up this simultaneous

assignment into successive assignments of values to individual ports. As intermediate results, we get traces whose last letter is a partial assignment in $\hat{\Sigma}$. This partial trace language of a Moore machine is a superset of the trace language. Moreover, the indivisible step of extending a partial trace requires to assign just one value to one port, which will later enable us to prove that the partial trace languages of compatible Moore machines do not block each other (see Proposition 5.19).

Just like traces are defined to be words, partial traces are defined to be partial words. Now, we introduce partial words formally.

Definition 5.9 (Partial Word).
We call $w \in \hat{\Sigma}^*$ a *partial word* if and only if all but the last letter of w are in Σ. We denote the set of all partial words by $\hat{\mathrm{W}}$, i. e., formally $\hat{\mathrm{W}} = \{\epsilon\} \cup \Sigma^*\hat{\Sigma}$.

Note that $\mathrm{W} \subset \hat{\mathrm{W}} \subset \hat{\Sigma}^* \subset \hat{\Sigma}^\infty$ and recall that $\trianglelefteq^\infty$ is the order on $\hat{\Sigma}^\infty$ which originates from lifting $\trianglelefteq$.

Notation. We use the same symbol $\trianglelefteq^\infty$ to denote the order on $\hat{\mathrm{W}}$ which is induced by the order $\trianglelefteq^\infty$ on $\hat{\Sigma}^\infty$. We denote the corresponding strict order on $\hat{\mathrm{W}}$ by $\triangleleft^\infty$. Given $w \in \hat{\mathrm{W}}$, we may write $\trianglelefteq^\infty(w)$ to denote the downward closure of w in $\langle \hat{\mathrm{W}}, \trianglelefteq^\infty \rangle$, i. e., $\trianglelefteq^\infty(w) = \{v \in \hat{\mathrm{W}} \mid v \trianglelefteq^\infty w\}$. By $\triangleleft^\infty(w)$, we denote the strict downward closure of w, i. e., $\triangleleft^\infty(w) = \trianglelefteq^\infty(w) \setminus \{w\}$. Given $S \subseteq \hat{\mathrm{W}}$, we may write $\trianglelefteq^\infty(S)$ to abbreviate the order ideal generated by S, i. e., $\trianglelefteq^\infty(S) = \bigcup_{w \in S} \trianglelefteq^\infty(w)$. Recall that $\trianglelefteq^\infty(S)$ is the least order ideal containing S, i. e., we have the equality $\trianglelefteq^\infty(S) = \bigcap\{X \in \mathcal{O}(\langle \hat{\mathrm{W}}, \trianglelefteq^\infty \rangle) \mid S \subseteq X\}$.

The downward closures of partial words will turn out to be of particular importance. Therefore, the following two lemmas provide more information; the first lemma characterizes the maximal elements in the strict downward closure of a partial word, which is finite by the second lemma.

Lemma 5.10. *Let $v, w \in \hat{\mathrm{W}}$. Then v is maximal in $\triangleleft^\infty(w)$ if and only if*

1. *$w = v\bot$, or*
2. *there are $u \in \Sigma^*$, $a, b \in \hat{\Sigma}$ and $x \in dom(b)$ such that $v = ua$, $w = ub$ and $a = b|_{dom(b)\setminus\{x\}}$.*

Proof. To prove the forward direction, assume that v is maximal in $\triangleleft^\infty(w)$, so $v \trianglelefteq^\infty w$, i. e., $len(v) \leq len(w)$ and $v_i \trianglelefteq w_i$ for all $i < len(v)$, and $v \neq w$. Let $n = len(v)$. Case distinction.

- $n < len(w)$. By maximality of v, $len(w) = n + 1$, otherwise we would have $w_0 \dots w_n \in \hat{\mathrm{W}}$ and $v \triangleleft^\infty w_0 \dots w_n \triangleleft^\infty w$. Also by maximality, $w_n = \bot$, otherwise we would have $w_0 \dots w_{n-1}\bot \in \hat{\mathrm{W}}$ and $v \triangleleft^\infty w_0 \dots w_{n-1}\bot \triangleleft^\infty w$. Thus, 1 holds.

- $n = len(w)$. Then $n > 0$ as $v \neq w$. Let $u = v_0 \dots v_{n-2}$, $a = v_{n-1}$ and $b = w_{n-1}$. Then $v = ua$ and as $v \in \hat{\mathrm{W}}$, $u \in \Sigma^*$ and $a \in \hat{\Sigma}$. So as $u_i \in \Sigma$ and $u_i = v_i \trianglelefteq w_i$ for all $i \in \{0, \dots, n-2\}$, we have $u = w_0 \dots w_{n-2}$. Thus, we get $w = ub$ and $b \in \hat{\Sigma}$. Furthermore, $ua \lhd^\infty ub$ implies $a \lhd b$, so there is $x \in dom(b) \setminus dom(a)$. By maximality of v, $a = b|_{dom(b)\setminus\{x\}}$; otherwise there were $c \in \hat{\Sigma}$, namely $c = b|_{dom(b)\setminus\{x\}}$, such that $ua \lhd^\infty uc \lhd^\infty ub$. Thus, 2 holds.

To prove the reverse direction, assume that 1 or 2 holds. Then obviously, $v \lhd^\infty w$ and there is no $v' \in \hat{\mathrm{W}}$ such that $v \lhd^\infty v' \lhd^\infty w$. Hence, v is maximal in $\lhd^\infty(w)$. □

Lemma 5.11. *Let $w \in \hat{\mathrm{W}}$. Then $\trianglelefteq^\infty(w)$ is finite.*

Proof. By induction on $len(w)$.

- $len(w) = 0$. Then $w = \epsilon$ and $\trianglelefteq^\infty(w) = \{\epsilon\}$ is finite.
- $len(w) > 0$. Then there are $v \in \Sigma^*$ and $b \in \hat{\Sigma}$ such that $w = vb$. As $v \in \hat{\mathrm{W}}$ and $v \trianglelefteq^\infty w$, we have $\trianglelefteq^\infty(w) = \trianglelefteq^\infty(v) \cup \big(\trianglelefteq^\infty(w) \setminus \trianglelefteq^\infty(v)\big)$. By induction hypothesis, $\trianglelefteq^\infty(v)$ is finite. Moreover, finiteness of $\mathcal{X}$ implies that the set $\{a \in \hat{\Sigma} \mid a \trianglelefteq b\}$ is finite, so $\trianglelefteq^\infty(w) \setminus \trianglelefteq^\infty(v) = \{va \mid a \in \hat{\Sigma}, a \trianglelefteq b\}$ is finite, too. □

In order to instantiate the assume-guarantee rules from Chapter 4 we need a complete and well-approximable lattice. We obtain a complete lattice via an ideal completion and we show that the completely join-irreducible elements of this lattice generate a well-founded approximation.

Definition 5.12 (Ideal Completion of Partial Words.).
We denote the ideal completion of $\langle \hat{\mathrm{W}}, \trianglelefteq^\infty \rangle$ by $\boldsymbol{\mathcal{L}_{\mathbf{W}}}$, so formally $\boldsymbol{\mathcal{L}_{\mathbf{W}}} = \langle \mathcal{L}_{\mathrm{W}}, \subseteq \rangle$ where $\mathcal{L}_{\mathrm{W}} = \mathcal{O}(\langle \hat{\mathrm{W}}, \trianglelefteq^\infty \rangle)$.

Lemma 5.13. $\mathcal{J}(\boldsymbol{\mathcal{L}_{\mathbf{W}}}) = \{\trianglelefteq^\infty(w) \mid w \in \hat{\mathrm{W}}\}$.

Proof. In every ideal completion, the completely join-irreducible elements are the principal ideals — cf. the paragraph on order ideals in Chapter 2. □

Proposition 5.14. $\boldsymbol{\mathcal{L}_{\mathbf{W}}}$ *is well-approximable.*

Proof. Let $\boldsymbol{\mathcal{J}}$ be the suborder of $\boldsymbol{\mathcal{L}_{\mathbf{W}}}$ which is generated by the completely join-irreducible elements, i. e., $\mathcal{J} = \mathcal{J}(\boldsymbol{\mathcal{L}_{\mathbf{W}}})$. By Lemma 5.13, $\mathcal{J}$ is the set of principal ideals in $\langle \hat{\mathrm{W}}, \trianglelefteq^\infty \rangle$, which is join-dense in $\boldsymbol{\mathcal{L}_{\mathbf{W}}}$ because in every ideal completion, the set of principal ideals is join-dense — cf. the paragraph on order ideals in Chapter 2. Moreover, $\boldsymbol{\mathcal{J}}$ satisfies DCC because every principal ideal in $\langle \hat{\mathrm{W}}, \trianglelefteq^\infty \rangle$ is finite by Lemma 5.11, so every infinite descending sequence of principal ideals must eventually stabilize. Hence, $\boldsymbol{\mathcal{J}}$ is a well-founded approximation of $\boldsymbol{\mathcal{L}_{\mathbf{W}}}$. □

Now, we extend the definition of traces to partial words, thus obtaining partial traces and partial trace languages.

Definition 5.15 (Partial Trace, Partial Trace Language).
Given a Moore machine $M = \langle I, O, S, \iota, \delta, \rho\rangle$, we call $w \in \hat{\mathrm{W}}$ a *partial trace* of M if and only if there is $r \in S^*$ such that r is a run of w in M. We define the *partial trace language* of M, denoted by $\hat{\mathrm{W}}(M)$, as the set of all partial traces of M.

Observe that the underlying definition of run need not be extended as it is already general enough to cope with partial traces. Given a run r of a partial trace w in M with $n = \mathit{len}(r)$, we have $w_i|_O = \rho(r_i)$ for all $i \in \{0, \ldots, n-2\}$, but on the last state of the run we may have $w_{n-1}|_O \lhd \rho(r_{n-1})$. This explains why we use of the order $\trianglelefteq$ in the definition of runs.

We investigate the relationship between the trace and the partial trace language of a Moore machine. It turns out that the partial trace language is the order ideal generated by the trace language, and the trace language consists of exactly those partial traces that happen to be words. From this characterization follows that trace language containment between two Moore machines is equivalent to partial trace language containment.

Lemma 5.16. *Let M be a Moore machine. Then $\mathrm{W}(M) = \mathrm{W} \cap \hat{\mathrm{W}}(M)$ and $\hat{\mathrm{W}}(M) = \trianglelefteq^\infty(\mathrm{W}(M))$.*

Proof. Let $M = \langle I, O, S, \iota, \delta, \rho\rangle$. We prove the claim $\hat{\mathrm{W}}(M) = \trianglelefteq^\infty(\mathrm{W}(M))$ first. Recall that $\trianglelefteq^\infty(\mathrm{W}(M)) = \bigcup\{\trianglelefteq^\infty(w) \mid w \in \mathrm{W}(M)\}$. Given $w \in \hat{\mathrm{W}}(M)$, there is a run $r \in S^*$ of w in M. Obviously, there exists $w' \in \mathrm{W}$ with $\mathit{len}(w') = \mathit{len}(w)$ and $w \trianglelefteq^\infty w'$ such that r is run of w' in M. Thus $w' \in \mathrm{W}(M)$, and therefore $w \in \trianglelefteq^\infty(w') \subseteq \trianglelefteq^\infty(\mathrm{W}(M))$. Now given $w \in \trianglelefteq^\infty(\mathrm{W}(M))$, there is $w' \in \mathrm{W}(M)$ such that $w \trianglelefteq^\infty w'$. There exists a run $r \in S^*$ of w' in M. Then obviously, $r_0 \ldots r_{n-1}$ is a run of w in M, where $n = \mathit{len}(w)$. Thus, $w \in \hat{\mathrm{W}}(M)$.

Now, we show that $\mathrm{W}(M) = \mathrm{W} \cap \hat{\mathrm{W}}(M)$. The left-to-right inclusion holds because $\mathrm{W}(M) \subseteq \mathrm{W}$ by definition and $\mathrm{W}(M) \subseteq \trianglelefteq^\infty(\mathrm{W}(M)) = \hat{\mathrm{W}}(M)$. The right-to-left inclusion holds by the definition of traces. □

Proposition 5.17. *Let M and M' be Moore machines. Then $\mathrm{W}(M) \subseteq \mathrm{W}(M')$ if and only if $\hat{\mathrm{W}}(M) \subseteq \hat{\mathrm{W}}(M')$.*

Proof. Recall that $\trianglelefteq^\infty(\mathrm{W}(M)) = \bigcup\{\trianglelefteq^\infty(w) \mid w \in \mathrm{W}(M)\}$, so by Lemma 5.16, $\mathrm{W}(M) \subseteq \mathrm{W}(M')$ implies $\hat{\mathrm{W}}(M) = \trianglelefteq^\infty(\mathrm{W}(M)) \subseteq \trianglelefteq^\infty(\mathrm{W}(M')) = \hat{\mathrm{W}}(M')$. The reverse direction holds because again by Lemma 5.16, $\hat{\mathrm{W}}(M) \subseteq \hat{\mathrm{W}}(M')$ implies $\mathrm{W}(M) = \mathrm{W} \cap \hat{\mathrm{W}}(M) \subseteq \mathrm{W} \cap \hat{\mathrm{W}}(M') = \mathrm{W}(M')$. □

In partial trace languages, composition corresponds to intersection. More precisely, the partial trace language of the parallel composition of two compatible Moore machines is the intersection of the two partial trace languages.

Proposition 5.18. *Let M_1 and M_2 be two compatible Moore machines. Then* $\hat{\mathrm{W}}(M_1 \parallel M_2) = \hat{\mathrm{W}}(M_1) \cap \hat{\mathrm{W}}(M_2)$.

Proof. Let $M_1 = \langle I_1, O_1, S_1, \iota_1, \delta_1, \rho_1\rangle$ and $M_2 = \langle I_2, O_2, S_2, \iota_2, \delta_2, \rho_2\rangle$ and note that $O_1 \cap O_2 = \emptyset$. Let $M_1 \parallel M_2 = \langle I, O, S, \iota, \delta, \rho\rangle$. Let $w \in \hat{\mathrm{W}}$ and let $r \in S^*$, $r^1 \in S_1^*$ and $r^2 \in S_2^*$ with $len(w) = len(r) = len(r^1) = len(r^2)$ such that $r_i = \langle r_i^1, r_i^2\rangle$ for all $i < len(r)$. We prove the claim $\hat{\mathrm{W}}(M_1 \parallel M_2) = \hat{\mathrm{W}}(M_1) \cap \hat{\mathrm{W}}(M_2)$ by showing that r is run of w in $M_1 \parallel M_2$ if and only if r^1 and r^2 are runs of w in M_1 and M_2, respectively. If $len(w) = 0$ then this equivalence is trivial, so assume $len(w) > 0$. We have to show the three conjuncts of the second case of the definition of runs.

- $\iota = \iota_1 \times \iota_2$ by definition of $M_1 \parallel M_2$, so $\langle r_0^1, r_0^2\rangle \in \iota$ iff $r_0^1 \in \iota_1$ and $r_0^2 \in \iota_2$.
- Fix an arbitrary $i \in \{0, \ldots, len(r) - 2\}$. By definition of $M_1 \parallel M_2$, we have $\delta(\langle r_i^1, r_i^2\rangle, w_i|_I) = \delta_1\big(r_i^1, \big(w_i|_I \triangledown \rho_2(r_i^2)\big)|_{I_1}\big) \times \delta_2\big(r_i^2, \big(w_i|_I \triangledown \rho_1(r_i^1)\big)|_{I_2}\big)$. We can simplify the right-hand side of this equation since

$$\begin{aligned}\delta_1\big(r_i^1, \big(w_i|_I \triangledown \rho_2(r_i^2)\big)|_{I_1}\big) &= \delta_1\big(r_i^1, w_i|_{I_1 \cap I} \triangledown \rho_2(r_i^2)|_{I_1}\big)\\ &= \delta_1\big(r_i^1, w_i|_{I_1 \setminus O_2} \triangledown \rho_2(r_i^2)|_{I_1}\big)\\ &= \delta_1(r_i^1, w_i|_{I_1 \setminus O_2} \triangledown w_i|_{I_1 \cap O_2})\\ &= \delta_1(r_i^1, w_i|_{I_1}),\end{aligned}$$

  where the third equality holds because $w_i|_{O_2} = \rho_2(r_i^2)$, and the second equality holds because

$$\begin{aligned}I_1 \cap I &= I_1 \cap \big((I_1 \cup I_2) \setminus O\big) = \big((I_1 \cap I_1) \setminus O\big) \cup \big((I_1 \cap I_2) \setminus O\big)\\ &= I_1 \setminus O = I_1 \setminus (O_1 \cup O_2) = I_1 \setminus O_2\end{aligned}$$

  With similar arguments, we get $\delta_2\big(r_i^2, \big(w_i|_I \triangledown \rho_1(r_i^1)\big)|_{I_2}\big) = \delta_2(r_i^2, w_i|_{I_2})$. So, $\langle r_{i+1}^1, r_{i+1}^2\rangle \in \delta(\langle r_i^1, r_i^2\rangle, w_i|_I)$ iff $r_{i+1}^1 \in \delta_1(r_i^1, w_i|_{I_1})$ and $r_{i+1}^2 \in \delta_2(r_i^2, w_i|_{I_2})$.
- Fix an arbitrary $i \in \{0, \ldots, len(r)-1\}$. We have $\rho(\langle r_i^1, r_i^2\rangle) = \rho_1(r_i^1) \triangledown \rho_2(r_i^2)$ by definition of $M_1 \parallel M_2$. Therefore

$$\begin{aligned}w_i|_O \trianglelefteq \rho(\langle r_i^1, r_i^2\rangle) \quad &\text{iff} \quad w_i|_{O_1} \triangledown w_i|_{O_2} \trianglelefteq \rho_1(r_i^1) \triangledown \rho_2(r_i^2)\\ &\text{iff} \quad w_i|_{O_1} \trianglelefteq \rho_1(r_i^1) \text{ and } w_i|_{O_2} \trianglelefteq \rho_2(r_i^2),\end{aligned}$$

  where the second equivalence holds because $O_1 \cap O_2 = \emptyset$. □

Next, we show how partial trace languages resolve the problem with blockingness from Example 5.8. We prove that the partial trace languages of compatible Moore machines do never block each other.

Proposition 5.19. *Let M_1 and M_2 be Moore machines. If M_1 and M_2 are compatible then* $\langle \hat{\mathrm{W}}, \hat{\mathrm{W}}(M_1), \hat{\mathrm{W}}(M_2)\rangle \in \mathrm{NB}$*, i. e., $\hat{\mathrm{W}}(M_1)$ and $\hat{\mathrm{W}}(M_2)$ do not block each other.*

Proof. Let $M_1 = \langle I_1, O_1, S_1, \iota_1, \delta_1, \rho_1\rangle$ and $M_2 = \langle I_2, O_2, S_2, \iota_2, \delta_2, \rho_2\rangle$ and assume that M_1 and M_2 are compatible, i. e., $O_1 \cap O_2 = \emptyset$. By Proposition 4.8, we have to show that every completely join-irreducible element of $\boldsymbol{\mathcal{L}_{\mathbf{W}}}$ is contained in $\hat{\mathrm{W}}(M_1)$ or in $\hat{\mathrm{W}}(M_2)$ whenever $\hat{\mathrm{W}}(M_1) \cap \hat{\mathrm{W}}(M_2)$ is an upper bound of its strict downward-closure.

By Lemma 5.13, $\mathcal{J}(\boldsymbol{\mathcal{L}_{\mathbf{W}}})$ is the set of principal ideals in $\langle \hat{\mathrm{W}}, \trianglelefteq^\infty\rangle$. We choose an arbitrary $w \in \hat{\mathrm{W}}$ with the property that $\hat{\mathrm{W}}(M_1) \cap \hat{\mathrm{W}}(M_2)$ is an upper bound of $\{\trianglelefteq^\infty(v) \mid v \in \hat{\mathrm{W}}, v \lhd^\infty w\}$, the strict downward closure of $\trianglelefteq^\infty(w)$ in $\langle \mathcal{J}(\boldsymbol{\mathcal{L}_{\mathbf{W}}}), \subseteq\rangle$. We have to show that $\trianglelefteq^\infty(w) \subseteq \hat{\mathrm{W}}(M_1)$ or $\trianglelefteq^\infty(w) \subseteq \hat{\mathrm{W}}(M_2)$. As $\hat{\mathrm{W}}(M_1)$ and $\hat{\mathrm{W}}(M_2)$ are downward-closed by Lemma 5.16, it suffices to show that $w \in \hat{\mathrm{W}}(M_1)$ or $w \in \hat{\mathrm{W}}(M_2)$. Case distinction.

- $w = \epsilon$. Then $w \in \mathrm{W}(M_1) \subseteq \hat{\mathrm{W}}(M_1)$ by Proposition 5.7 and Lemma 5.16.
- $w \neq \epsilon$. Then $\lhd^\infty(w)$ is non-empty and, by Lemma 5.11, finite, so there exists a maximal element v in $\lhd^\infty(w)$. Because of our assumption that $\hat{\mathrm{W}}(M_1) \cap \hat{\mathrm{W}}(M_2)$ is an upper bound of the strict downward closure of $\trianglelefteq^\infty(w)$, we know that $\trianglelefteq^\infty(v) \subseteq \hat{\mathrm{W}}(M_1) \cap \hat{\mathrm{W}}(M_2)$. Lemma 5.10 provides more information about v, which leads to another case distinction.
  - $w = v\bot$.
    As $v \in \hat{\mathrm{W}}(M_1)$ there exists a run $r \in S_1^*$ of v in M_1. We will show that r can be extended to a run $r' \in S_1^*$ of w in M_1. This requires yet another case distinction.
    - $\mathit{len}(r) = 0$. Then $v = \epsilon$ and $w = \bot$. As $\iota_1 \neq \emptyset$, we can construct $r' \in S_1^*$ with $\mathit{len}(r') = 1$ such that $r'_0 \in \iota_1$. Obviously, r' is a run of w in M_1, so $w \in \hat{\mathrm{W}}(M_1)$.
    - $\mathit{len}(r) > 0$. Let $n = \mathit{len}(r)$. Then $\mathit{len}(w) = n + 1$. Since w_{n-1} is not the last letter of w, we have $w_{n-1} \in \Sigma$ by the definition of partial words, so the second argument of $\delta_1(r_{n-1}, w_{n-1}|_{I_1})$ is defined. And as $\delta_1(r_{n-1}, w_{n-1}|_{I_1}) \neq \emptyset$, we can construct $r' \in S_1^*$ with $\mathit{len}(r') = n+1$ and $r = r'_0 \dots r'_{n-1}$ such that $r'_n \in \delta_1(r'_{n-1}, w_{n-1}|_{I_1})$. Obviously, r' is a run of w in M_1, so $w \in \hat{\mathrm{W}}(M_1)$.
  - There are $u \in \Sigma^*$, $a, b \in \hat{\Sigma}$ and $x \in \mathit{dom}(b)$ such that $v = ua$, $w = ub$ and $a = b|_{\mathit{dom}(b)\setminus\{x\}}$.
    Let $n = \mathit{len}(w)$. As O_1 and O_2 are disjoint, there is $i \in \{1, 2\}$ such that $x \notin O_i$. Because $v \in \hat{\mathrm{W}}(M_i)$, there exists a run $r \in S_i^*$ of v in M_i. And as $a = b|_{\mathit{dom}(b)\setminus\{x\}}$ and $x \notin O_i$, we have $w_{n-1}|_{O_i} = b|_{O_i} = a|_{O_i} = v_{n-1}|_{O_i} \trianglelefteq \rho_i(r_{n-1})$. Therefore, r is also a run of w in M_i, so $w \in \hat{\mathrm{W}}(M_i)$. □

Now, we are ready to instantiate the A-G rule $\mathrm{AG}_2^{\mathrm{c}}$ from Section 4.6 to our lattice $\boldsymbol{\mathcal{L}_{\mathbf{W}}}$ in order to derive a circular A-G rule for partial trace languages. To this end,

we introduce the binary relation MC_W, which will function as the side condition of the derived rule.

Definition 5.20 (Moore-compatibility).
We say that two elements $S_1, S_2 \in \mathcal{L}_\mathrm{W}$ are *Moore-compatible* (in $\boldsymbol{\mathcal{L}_\mathbf{W}}$) if and only if there are compatible Moore machines M_1 and M_2 such that $S_1 = \hat{\mathrm{W}}(M_1)$ and $S_2 = \hat{\mathrm{W}}(M_2)$. We define the binary relation MC_W on $\mathcal{L}_\mathrm{W}$ by $\langle S_1, S_2\rangle \in \mathrm{MC}_\mathrm{W}$ if and only if S_1 and S_2 are Moore-compatible (in $\boldsymbol{\mathcal{L}_\mathbf{W}}$).

Recall the variables $\mathsf{S}_1, \mathsf{S}_2, \mathsf{P}_1, \mathsf{P}_2 \in \mathcal{V}$ from Section 4.6.

Theorem 5.21. *The A-G rule* $\mathrm{AG}_\mathrm{W}^\mathrm{Moore}$ *is weaker than* AG_2^c *and sound, where*

$$\mathrm{AG}_\mathrm{W}^\mathrm{Moore} : \frac{\mathsf{P}_2 \sqcap \mathsf{S}_1 \sqsubseteq \mathsf{P}_1 \qquad \mathsf{P}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_2}{\mathsf{S}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2} \text{ if } \mathrm{MC}_\mathrm{W}[\mathsf{P}_1, \mathsf{P}_2].$$

Proof. That $\mathrm{AG}_\mathrm{W}^\mathrm{Moore}$ is an A-G rule is obvious. We show that $\mathrm{AG}_\mathrm{W}^\mathrm{Moore}$ is weaker than AG_2^c using Corollary 3.36. As substitution we take $\mathrm{id}|_{\{\mathsf{S}_1,\mathsf{S}_2,\mathsf{P}_1,\mathsf{P}_2\}}$. Then the conditions 1 and 2 are trivially true. And as $\mathrm{MC}_\mathrm{W}[\mathsf{P}_1, \mathsf{P}_2] \models^{\boldsymbol{\mathcal{L}_\mathbf{W}}} \mathrm{NB}[\top, \mathsf{P}_1, \mathsf{P}_2]$ by definition of Moore-compatibility and Proposition 5.19, condition 3 is also true.

By Proposition 3.39, soundness of $\mathrm{AG}_\mathrm{W}^\mathrm{Moore}$ follows from soundness of AG_2^c, see Theorem 4.37. Note that soundness of AG_2^c requires $\boldsymbol{\mathcal{L}_\mathbf{W}}$ to be well-approximable, which it is according to Proposition 5.14. □

Though $\mathrm{AG}_\mathrm{W}^\mathrm{Moore}$ is a circular A-G rule for Moore machines, it is unsatisfactory for two reasons. One reason is the use of a non-standard semantics for Moore machines, the partial trace languages; specifications are usually given w. r. t. the standard semantics, i. e., (in the linear-time setting) as trace languages. The other reason is that $\mathrm{AG}_\mathrm{W}^\mathrm{Moore}$ does not involve parallel composition of systems but only the intersection of partial trace languages. For these reasons, it is not immediately clear whether the rule $\mathrm{AG}_\mathrm{W}^\mathrm{Moore}$ could be used for verification, when the goal is to establish that a parallel composition of two systems refines a parallel composition of two specifications, where refinement means trace language containment. Fortunately, $\mathrm{AG}_\mathrm{W}^\mathrm{Moore}$ can be used to prove soundness of the following more useful rule, which involves parallel composition and standard semantics, i. e., trace languages.

Corollary 5.22. *Let* M_1, M_2, M_1', M_2' *be four Moore machines such that* M_2' *and* M_1, M_1' *and* M_2, M_1 *and* M_2 *as well as* M_1' *and* M_2' *all are compatible. Then* (5.1) *is well-defined and sound.*

$$\frac{\mathrm{W}(M_2' \parallel M_1) \subseteq \mathrm{W}(M_1') \qquad \mathrm{W}(M_1' \parallel M_2) \subseteq \mathrm{W}(M_2')}{\mathrm{W}(M_1 \parallel M_2) \subseteq \mathrm{W}(M_1' \parallel M_2')} \tag{5.1}$$

Proof. As all compositions occuring in premises and conclusion of (5.1) exist due to the stated compatibilities, (5.1) is well-defined. We reduce soundness of (5.1) to soundness of $\mathrm{AG}_{\mathrm{W}}^{\mathrm{Moore}}$ (Theorem 5.21). Assume that the premises of (5.1) hold. By Proposition 5.17, the first premise implies $\hat{\mathrm{W}}(M_2' \parallel M_1) \subseteq \hat{\mathrm{W}}(M_1')$, which implies $\hat{\mathrm{W}}(M_2') \cap \hat{\mathrm{W}}(M_1) \subseteq \hat{\mathrm{W}}(M_1')$ by Proposition 5.18 because M_2' and M_1 are compatible. With similar arguments, the second premise of (5.1) implies $\hat{\mathrm{W}}(M_1') \cap \hat{\mathrm{W}}(M_2) \subseteq \hat{\mathrm{W}}(M_2')$ (because M_1' and M_2 are compatible). And as M_1' and M_2' are compatible, $\hat{\mathrm{W}}(M_1')$ and $\hat{\mathrm{W}}(M_2')$ are surely Moore-compatible in $\boldsymbol{\mathcal{L}}_{\mathbf{W}}$, so by soundness of $\mathrm{AG}_{\mathrm{W}}^{\mathrm{Moore}}$, we can deduce $\hat{\mathrm{W}}(M_1) \cap \hat{\mathrm{W}}(M_2) \subseteq \hat{\mathrm{W}}(M_1') \cap \hat{\mathrm{W}}(M_2')$. This implies $\hat{\mathrm{W}}(M_1 \parallel M_2) \subseteq \hat{\mathrm{W}}(M_1' \parallel M_2')$ by Proposition 5.18 because M_1 and M_2 as well as M_1' and M_2' are compatible. And as $\hat{\mathrm{W}}(M_1 \parallel M_2) \subseteq \hat{\mathrm{W}}(M_1' \parallel M_2')$ implies the conclusion of (5.1) by Proposition 5.17, we are done. □

Frequently, the specification machines M_1' and M_2' are viewed as abstractions of the system machines M_1 and M_2, respectively. If this is the case, it is reasonable to demand that every output port of M_i' is also an output port of M_i for all $i \in \{1, 2\}$, for abstractions should rather exhibit less functionality than more. Under this assumption, we can relax the well-definedness condition of (5.1), so that only compatibility of the system machines M_1 and M_2 is required. Note that this compatibility of the systems is not artificial but very desirable usually because it also ensures *realizability* (i. e., absence of deadlocks) of the composition $M_1 \parallel M_2$.

Corollary 5.23. *Let M_1, M_2, M_1', M_2' be four Moore machines and let O_1, O_2, O_1', O_2' be their respective sets of output ports. If M_1 and M_2 are compatible and $O_1' \subseteq O_1$ and $O_2' \subseteq O_2$ then* (5.1) *is well-defined and sound.*

Proof. Assume that $O_1' \subseteq O_1$ and $O_2' \subseteq O_2$. Then compatibility of M_1 and M_2 (i. e., $O_1 \cap O_2 = \emptyset$) implies compatibility of M_2' and M_1, M_1' and M_2 as well as M_1' and M_2'. Hence (5.1) is well-defined and sound by Corollary 5.22. □

5.1.3 Branching-Time Semantics

Besides linear-time semantics, Moore machines may be given branching-time semantics, for instance by associating with a Moore machine the set of finite trees which it generates. In this section, we will show that our assume-guarantee framework can be instantiated to such a branching-time setting in exactly the same way than to the previous linear-time setting. One by one, we will adapt all the definitions, lemmas and propositions from Section 5.1.2 to trees. All the proofs in this section are rather similar to their linear-time counterparts, and we have replaced proofs by references to Section 5.1.2 if the respective proofs were in exact one-to-one correspondence.

Definition 5.24 (Tree).
A *tree* is an element of $\Sigma^{\blacktriangle}$. We denote the set of all trees by T.

Definition 5.25 (Run Tree, Trace Tree, Trace Tree Language).
Given a Moore machine $M = \langle I, O, S, \iota, \delta, \rho \rangle$ and a tree $t \in \mathrm{T}$, we call $r \in S^{\blacktriangle}$ a *run tree* of t in M if and only if

- $dom(r) = dom(t) = \emptyset$ or
- $dom(r) = dom(t) \neq \emptyset$ and
  - $r_\epsilon \in \iota$,
  - $r_{wn} \in \delta(r_w, t_w|_I)$ for all $\langle w, n \rangle \in \mathbb{N}^* \times \mathbb{N}$ with $wn \in dom(r)$, and
  - $t_w|_O \trianglelefteq \rho(r_w)$ for all $w \in dom(r)$.

We call t a *trace tree* of M if and only if there is $r \in S^{\blacktriangle}$ such that r is a run tree of t in M. We define the *trace tree language* (or just *tree language*) of M, denoted by $\mathrm{T}(M)$, as the set of all trace trees of M.

Note that $t_w|_O = \rho(r_w)$ for all nodes w of a run tree r of t in M. The use of the order $\trianglelefteq$ instead of equality in the above definition will become handy later when we define partial trace trees.

The tree language of a Moore machine is never empty, as the empty tree λ is a run tree of the empty tree λ.

Proposition 5.26. *Let M be a Moore machine. Obviously, $\lambda \in \mathrm{T}(M)$.*

Analogously to Section 5.1.2, we introduce partial trees whose leaf nodes are labeled by partial assignments in $\hat{\Sigma}$.

Definition 5.27 (Partial Tree).
We call $t \in \hat{\Sigma}^{\blacktriangle}$ a *partial tree* if and only if $t_w \in \Sigma$ for all non-leaf nodes w of t. We denote the set of all partial trees by $\hat{\mathrm{T}}$.

Note that $\mathrm{T} \subset \hat{\mathrm{T}} \subset \hat{\Sigma}^{\blacktriangle} \subset \hat{\Sigma}^{\vartriangle}$ and recall that $\trianglelefteq^{\vartriangle}$ is the order on $\hat{\Sigma}^{\vartriangle}$ which originates from lifting $\trianglelefteq$.

Notation. We use the same symbol $\trianglelefteq^{\vartriangle}$ to denote the order on $\hat{\mathrm{T}}$ which is induced by the order $\trianglelefteq^{\vartriangle}$ on $\hat{\Sigma}^{\vartriangle}$. We denote the corresponding strict order on $\hat{\mathrm{T}}$ by $\triangleleft^{\vartriangle}$. Given $t \in \hat{\mathrm{T}}$, we may write $\trianglelefteq^{\vartriangle}(t)$ to denote the downward closure of t in $\langle \hat{\mathrm{T}}, \trianglelefteq^{\vartriangle} \rangle$, i. e., $\trianglelefteq^{\vartriangle}(t) = \{s \in \hat{\mathrm{T}} \mid s \trianglelefteq^{\vartriangle} t\}$. By $\triangleleft^{\vartriangle}(t)$, we denote the strict downward closure of t, i. e., $\triangleleft^{\vartriangle}(t) = \trianglelefteq^{\vartriangle}(t) \setminus \{t\}$. Given $S \subseteq \hat{\mathrm{T}}$, we may also write $\trianglelefteq^{\vartriangle}(S)$ to abbreviate the order ideal generated by S, i. e., $\trianglelefteq^{\vartriangle}(S) = \bigcup_{t \in S} \trianglelefteq^{\vartriangle}(t)$. Recall that $\trianglelefteq^{\vartriangle}(S)$ is the least order ideal containing S, i. e., we have the equality $\trianglelefteq^{\vartriangle}(S) = \bigcap \{X \in \mathcal{O}(\langle \hat{\mathrm{T}}, \trianglelefteq^{\vartriangle} \rangle) \mid S \subseteq X\}$.

The following two lemmas examine the downward closures of partial trees; the first lemma characterizes the maximal elements in the strict downward closure of a partial tree, which is finite by the second lemma.

Lemma 5.28. *Let $s, t \in \hat{\mathrm{T}}$. Then s is maximal in $\lhd^\Delta(t)$ if and only if*

1. *there is $w \in dom(t)$ such that $s = t|_{dom(t)\setminus\{w\}}$ and $t_w = \bot$, or*
2. *$dom(s) = dom(t)$ and there are $w \in dom(t)$ and $x \in dom(t_w)$ such that $s_w = t_w|_{dom(t_w)\setminus\{x\}}$ and $s_v = t_v$ for all $v \in dom(t) \setminus \{w\}$.*

Proof. To prove the forward direction, assume that s is maximal in $\lhd^\Delta(t)$, so $s \trianglelefteq^\Delta t$, i. e., $dom(s) \subseteq dom(t)$ and $s_w \trianglelefteq t_w$ for all $w \in dom(s)$, and $s \neq t$. Case distinction.

- $dom(s) \neq dom(t)$. Then there is $w \in dom(t) \setminus dom(s)$. By maximality of s, $dom(s) = dom(t) \setminus \{w\}$, otherwise there were $r \in \hat{\mathrm{T}}$ with $r = t|_{dom(t)\setminus\{w\}}$ such that $s \lhd^\Delta r \lhd^\Delta t$. Again by maximality, $s_v = t_v$ for all $v \in dom(s)$; otherwise there were $v \in dom(s)$ and $x \in dom(t_v) \setminus dom(s_v)$, so we could construct $r \in \hat{\mathrm{T}}$ with $dom(r) = dom(t)$, $r_v = t_v|_{dom(t_v)\setminus\{x\}}$ and $r_u = t_u$ for all $u \in dom(r) \setminus \{v\}$ such that $s \lhd^\Delta r \lhd^\Delta t$. Thus, we have $s = t|_{dom(t)\setminus\{w\}}$. Finally by maximality of s, $t_w = \bot$, otherwise there were $r \in \hat{\mathrm{T}}$ with $dom(r) = dom(t)$, $r_w = \bot$ and $r_v = t_v$ for all $v \in dom(r) \setminus \{w\}$ such that $s \lhd^\Delta r \lhd^\Delta t$. Thus, 1 holds.
- $dom(s) = dom(t)$. Then there is $w \in dom(s)$ such that $s_w \neq t_w$, i. e., there is $x \in dom(t_w) \setminus dom(s_w)$. By maximality of s, $dom(s_w) = dom(t_w) \setminus \{x\}$, otherwise there were $r \in \hat{\mathrm{T}}$ with $dom(r) = dom(t)$, $r_w = t_w|_{dom(t_w)\setminus\{x\}}$ and $r_v = t_v$ for all $v \in dom(r) \setminus \{w\}$ such that $s \lhd^\Delta r \lhd^\Delta t$. As $s_w \trianglelefteq t_w$, this yields $s_w = t_w|_{dom(t_w)\setminus\{x\}}$. Moreover by maximality of s, $s_v = t_v$ for all $v \in dom(t) \setminus \{w\}$; otherwise there would exist $v \in dom(t) \setminus \{w\}$ and $y \in dom(t_v) \setminus dom(s_v)$, so we could construct $r \in \hat{\mathrm{T}}$ with $dom(r) = dom(t)$, $r_v = t_v|_{dom(t_v)\setminus\{y\}}$ and $r_u = t_u$ for all $u \in dom(r)\setminus\{v\}$ such that $s \lhd^\Delta r \lhd^\Delta t$. Thus, 2 holds.

To prove the reverse direction, assume that 1 or 2 holds. Then obviously, $s \lhd^\Delta t$ and there is no $r \in \hat{\mathrm{T}}$ such that $s \lhd^\Delta r \lhd^\Delta t$. Hence, s is maximal in $\lhd^\Delta(t)$. □

Lemma 5.29. *Let $t \in \hat{\mathrm{T}}$. Then $\trianglelefteq^\Delta(t)$ is finite.*

Proof. By induction on the cardinality of $dom(t)$, which is finite.

- $|dom(t)| = 0$. Then $t = \lambda$ and $\trianglelefteq^\Delta(t) = \{\lambda\}$ is finite.
- $|dom(t)| > 0$. Then $t \neq \lambda$ and as every non-empty finite tree has leaf nodes, the set $L = \{w \in dom(t) \mid w \text{ leaf of } t\}$ is finite and non-empty. We decompose $\trianglelefteq^\Delta(t)$ according to the equation

$$\trianglelefteq^\Delta(t) = \bigcup_{w \in L} \trianglelefteq^\Delta(t|_{dom(t)\setminus\{w\}}) \cup \{s \in \hat{\mathrm{T}} \mid dom(s) = dom(t) \text{ and } s \trianglelefteq^\Delta t\}.$$

Thus, it suffices to show finiteness of each of the sets in the union above. By induction hypothesis, $\trianglelefteq^{\Delta}(t|_{dom(t)\setminus\{w\}})$ is finite for all $w \in L$. To show that the last of the above sets is finite, note that for all $s \in \hat{\mathrm{T}}$ with $dom(s) = dom(t)$, $s \trianglelefteq^{\Delta} t$ if and only if $s|_{dom(t)\setminus L} = t|_{dom(t)\setminus L}$ and $s_w \trianglelefteq t_w$ for all $w \in L$. Hence, the cardinality of $\{s \in \hat{\mathrm{T}} \mid dom(s) = dom(t) \text{ and } s \trianglelefteq^{\Delta} t\}$ is bounded by the product of the cardinalities of L and $\mathcal{X}$. □

In order to instantiate the assume-guarantee rules from Chapter 4 we need a complete and well-approximable lattice. Again, we obtain this lattice via an ideal completion and then show that the completely join-irreducible elements generate a well-founded approximation.

Definition 5.30 (Ideal Completion of Partial Trees.).
We denote the ideal completion of $\langle \hat{\mathrm{T}}, \trianglelefteq^{\Delta} \rangle$ by $\boldsymbol{\mathcal{L}_{\mathbf{T}}}$, so formally $\boldsymbol{\mathcal{L}_{\mathbf{T}}} = \langle \mathcal{L}_{\mathrm{T}}, \subseteq \rangle$ where $\mathcal{L}_{\mathrm{T}} = \mathcal{O}(\langle \hat{\mathrm{T}}, \trianglelefteq^{\Delta} \rangle)$.

Lemma 5.31. $\mathcal{J}(\boldsymbol{\mathcal{L}_{\mathbf{T}}}) = \{\trianglelefteq^{\Delta}(t) \mid t \in \hat{\mathrm{T}}\}$.

Proof. Analogous to the proof of Lemma 5.13. □

Proposition 5.32. $\boldsymbol{\mathcal{L}_{\mathbf{T}}}$ *is well-approximable.*

Proof. Analogous to the proof of Proposition 5.14. □

Now, we extend the definition of trace trees to partial trees, thus obtaining partial trace trees and partial trace tree languages.

Definition 5.33 (Partial Trace Tree, Partial Trace Tree Language).
Given a Moore machine $M = \langle I, O, S, \iota, \delta, \rho \rangle$, we call $t \in \hat{\mathrm{T}}$ a *partial trace tree* of M if and only if there is $r \in S^{\blacktriangle}$ such that r is a run tree of t in M. We define the *partial trace tree language* of M, denoted by $\hat{\mathrm{T}}(M)$, as the set of all partial trace trees of M.

Observe that the underlying definition of run tree need not be extended as it is already general enough to cope with partial trace trees. Given a run tree r of a partial trace tree t in M, $t_w|_O = \rho(r_w)$ for all non-leaf nodes w of r, but there may be leaf nodes w of r with $t_w|_O \lhd \rho(r_w)$. This explains why we use of the order $\trianglelefteq$ in the definition of run trees.

The partial trace tree language is the order ideal generated by the trace tree language, and the trace tree language consists of exactly those partial trace trees that happen to be trees. Thus, trace tree language containment is equivalent to partial trace tree language containment.

Lemma 5.34. *Let M be a Moore machine. Then* $\mathrm{T}(M) = \mathrm{T} \cap \hat{\mathrm{T}}(M)$ *and* $\hat{\mathrm{T}}(M) = \trianglelefteq^{\Delta}(\mathrm{T}(M))$.

Proof. Let $M = \langle I, O, S, \iota, \delta, \rho\rangle$. We prove the claim $\hat{\mathrm{T}}(M) = \trianglelefteq^{\Delta}(\mathrm{T}(M))$ first. Recall that $\trianglelefteq^{\Delta}(\mathrm{T}(M)) = \bigcup\{\trianglelefteq^{\Delta}(t) \mid t \in \mathrm{T}(M)\}$. Given $t \in \hat{\mathrm{T}}(M)$, there is a run tree $r \in S^{\blacktriangle}$ of t in M. Obviously, there exists $t' \in \mathrm{T}$ with $dom(t') = dom(t)$ and $t \trianglelefteq^{\Delta} t'$ such that r is run tree of t' in M. Thus $t' \in \mathrm{T}(M)$, and therefore $t \in \trianglelefteq^{\Delta}(t') \subseteq \trianglelefteq^{\Delta}(\mathrm{T}(M))$. Now given $t \in \trianglelefteq^{\Delta}(\mathrm{T}(M))$, there is $t' \in \mathrm{T}(M)$ such that $t \trianglelefteq^{\Delta} t'$. There exists a run tree $r \in S^{\blacktriangle}$ of t' in M. Then obviously, $r|_{dom(t)}$ is a run tree of t in M, so $t \in \hat{\mathrm{T}}(M)$.

Now, we show that $\mathrm{T}(M) = \mathrm{T} \cap \hat{\mathrm{T}}(M)$. The left-to-right inclusion holds because $\mathrm{T}(M) \subseteq \mathrm{T}$ by definition and $\mathrm{T}(M) \subseteq \trianglelefteq^{\Delta}(\mathrm{T}(M)) = \hat{\mathrm{T}}(M)$. The right-to-left inclusion holds by the definition of trace trees. □

Proposition 5.35. *Let M and M' be Moore machines. Then $\mathrm{T}(M) \subseteq \mathrm{T}(M')$ if and only if $\hat{\mathrm{T}}(M) \subseteq \hat{\mathrm{T}}(M')$.*

Proof. Analogous to the proof of Proposition 5.17. □

In partial trace tree languages, parallel composition corresponds to partial trace tree language intersection.

Proposition 5.36. *Let M_1 and M_2 be two compatible Moore machines. Then $\hat{\mathrm{T}}(M_1 \parallel M_2) = \hat{\mathrm{T}}(M_1) \cap \hat{\mathrm{T}}(M_2)$.*

Proof. Let $M_1 = \langle I_1, O_1, S_1, \iota_1, \delta_1, \rho_1\rangle$ and $M_2 = \langle I_2, O_2, S_2, \iota_2, \delta_2, \rho_2\rangle$ and note that $O_1 \cap O_2 = \emptyset$. Let $M_1 \| M_2 = \langle I, O, S, \iota, \delta, \rho\rangle$. Let $t \in \hat{\mathrm{T}}$ and let $r \in S^{\blacktriangle}$, $r^1 \in S_1^{\blacktriangle}$ and $r^2 \in S_2^{\blacktriangle}$ with $dom(t) = dom(r) = dom(r^1) = dom(r^2)$ such that $r_w = \langle r^1_w, r^2_w\rangle$ for all $w \in dom(r)$. We prove the claim $\hat{\mathrm{T}}(M_1 \parallel M_2) = \hat{\mathrm{T}}(M_1) \cap \hat{\mathrm{T}}(M_2)$ by showing that r is run tree of t in $M_1 \parallel M_2$ if and only if r^1 and r^2 are run trees of t in M_1 and M_2, respectively. If $dom(t) = \emptyset$ then this equivalence is trivial, so assume $dom(t) \neq \emptyset$. We have to show the three conjuncts of the second case of the definition of run trees.

- $\iota = \iota_1 \times \iota_2$ by definition of $M_1 \parallel M_2$, so $\langle r^1_\epsilon, r^2_\epsilon\rangle \in \iota$ iff $r^1_\epsilon \in \iota_1$ and $r^2_\epsilon \in \iota_2$.
- Fix arbitrary $\langle w, n\rangle \in \mathbb{N}^* \times \mathbb{N}$ with $wn \in dom(r)$. By definition of $M_1 \parallel M_2$, $\delta(\langle r^1_w, r^2_w\rangle, t_w|_I) = \delta_1\big(r^1_w, \big(t_w|_I \triangledown \rho_2(r^2_w)\big)|_{I_1}\big) \times \delta_2\big(r^2_w, \big(t_w|_I \triangledown \rho_1(r^1_w)\big)|_{I_2}\big)$. We can simplify the right-hand side of this equation since

$$\begin{aligned}\delta_1\big(r^1_w, \big(t_w|_I \triangledown \rho_2(r^2_w)\big)|_{I_1}\big) &= \delta_1\big(r^1_w, t_w|_{I_1 \cap I} \triangledown \rho_2(r^2_w)|_{I_1}\big)\\ &= \delta_1\big(r^1_w, t_w|_{I_1 \setminus O_2} \triangledown \rho_2(r^2_w)|_{I_1}\big)\\ &= \delta_1(r^1_w, t_w|_{I_1 \setminus O_2} \triangledown t_w|_{I_1 \cap O_2})\\ &= \delta_1(r^1_w, t_w|_{I_1}),\end{aligned}$$

  where the third equality holds because $t_w|_{O_2} = \rho_2(r^2_w)$, and the second equality holds because

$$\begin{aligned}I_1 \cap I &= I_1 \cap \big((I_1 \cup I_2) \setminus O\big) = \big((I_1 \cap I_1) \setminus O\big) \cup \big((I_1 \cap I_2) \setminus O\big)\\ &= I_1 \setminus O = I_1 \setminus (O_1 \cup O_2) = I_1 \setminus O_2\end{aligned}$$

With similar arguments, $\delta_2\big(r^2_w, \big(t_w|_I \triangledown \rho_1(r^1_w)\big)|_{I_2}\big) = \delta_2(r^2_w, t_w|_{I_2})$. Hence $\langle r^1_{wn}, r^2_{wn}\rangle \in \delta(\langle r^1_w, r^2_w\rangle, t_w|_I)$ iff $r^1_{wn} \in \delta_1(r^1_w, t_w|_{I_1})$ and $r^2_{wn} \in \delta_2(r^2_w, t_w|_{I_2})$.

- Fix an arbitrary $w \in dom(r)$. We have $\rho(\langle r^1_w, r^2_w\rangle) = \rho_1(r^1_w) \triangledown \rho_2(r^2_w)$ by definition of $M_1 \parallel M_2$. Therefore

$$\begin{array}{lll} t_w|_O \trianglelefteq \rho(\langle r^1_w, r^2_w\rangle) & \text{iff} & t_w|_{O_1} \triangledown t_w|_{O_2} \trianglelefteq \rho_1(r^1_w) \triangledown \rho_2(r^2_w) \\ & \text{iff} & t_w|_{O_1} \trianglelefteq \rho_1(r^1_w) \text{ and } t_w|_{O_2} \trianglelefteq \rho_2(r^2_w), \end{array}$$

  where the second equivalence holds because $O_1 \cap O_2 = \emptyset$. □

The partial trace tree languages of compatible Moore machines do not block each other.

Proposition 5.37. *Let M_1 and M_2 be Moore machines. If M_1 and M_2 are compatible then $\langle \hat{\mathrm{T}}, \hat{\mathrm{T}}(M_1), \hat{\mathrm{T}}(M_2)\rangle \in \mathrm{NB}$, i. e., $\hat{\mathrm{T}}(M_1)$ and $\hat{\mathrm{T}}(M_2)$ do not block each other.*

Proof. Let $M_1 = \langle I_1, O_1, S_1, \iota_1, \delta_1, \rho_1\rangle$ and $M_2 = \langle I_2, O_2, S_2, \iota_2, \delta_2, \rho_2\rangle$ and assume that M_1 and M_2 are compatible, i. e., $O_1 \cap O_2 = \emptyset$. By Proposition 4.8, we have to show that every completely join-irreducible element of $\boldsymbol{\mathcal{L}}_{\mathbf{T}}$ is contained in $\hat{\mathrm{T}}(M_1)$ or in $\hat{\mathrm{T}}(M_2)$ whenever $\hat{\mathrm{T}}(M_1) \cap \hat{\mathrm{T}}(M_2)$ is an upper bound of its strict downward-closure.

By Lemma 5.31, $\mathcal{J}(\boldsymbol{\mathcal{L}}_{\mathbf{T}})$ is the set of principal ideals in $\langle \hat{\mathrm{T}}, \trianglelefteq^\Delta\rangle$. Let $t \in \hat{\mathrm{T}}$ and assume that $\hat{\mathrm{T}}(M_1) \cap \hat{\mathrm{T}}(M_2)$ is an upper bound of $\{\trianglelefteq^\Delta(s) \mid s \in \hat{\mathrm{T}}, s \lhd^\Delta t\}$, the strict downward closure of $\trianglelefteq^\Delta(t)$ in $\langle \mathcal{J}(\boldsymbol{\mathcal{L}}_{\mathbf{T}}), \subseteq\rangle$. We have to show that $\trianglelefteq^\Delta(t) \subseteq \hat{\mathrm{T}}(M_1)$ or $\trianglelefteq^\Delta(t) \subseteq \hat{\mathrm{T}}(M_2)$. As $\hat{\mathrm{T}}(M_1)$ and $\hat{\mathrm{T}}(M_2)$ are downward-closed by Lemma 5.34, it actually suffices to show that $t \in \hat{\mathrm{T}}(M_1)$ or $t \in \hat{\mathrm{T}}(M_2)$. Case distinction.

- $t = \lambda$. Then $t \in \mathrm{T}(M_1) \subseteq \hat{\mathrm{T}}(M_1)$ by Proposition 5.26 and Lemma 5.34.
- $t \neq \lambda$. Then $\lhd^\Delta(t)$ is non-empty and, by Lemma 5.29, finite. Therefore, there exists a maximal element s in $\lhd^\Delta(t)$. Because of our assumption that $\hat{\mathrm{T}}(M_1) \cap \hat{\mathrm{T}}(M_2)$ is an upper bound of the strict downward closure of $\trianglelefteq^\Delta(t)$, we know that $\trianglelefteq^\Delta(s) \subseteq \hat{\mathrm{T}}(M_1) \cap \hat{\mathrm{T}}(M_2)$. Lemma 5.28 provides more information about s, which leads to another case distinction.
  - There is $w \in dom(t)$ such that $s = t|_{dom(t)\setminus\{w\}}$ and $t_w = \bot$.
    As $s \in \hat{\mathrm{T}}(M_1)$ there exists a run tree $r \in S_1^\blacktriangle$ of s in M_1. We will show that r can be extended to a run tree $r' \in S_1^\blacktriangle$ of t in M_1. This requires yet another case distinction.
    - $w = \epsilon$. Then $dom(r) = dom(s) = \emptyset$, and therefore $dom(t) = \{\epsilon\}$. As $\iota_1 \neq \emptyset$, we can construct $r' \in S_1^\blacktriangle$ with $dom(r') = \{\epsilon\}$ such that $r'_\epsilon \in \iota_1$. Obviously, r' is a run tree of t in M_1, so $t \in \hat{\mathrm{T}}(M_1)$.

  - $w \neq \epsilon$. Then there are $v \in \mathbb{N}^*$ and $n \in \mathbb{N}$ such that $w = vn$. As v is a non-leaf node in t, the definition of partial trees forces $t_v \in \Sigma$, so the second argument of $\delta_1(r_v, t_v|_{I_1})$ is defined. And as $\delta_1(r_v, t_v|_{I_1}) \neq \emptyset$, we are able to construct $r' \in S_1^{\blacktriangle}$ with $dom(r') = dom(r) \cup \{vn\} = dom(s) \cup \{w\} = dom(t)$ and $r'|_{dom(r)} = r$ such that $r'_{vn} \in \delta_1(r'_v, t_v|_{I_1})$. Obviously, r' is a run tree of t in M_1, so $t \in \hat{\mathrm{T}}(M_1)$.

– $dom(s) = dom(t)$ and there are $w \in dom(t)$ and $x \in dom(t_w)$ such that $s_w = t_w|_{dom(t_w)\setminus\{x\}}$ and $s_v = t_v$ for all $v \in dom(t) \setminus \{w\}$.
As O_1 and O_2 are disjoint, there is $i \in \{1, 2\}$ such that $x \notin O_i$. Because $s \in \hat{\mathrm{T}}(M_i)$, there is a run tree $r \in S_i^{\blacktriangle}$ of s in M_i. And as $s_w = t_w|_{dom(t_w)\setminus\{x\}}$ and $x \notin O_i$, we have $t_w|_{O_i} = s_w|_{O_i} \trianglelefteq \rho_i(r_w)$. Therefore, r is also a run tree of t in M_i, so $t \in \hat{\mathrm{T}}(M_i)$. □

Now, we derive a circular A-G rule for partial trace tree languages by instantiating the A-G rule $\mathrm{AG}_2^{\mathrm{c}}$ from Section 4.6 to the lattice $\boldsymbol{\mathcal{L}}_{\mathbf{T}}$. To this end, we introduce the binary relation MC_{T}, which will function as the side condition of the derived rule.

Definition 5.38 (Moore-compatibility).
We say that two elements $S_1, S_2 \in \mathcal{L}_{\mathrm{T}}$ are *Moore-compatible* (in $\boldsymbol{\mathcal{L}}_{\mathbf{T}}$) if and only if there are compatible Moore machines M_1 and M_2 such that $S_1 = \hat{\mathrm{T}}(M_1)$ and $S_2 = \hat{\mathrm{T}}(M_2)$. We define the binary relation MC_{T} on $\mathcal{L}_{\mathrm{T}}$ by $\langle S_1, S_2 \rangle \in \mathrm{MC}_{\mathrm{T}}$ if and only if S_1 and S_2 are Moore-compatible (in $\boldsymbol{\mathcal{L}}_{\mathbf{T}}$).

Theorem 5.39. *The A-G rule* $\mathrm{AG}_{\mathrm{T}}^{\mathrm{Moore}}$ *is weaker than* $\mathrm{AG}_2^{\mathrm{c}}$ *and sound, where*

$$\mathrm{AG}_{\mathrm{T}}^{\mathrm{Moore}} : \frac{\mathsf{P}_2 \sqcap \mathsf{S}_1 \sqsubseteq \mathsf{P}_1 \qquad \mathsf{P}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_2}{\mathsf{S}_1 \sqcap \mathsf{S}_2 \sqsubseteq \mathsf{P}_1 \sqcap \mathsf{P}_2} \text{ if } \mathrm{MC}_{\mathrm{T}}[\mathsf{P}_1, \mathsf{P}_2].$$

Proof. Analogous to the proof of Theorem 5.21. The proof rests on the fact that $\mathrm{MC}_{\mathrm{T}}[\mathsf{P}_1, \mathsf{P}_2] \models^{\boldsymbol{\mathcal{L}}_{\mathbf{T}}} \mathrm{NB}[\top, \mathsf{P}_1, \mathsf{P}_2]$ (Proposition 5.37), so $\mathrm{AG}_{\mathrm{T}}^{\mathrm{Moore}}$ is weaker than $\mathrm{AG}_2^{\mathrm{c}}$, which is sound because $\boldsymbol{\mathcal{L}}_{\mathbf{T}}$ is well-approximable (Proposition 5.32). □

As in Section 5.1.2, we can use the A-G rule $\mathrm{AG}_{\mathrm{T}}^{\mathrm{Moore}}$ for partial trace tree languages to derive a more useful rule (5.2) for trace tree languages. Note that the well-definedness conditions for the trace tree language rule (5.2) are exactly the same than for the trace language rule (5.1). Therefore, they can be relaxed in the same way if the specification machines are abstractions of the system machines.

Corollary 5.40. *Let* M_1, M_2, M_1', M_2' *be four Moore machines such that* M_2' *and* M_1, M_1' *and* M_2, M_1 *and* M_2 *as well as* M_1' *and* M_2' *all are compatible. Then* (5.2) *is well-defined and sound.*

$$\frac{\mathrm{T}(M_2' \parallel M_1) \subseteq \mathrm{T}(M_1') \qquad \mathrm{T}(M_1' \parallel M_2) \subseteq \mathrm{T}(M_2')}{\mathrm{T}(M_1 \parallel M_2) \subseteq \mathrm{T}(M_1' \parallel M_2')} \tag{5.2}$$

Proof. Analogous to the proof of Corollary 5.22. □

Corollary 5.41. *Let M_1, M_2, M_1', M_2' be four Moore machines and let O_1, O_2, O_1', O_2' be their respective sets of output ports. If M_1 and M_2 are compatible and $O_1' \subseteq O_1$ and $O_2' \subseteq O_2$ then (5.2) is well-defined and sound.*

Proof. Analogous to the proof of Corollary 5.23. □

Note that given two Moore machines M and M', $\mathrm{T}(M) \subseteq \mathrm{T}(M')$ implies $\mathrm{W}(M) \subseteq \mathrm{W}(M')$ because every trace can be viewed as a (degenerate) trace tree. However, the reverse implication is not true in general, as the following example shows. As a consequence, the rules (5.1) and (5.2) are not equivalent, despite their similarity.

Example 5.42. Let $x \in \mathcal{X}$ be a port and assume that the three distinct values 0, 1, 2 are contained in the domain of values $\mathcal{D}$. Let $M = \langle \emptyset, \{x\}, \{\circ, \circledcirc, *\}, \{\circledcirc\}, \delta, \rho \rangle$ and $M' = \langle \emptyset, \{x\}, \{\circ, \circledcirc, *, \circledast\}, \{\circledcirc, \circledast\}, \delta', \rho' \rangle$ be Moore machines, where the transition and output functions are defined by the following equations:

$$\begin{array}{llll}
\delta(\circledcirc, \bot) = \{\circ, *\} & \rho(\circledcirc)(x) = 0 & \delta'(\circledcirc, \bot) = \{\circ\} & \rho'(\circledcirc)(x) = 0 \\
 & & \delta'(\circledast, \bot) = \{*\} & \rho'(\circledast)(x) = 0 \\
\delta(\circ, \bot) = \{\circ\} & \rho(\circ)(x) = 1 & \delta'(\circ, \bot) = \{\circ\} & \rho'(\circ)(x) = 1 \\
\delta(*, \bot) = \{*\} & \rho(*)(x) = 2 & \delta'(*, \bot) = \{*\} & \rho'(*)(x) = 2
\end{array}$$

It is easy to see that $\mathrm{W}(M) = \mathrm{W}(M')$, as for all words $w \in \mathrm{W}$, $w \in \mathrm{W}(M)$ resp. $w \in \mathrm{W}(M')$ iff $w = \epsilon$ or $w_0(x) = 0$ and $w_1(x) = \cdots = w_{n-1}(x) \in \{1, 2\}$, where $n = \mathit{len}(w)$. However, $\mathrm{T}(M) \not\subseteq \mathrm{T}(M')$, as the tree $t \in \mathrm{T}$ with $\mathit{dom}(t) = \{\epsilon, 0, 1\}$, $t_\epsilon(x) = 0$, $t_0(x) = 1$ and $t_1(x) = 2$ is a trace tree of M but not of M'.

5.1.4 Comparison to Other Work

A number of circular assume-guarantee rules for Moore machines and similar transition system based formalisms have been developed in the past. Many of these rules can be proven to be instances of our assume-guarantee framework using similar techniques than for the rules (5.1) and (5.2). We support this claim in more detail now.

In [HQRT02, Theorem 3], Henzinger et al. prove soundness of a circular rule for Moore machines with simulation [Mil71] as refinement. They require their machines to be finitely non-deterministic, which implies that simulation is equivalent to trace tree containment, cf. Rajamani's thesis [Raj99]. Thus, their rule is slightly less general than our rule (5.2) — actually the two rules are the same except that (5.2) does not require finite non-determinism.

Moore machines can be extended in various ways. For instance, we can admit the value of an output port to depend on the current state *and* on the current input values, which results in Mealy machines [McM97]. This new definition

of the output function induces a dependency relation on the ports of a Mealy machine, more precisely, an output port y depends on an input port x if there exists a state in which the value of the output function for y actually depends on the current value of x. Compatibility is redefined to take into account this dependency relation — two Mealy machines are compatible if their sets of output ports are disjoint and the union of their dependency relations is acyclic. The parallel composition of two compatible Mealy machines is again a Mealy machine which is, like in the case of Moore machines, defined by cross-wiring inputs and outputs. Alur and Henzinger [AH99, Proposition 5] as well as McMillan [McM97, Theorem 1] both develop circular assume-guarantee rules for Mealy machines (termed 'reactive modules' in [AH99]), with trace containment as refinement. The structure of the rule by Alur and Henzinger is the same than that of our rule (5.1), and in fact, (5.1) can be generalized to Mealy machines. The key observation is that one can define a partial trace language for Mealy machines in much the same way than for Moore machines, only one has to take into account the dependency relation; technically speaking, the last letter of a partial trace must be a partial assignment whose domain is downward-closed w. r. t. the order induced by the dependency relation. With this definition, partial trace languages for Mealy machines inherit all the properties from those languages for Moore machine, in particular, the partial trace languages of two compatible Mealy machines do not block each other. Thus, we can instantiate the A-G rule $\mathrm{AG}_2^{\mathrm{c}}$ in the same way than we have done for Moore machines in Theorem 5.21. Analogously to Corollary 5.22, we can then derive a rule for trace containment, similar to (5.1).

Another extension of Moore machines are timed systems [TAKB96]. In a timed system, each state is a pair consisting of a location, i. e., a value from a (usually finite) discrete set of locations, and a vector of real-valued clocks. Transitions can only change the location part of the state and possibly reset some clocks, the other clocks increase continuously at a fixed speed. Also, the value of an output port may only depend on the location part of the state, not on the clocks. Because of the latter requirement, the computations of such a system may be described as timed traces, i. e., as finite sequences of pairs $\langle \delta, a \rangle$ (called timed events) where a is an assignment of values to the input and output ports and δ is a positive increment of the clocks. The idea of a timed trace $\langle \delta_0, a_0 \rangle, \langle \delta_1, a_1 \rangle, \ldots$ is that at first a_0 is observable on the input and output ports for δ_0 time units, then a_1 is observable for δ_1 time units, and so on. Compatibility and parallel composition of timed systems are defined analogously to the respective notions for Moore machines. Tasiran et al. [TAKB96, Proposition 8] present a circular assume-guarantee rule for timed systems, with trace containment as refinement. The structure of their rule is similar to that of our rule (5.1), and as the structure of timed traces is similar to the structure of traces of Moore machines, it seems plausible that (5.1) can be generalized to timed systems with trace containment by instantiating the assume-guarantee framework to partial trace languages of timed systems along the lines of Section 5.1.2.

Kripke structures are a well-known representation of transition systems. They differ from Moore machines in giving up the distinction between input and output ports (in the terminology of Moore machines, a Kripke structure has only output ports) and in the definition of parallel composition, which is a total operation on Kripke structures. Two Kripke structures running in parallel synchronize and communicate via their shared ports, so Kripke structures are appropriate for modeling shared-memory systems. Note that the value of a port must be well-defined at every moment in time. Thus, in a situation where the two structures cannot agree on a common value for some port, deadlock arises. In [Mai01], we prove soundness of a circular assume-guarantee rule for Kripke structures with trace containment as refinement, similar to the rule (5.1). The rule is restricted to compatible Kripke structures, where compatibility is a condition on a pair of structures that ensures their product to be free of deadlocks. Note that, unlike in the case of Moore and Mealy machines, compatibility of Kripke structures is a semantic condition, i. e., checking compatibility involves inspection of the reachable state space.

Recently, Rajamani and Rehof [RR01] have designed a trace language for CCS processes [Mil89] and proven soundness of a circular assume-guarantee rule for such processes with trace containment as refinement. Their rule bears some structural similarities to our rule (5.1), however, we do not see a way how to generalize their trace language to a partial trace language. Therefore, we cannot generalize (5.1) to CCS processes. Yet, the third premise of the rule in [RR01, Theorem 1] puts non-blockingness constraints on the communication channels of parallel processes. Roughly, a channel x is non-blocking for a process P which interacts with a process Q if, after any sequence of joint transitions of both processes, a send (resp. receive) action by P on x can be matched by Q with the corresponding receive (resp. send) action on x. This vaguely resembles non-blockingness as we have defined in Section 4.2 (see also Example 4.9), more precisely, it resembles an asymmetric version of non-blockingness where, after any sequence of joint transitions, one distinguished process (P in this case) is not blocked from communicating with the other process via x. Whether this view admits instantiating our assume-guarantee framework to CCS processes needs to be investigated still.

It should be said that there is work on circular assume-guarantee reasoning for proving refinement, which probably cannot be covered by our assume-guarantee framework because of the use of structure beyond what is expressible in our semilattice based setting. For instance, Alur and Grosu [AG00] combine the reactive modules of [AH99] with formalisms for structured, hierarchical specification of sequential behavior, inspired by Statecharts [Har87] and also present in object-oriented modeling languages such as UML [BRJ98]. In a similar fashion, Henzinger et al. [HMP01] present a language which allows arbitrary nesting of parallel and sequential compositions of hybrid systems. Both papers introduce notions of refinement and prove soundness of corresponding assume-guarantee

rules. Because of the use of sequential composition, which is a non-commutative, non-idempotent operation, we doubt that it is possible to reformulate these approaches as A-G rules with side conditions in the style of Chapter 3, where premises and conclusion are restricted to formulas involving only the order and the (commutative and idempotent) meet of a meet-semilattice.

5.2 Rules for Assume-Guarantee Specifications

Systems are usually designed to operate in specific environments, and they may badly malfunction when exposed to an unsuited environment. Therefore, the specification of a system should not only state what the system does but also what the world around it must (or must not) do. According to this view, every specification falls naturally into two parts, a specification how the system assumes its environment to act and a specification what the system guarantees provided the environment acts as assumed. Such a two-part specification is commonly termed an 'assume-guarantee specification'.

In the realm of discrete concurrent systems, based on the distinction between system and environment, it is natural to assume interleaving execution, i. e., the environment and the system take turns in executing atomic steps of their respective behavior. In this setting, for a system to satisfy an assume-guarantee specification, the system's steps must satisfy the guarantee whenever the preceding steps of the environment did satisfy the assumption. I. e., along every sequence of atomic steps, the guarantee must hold at least one step longer than the assumption.

When verifying a complex system, say a parallel composition of two subsystems, we may compose the assume-guarantee specifications of the two subsystems in such a way that the assumptions are mutually discharged, so the complex system satisfies the conjunction of both guarantees. Despite the apparent circularity, a number of papers, for example [AP93, JT96], have shown this composition principle to be sound. All soundness proofs require induction and make use of the special property of assume-guarantee specifications, that the guarantee must hold at least one step longer than the assumption.

In this section, we are going to establish soundness of this composition principle by showing it to be an instance the assume-guarantee framework from Chapter 4. To this end we introduce a (fragment of a) linear-time temporal logic whose models are both finite and infinite sequences of observations. We develop the logical operators that are needed for circular assume-guarantee reasoning, namely conjunction, implication and the formation of assume-guarantee specifications. The properties expressible in this logic form our complete and well-approximable lattice, for which we then derive A-G rules from the family of rules $\{\mathrm{AG}'_n \mid n \in \mathbb{N}\}$, see Section 4.5.

5.2.1 (Linear-Time) Behaviors And Properties

Definition 5.43 (Observation, Behavior).
We fix an alphabet Σ whose cardinality is at least 2. The elements of Σ are called *observations*. A *behavior* is a finite or infinite sequence of observations, i. e., a behavior is a word over Σ, and the set of all behaviors is denoted by Σ^∞. We say that a behavior σ is *finite* resp. *infinite* if and only if $\sigma \in \Sigma^*$ resp. $\sigma \in \Sigma^\omega$.

Recall from Chapter 2 that $\preceq$, the prefix relation, is an order on Σ^∞. When we view a behavior σ as a sequence of observations observed at discrete successive events in time then the prefix order $\preceq$ characterizes the past in the following sense. A behavior τ with $\tau \preceq \sigma$ corresponds to how the behavior σ did look some time in the past before it has gradually evolved into the σ of now.

We will view temporal properties as the sets of behaviors that satisfy them. To be consistent with the metaphor of behaviors being evolving sequences of observations, we demand that when a behavior σ satisfies a property P then all its past behaviors also satisfy P. Thus, a property must be prefix-closed, or in other words: an order ideal in $\langle \Sigma^\infty, \preceq \rangle$.

Definition 5.44 (Property, Truth, Entailment).
A *property* is a non-empty order ideal in $\langle \Sigma^\infty, \preceq \rangle$. We say that a property P holds *true* of a behavior σ (or σ *satisfies* P) if and only if $\sigma \in P$. We say that a property P *entails* a property Q if and only if $P \subseteq Q$.

For technical reasons we require properties to be non-empty. This is not a loss of expressiveness because with respect to sequences of observations, $\emptyset$ and $\{\epsilon\}$ are indistinguishable since the empty behavior ϵ corresponds to a nil sequence of observations, i. e., to no observations.

Definition 5.45 (Ordered Set of Properties).
We denote the set of all properties by $\mathcal{L}_{\mathrm{AG}}$, i. e., $\mathcal{L}_{\mathrm{AG}} = \mathcal{O}(\langle \Sigma^\infty, \preceq \rangle) \setminus \{\emptyset\}$. By $\leq$, we denote the order on $\mathcal{L}_{\mathrm{AG}}$ which is induced by the order $\subseteq$ on $\mathcal{O}(\langle \Sigma^\infty, \preceq \rangle)$. We write $\boldsymbol{\mathcal{L}_{\mathbf{AG}}}$ to abbreviate the ordered set $\langle \mathcal{L}_{\mathrm{AG}}, \leq \rangle$.

Note that for all $P, Q \in \mathcal{L}_{\mathrm{AG}}$, $P \wedge Q$ and $P \vee Q$ exist and $P \wedge Q = P \cap Q$ and $P \vee Q = P \cup Q$. Thus, $\boldsymbol{\mathcal{L}_{\mathbf{AG}}}$ inherits the distributive lattice structure from $\langle \mathcal{O}(\langle \Sigma^\infty, \preceq \rangle), \subseteq \rangle$. Actually, for every subset $S \subseteq \mathcal{L}_{\mathrm{AG}}$, $\bigwedge S$ exists and $\bigwedge S = \bigcap S$, and furthermore, $\bigvee \emptyset = \{\epsilon\}$ and for every non-empty subset $S \subseteq \mathcal{L}_{\mathrm{AG}}$, $\bigvee S$ exists and $\bigvee S = \bigcup S$. Hence, $\boldsymbol{\mathcal{L}_{\mathbf{AG}}}$ is complete.

Proposition 5.46. $\boldsymbol{\mathcal{L}_{\mathbf{AG}}}$ *is a complete distributive lattice.*

We already know that the entailment relation on properties is the order of $\boldsymbol{\mathcal{L}_{\mathbf{AG}}}$. With meet and join, the lattice structure provides two logical operators on properties, namely *conjunction* and *disjunction*. For our purposes conjunction is much more relevant than disjunction, so we will state only relatively few propositions involving disjunction. Note that $\{\epsilon\}$ and Σ^∞, the least resp. greatest element in $\boldsymbol{\mathcal{L}_{\mathbf{AG}}}$, play the roles of the logical constants *false* and *true*, respectively. Note also that the set of finite behaviors Σ^* is a property whereas the set of infinite behaviors Σ^ω is not.

In order to instantiate the assume-guarantee rules from Chapter 4, we must show $\boldsymbol{\mathcal{L}_{\mathbf{AG}}}$ to be well-approximable. To this end, we investigate the completely join-irreducible elements and show that they generate a well-founded approximation.

Lemma 5.47. $\mathcal{J}(\boldsymbol{\mathcal{L}}_{\mathbf{AG}}) = \big\{\preceq(\sigma) \mid \sigma \in \Sigma^\infty \setminus \{\epsilon\}\big\}$.

Proof. To prove the left-to-right inclusion, let $P \in \mathcal{J}(\boldsymbol{\mathcal{L}}_{\mathbf{AG}})$. Then $P \neq \{\epsilon\}$ as $\{\epsilon\} = \bigvee \emptyset$ is not completely join-irreducible. As $P = \bigvee\big\{\preceq(\sigma) \mid \sigma \in P \setminus \{\epsilon\}\big\}$, by complete join-irreducibility there is $\sigma \in P$ such that $P = \preceq(\sigma)$. And as $P \neq \{\epsilon\}$, we have $\sigma \in \Sigma^\infty \setminus \{\epsilon\}$. To prove the other inclusion, let $\sigma \in \Sigma^\infty \setminus \{\epsilon\}$ and let $S \subseteq \mathcal{L}_{\mathrm{AG}}$ with $\preceq(\sigma) = \bigvee S$. Then $S \neq \emptyset$ and $\preceq(\sigma) = \bigcup S$, so there is $P \in S$ such that $\sigma \in P$, so $\preceq(\sigma) \subseteq P$. And as $P \subseteq \bigcup S = \preceq(\sigma)$, we get $\preceq(\sigma) = P$. Hence, $\preceq(\sigma)$ is completely join-irreducible. □

Proposition 5.48. $\boldsymbol{\mathcal{L}}_{\mathbf{AG}}$ *is well-approximable.*

Proof. Let $\boldsymbol{\mathcal{J}}$ be the suborder of $\boldsymbol{\mathcal{L}}_{\mathbf{AG}}$ which is generated by the completely join-irreducible elements, i. e., $\mathcal{J} = \mathcal{J}(\boldsymbol{\mathcal{L}}_{\mathbf{AG}}) = \big\{\preceq(\sigma) \mid \sigma \in \Sigma^\infty \setminus \{\epsilon\}\big\}$ by Lemma 5.47. We are going to show that $\boldsymbol{\mathcal{J}}$ is a well-founded approximation of $\boldsymbol{\mathcal{L}}_{\mathbf{AG}}$. Obviously $P = \bigvee\big\{\preceq(\sigma) \mid \sigma \in P \setminus \{\epsilon\}\big\}$ for all $P \in \mathcal{L}_{\mathrm{AG}}$, so $\mathcal{J}$ is join-dense in $\boldsymbol{\mathcal{L}}_{\mathbf{AG}}$. It remains to show that $\boldsymbol{\mathcal{J}}$ satisfies DCC, so let $\sigma_0, \sigma_1, \sigma_2, \cdots \in \Sigma^\infty \setminus \{\epsilon\}$ such that $\preceq(\sigma_0) \geq \preceq(\sigma_1) \geq \preceq(\sigma_2) \geq \ldots$ is an infinite descending sequence. Either we have $\sigma_0 = \sigma_j$ for all $j > 0$, i. e., the sequence stabilizes immediately, or there is $j > 0$ such that $\sigma_0 \neq \sigma_j$. The latter implies that $\sigma_j \in \Sigma^*$, so the sequence must stabilize because the set $\{P \in \mathcal{J} \mid P \leq \preceq(\sigma_j)\} \subseteq \{\preceq(\tau) \mid \tau \in \preceq(\sigma_j)\}$ is finite. □

5.2.2 Implication

So far, the distributive lattice structure of $\boldsymbol{\mathcal{L}}_{\mathbf{AG}}$ has supplied us with the logical operators conjunction and disjunction with their usual (classical) properties. In fact, $\boldsymbol{\mathcal{L}}_{\mathbf{AG}}$ possesses more structure; it is a Heyting algebra. This induces a notion of implication, although not in the classical but in the intuitionistic sense.

Proposition 5.49. $\boldsymbol{\mathcal{L}}_{\mathbf{AG}}$ *is a Heyting algebra.*

Proof. Let $P, Q \in \mathcal{L}_{\mathrm{AG}}$ be two arbitrary properties. We must show that the set of properties $S = \{R \in \mathcal{L}_{\mathrm{AG}} \mid P \wedge R \leq Q\}$ has a greatest element. We define the set $I = \{\sigma \in \Sigma^\infty \mid \preceq(\sigma) \wedge P \leq Q\}$. Evidently, I is a non-empty order ideal in $\langle \Sigma^\infty, \preceq \rangle$, i. e., $I \in \mathcal{L}_{\mathrm{AG}}$ is a property. It is also easy to see that $P \wedge I \leq Q$, so $I \in S$. To show that I is the greatest element of S, choose any $R \in S$. We have to show that $R \leq I$, so let $\sigma \in \Sigma^\infty$ with $\sigma \in R$. Then $\preceq(\sigma) \leq R$, so $\preceq(\sigma) \wedge P \leq P \wedge R \leq Q$, hence $\sigma \in I$. □

Definition 5.50 (Implication).
Given two properties P and Q, we call the relative pseudo-complement $P \Rightarrow Q$ an *implication*.

Notation. As operators on properties, we assume that conjunction and disjunction bind tighter than implication, e. g., given the properties P, Q, R and S, we may omit parentheses in the expression $(P \wedge Q) \Rightarrow (R \vee S)$.

The proof of Proposition 5.49 does not only reveal that implication is a well-defined binary operation on properties, it also provides a characterization of implication in terms of prefixes. Given two properties P and Q, a behavior σ satisfies the implication $P \Rightarrow Q$ iff all prefixes of σ which satisfy P do also satisfy Q. This is formalized by the proposition below.

Proposition 5.51. *Let $P, Q \in \mathcal{L}_{\mathrm{AG}}$. Clearly, $P \Rightarrow Q = \{\sigma \in \Sigma^\infty \mid \preceq(\sigma) \wedge P \leq Q\}$.*

The following proposition states a number of useful facts about implication, in particular about monotonicity and the interaction with conjunction. Note that the statements 8 and 9 express the well-known reasoning principles *Modus Ponens* and *Cut*, which we will use later on in proofs.

Proposition 5.52. *Let $P, Q, R \in \mathcal{L}_{\mathrm{AG}}$. The following statements are true.*

1. *$P \leq Q \Rightarrow R$ if and only if $P \wedge Q \leq R$.*
2. *If $Q \leq R$ then $P \Rightarrow Q \leq P \Rightarrow R$.*
3. *If $P \leq Q$ then $Q \Rightarrow R \leq P \Rightarrow R$.*
4. *$P \Rightarrow P = \Sigma^\infty$.*
5. *$\Sigma^\infty \Rightarrow P = P$.*
6. *$P \Rightarrow Q \wedge R = (P \Rightarrow Q) \wedge (P \Rightarrow R)$.*
7. *$P \wedge Q \Rightarrow R = P \Rightarrow (Q \Rightarrow R)$.*
8. *$P \wedge (P \Rightarrow Q) \leq Q$.*
9. *$(P \Rightarrow Q) \wedge (Q \Rightarrow R) \leq P \Rightarrow R$.*

Proof. Direct consequences of the laws of Heyting algebras, see Chapter 2. □

It remains to be shown that our implication is not classical but genuinely intuitionistic. We do so by an example demonstrating that $\boldsymbol{\mathcal{L}_{\mathbf{AG}}}$ is not a Boolean algebra.

Example 5.53. Recall that in the Heyting algebra $\boldsymbol{\mathcal{L}_{\mathbf{AG}}}$, the pseudo-complement $\neg P$ of a property $P \in \mathcal{L}_{\mathrm{AG}}$ is defined by $\neg P = P \Rightarrow \{\epsilon\}$. Let $a \in \Sigma$. Note that the cardinality of Σ is at least 2, so $\neg(a^*) = (\Sigma \setminus \{a\})\Sigma^\infty \cup \{\epsilon\}$. The behavior a^ω witnesses that the law of excluded middle fails, for $a^\omega \notin a^*$ and $a^\omega \notin \neg(a^*)$. For this reason, $\boldsymbol{\mathcal{L}}$ cannot be a Boolean algebra. As a consequence, linear-time temporal logic interpreted over finite and infinite behaviors, unlike LTL, is not classical.

Due to the model theoretic connection between Heyting algebras and intuitionistic logic, see [Gol84], linear-time temporal logic over finite and infinite

behaviors is intuitionistic. In fact, the intuitionistic flavor of this logic arises directly from viewing behaviors as sequences of observations evolving over time. According to this view, we can not certify (in finite time), that the behavior a^ω does not satisfy the property a^*, which is why excluded middle fails.

5.2.3 Safety And Liveness

In our temporal logic, which admits speaking about finite and infinite behaviors at the same time, we can give a neat characterization of safety and liveness. According to [AS85], the difference between a safety and a liveness property is in its reaction to a violating infinite behavior σ. For a safety property P, already some finite prefix of σ must have violated P, whereas for a liveness property Q, no finite prefix of σ may violate Q.

Definition 5.54 (Safety Property, Liveness Property).
Given a property P, we call P a *safety property* if and only if for all $\sigma \in \Sigma^\omega$, $\preceq(\sigma) \wedge \Sigma^* \leq P$ implies $\sigma \in P$. We call P a *liveness property* if and only if for all $\sigma \in \Sigma^*$, $\sigma \in P$.

As a direct consequence of the above definition, checking entailment becomes easier if the right-hand side is known to be a safety resp. liveness property. In the light of the following proposition, we can also characterize safety properties as constraining finite behaviors only whereas liveness properties constrain infinite behaviors only.

Proposition 5.55. *Let $P, Q \in \mathcal{L}_{\mathrm{AG}}$. Clearly, the following statements are true.*

1. *If Q is a safety property then $P \leq Q$ if and only if $\Sigma^* \cap P \subseteq Q$.*
2. *If Q is a liveness property then $P \leq Q$ if and only if $\Sigma^\omega \cap P \subseteq Q$.*

The following proposition provides an algebraic characterization of safety resp. liveness properties, which we then will use to examine how safety resp. liveness properties can be constructed using conjunction and implication.

Proposition 5.56. *Let $P \in \mathcal{L}_{\mathrm{AG}}$. Obviously, the following equivalences are true.*

1. *P is a safety property if and only if $\Sigma^* \Rightarrow P \leq P$.*
2. *P is a liveness property if and only if $\Sigma^* \leq P$.*

Proposition 5.57. *Let $P, Q \in \mathcal{L}_{\mathrm{AG}}$. The following statements are true.*

1. *If P and Q are safety properties then $P \wedge Q$ is a safety property.*
2. *If P and Q are liveness properties then $P \wedge Q$ is a liveness property.*

3. *If Q is a safety property then $P \Rightarrow Q$ is a safety property.*

4. *If Q is a liveness property then $P \Rightarrow Q$ is a liveness property.*

Proof. We show the four claims one by one.

1. Assume that P and Q are safety properties. Then we have $\Sigma^* \Rightarrow P \wedge Q = (\Sigma^* \Rightarrow P) \wedge (\Sigma^* \Rightarrow Q) \leq P \wedge Q$, so $P \wedge Q$ is a safety property.

2. Assume that P and Q are liveness properties. Then we have $\Sigma^* \leq P \wedge Q$, so $P \wedge Q$ is a liveness property.

3. Assume that Q is a safety property. Then we have $\Sigma^* \Rightarrow (P \Rightarrow Q) = \Sigma^* \wedge P \Rightarrow Q = P \wedge \Sigma^* \Rightarrow Q = P \Rightarrow (\Sigma^* \Rightarrow Q) \leq P \Rightarrow Q$, so $P \Rightarrow Q$ is a safety property.

4. Assume that Q is a liveness property. Then we have $\Sigma^* \leq Q = \Sigma^\infty \Rightarrow Q \leq P \Rightarrow Q$, so $P \Rightarrow Q$ is a liveness property. □

As an immediate consequence of the definition of safety properties, $\preceq(\sigma)$ is a safety property for all $\sigma \in \Sigma^\infty$. Note that Σ^∞ is the only property which is both a safety and a liveness property. Many properties are neither safety nor liveness properties. For instance, if $a \in \Sigma$ then $\{\epsilon\} \cup a\Sigma^*$ and $a^\infty \cup a\Sigma^*$ both are neither safety nor liveness properties. However, every property is expressible as a conjunction of a safety and a liveness property. To this end, we define the safety (liveness) closure of a property, i. e., the strongest safety (liveness) property which the property entails.

Definition 5.58 (Safety Closure, Liveness Closure).
Given a property P, we define the *safety closure* of P, denoted by $\mathit{safe}(P)$, as the property $\Sigma^* \Rightarrow P$. We define the *liveness closure* of P, denoted by $\mathit{live}(P)$, as the property $\Sigma^* \vee P$.

As the following proposition shows, $\mathit{safe}(P)$ resp. $\mathit{live}(P)$ is in fact the least safety resp. liveness property which is entailed by P. Using this knowledge, we are able to prove that every property is the conjunction of its safety and liveness closures.

Proposition 5.59. *Let $P \in \mathcal{L}_{\mathrm{AG}}$. The following statements are true.*

1. *$\mathit{safe}(P) = \bigwedge\{Q \in \mathcal{L}_{\mathrm{AG}} \mid P \leq Q$ and $\Sigma^* \Rightarrow Q \leq Q\}$ is a safety property.*

2. *$\mathit{live}(P) = \bigwedge\{Q \in \mathcal{L}_{\mathrm{AG}} \mid P \leq Q$ and $\Sigma^* \leq Q\}$ is a liveness property.*

Proof. To show 1, note that $\mathit{safe}(P)$ is a safety property which is entailed by P as $\Sigma^* \Rightarrow (\Sigma^* \Rightarrow P) = \Sigma^* \wedge \Sigma^* \Rightarrow P = \Sigma^* \Rightarrow P$ and $P = \Sigma^\infty \Rightarrow P \leq \Sigma^* \Rightarrow P$. It remains to prove that $\mathit{safe}(P)$ is the least such property, so choose any safety property Q with $P \leq Q$. Then we have $\mathit{safe}(P) = \Sigma^* \Rightarrow P \leq \Sigma^* \Rightarrow Q \leq Q$.

To show 2, note that $\mathit{live}(P)$ is a liveness property which is entailed by P as $\Sigma^* \leq \Sigma^* \vee P$ and $P \leq \Sigma^* \vee P$. It remains to prove that $\mathit{live}(P)$ is the least such property, so choose any liveness property Q with $P \leq Q$. Then we have $\mathit{live}(P) = \Sigma^* \vee P \leq \Sigma^* \vee Q = Q$. □

Proposition 5.60. *Let $P \in \mathcal{L}_{\mathrm{AG}}$. Then $P = \mathit{safe}(P) \wedge \mathit{live}(P)$.*

Proof. $\mathit{live}(P) \wedge \mathit{safe}(P) = (\Sigma^* \vee P) \wedge \mathit{safe}(P) = \big(\Sigma^* \wedge \mathit{safe}(P)\big) \vee \big(P \wedge \mathit{safe}(P)\big) = \big(\Sigma^* \wedge (\Sigma^* \Rightarrow P)\big) \vee P = P$. □

To end our investigation of safety and liveness properties, we examine how the safety and liveness closures interact with conjunction and implication.

Proposition 5.61. *Let $P, Q \in \mathcal{L}_{\mathrm{AG}}$. The following equalities are true.*

1. $\mathit{safe}(P) \wedge \mathit{safe}(Q) = \mathit{safe}(P \wedge Q)$.
2. $\mathit{live}(P) \wedge \mathit{live}(Q) = \mathit{live}(P \wedge Q)$.
3. $\mathit{safe}(P) \Rightarrow \mathit{safe}(Q) = P \Rightarrow \mathit{safe}(Q) = \mathit{safe}(P \Rightarrow Q)$.
4. $\mathit{live}(P) \Rightarrow \mathit{live}(Q) = P \Rightarrow \mathit{live}(Q)$.

Proof. We show the four claims one by one.

1. $\mathit{safe}(P) \wedge \mathit{safe}(Q) = (\Sigma^* \Rightarrow P) \wedge (\Sigma^* \Rightarrow Q) = \Sigma^* \Rightarrow P \wedge Q = \mathit{safe}(P \wedge Q)$.
2. $\mathit{live}(P) \wedge \mathit{live}(Q) = (\Sigma^* \vee P) \wedge (\Sigma^* \vee Q) = \Sigma^* \vee (P \wedge Q) = \mathit{live}(P \wedge Q)$.
3. For proving the first equality, note that by monotonicity, $P \leq \mathit{safe}(P)$ implies $\mathit{safe}(P) \Rightarrow \mathit{safe}(Q) \leq P \Rightarrow \mathit{safe}(Q)$ To show the reverse entailment, let $\sigma \in \Sigma^\infty$ with $\sigma \in P \Rightarrow \mathit{safe}(Q)$. By Proposition 5.51, we have to show $\preceq(\sigma) \wedge \mathit{safe}(P) \leq \mathit{safe}(Q)$, so choose any $\tau \in \Sigma^\infty$ with $\tau \in \preceq(\sigma) \wedge \mathit{safe}(P)$. If $\tau \in \Sigma^*$ then $\tau \in \Sigma^* \wedge (\Sigma^* \Rightarrow P)$, so $\tau \in P$ by Modus Ponens. Thus $\tau \in P \wedge (P \Rightarrow \mathit{safe}(Q))$, so $\tau \in \mathit{safe}(Q)$ by Modus Ponens again. If $\tau \notin \Sigma^*$ then $\tau \in \Sigma^\omega$ and $\preceq(\tau) \wedge \Sigma^* \leq \mathit{safe}(P)$, so $\preceq(\tau) \wedge \Sigma^* = \preceq(\tau) \wedge \Sigma^* \wedge (\Sigma^* \Rightarrow P) \leq \preceq(\tau) \wedge P$ by Modus Ponens. As $\tau \in P \Rightarrow \mathit{safe}(Q)$, we get $\preceq(\tau) \wedge \Sigma^* \leq \preceq(\tau) \wedge P \leq \mathit{safe}(Q)$ by Proposition 5.51, which implies $\tau \in \mathit{safe}(Q)$ since $\tau \in \Sigma^\omega$ and $\mathit{safe}(Q)$ is a safety property. In both cases, we get $\tau \in \mathit{safe}(Q)$, so we have proven $\preceq(\sigma) \wedge \mathit{safe}(P) \leq \mathit{safe}(Q)$.

   The second equality holds since $P \Rightarrow \mathit{safe}(Q) = P \Rightarrow (\Sigma^* \Rightarrow Q) = P \wedge \Sigma^* \Rightarrow Q = \Sigma^* \wedge P \Rightarrow Q = \Sigma^* \Rightarrow (P \Rightarrow Q) = \mathit{safe}(P \Rightarrow Q)$.
4. By monotonicity, $P \leq \mathit{live}(P)$ implies $\mathit{live}(P) \Rightarrow \mathit{live}(Q) \leq P \Rightarrow \mathit{live}(Q)$. To show the reverse entailment, let $\sigma \in \Sigma^\infty$ with $\sigma \in P \Rightarrow \mathit{live}(Q)$, i. e., $\preceq(\sigma) \wedge P \leq \mathit{live}(Q)$ by Proposition 5.51. Then we have $\preceq(\sigma) \wedge \mathit{live}(P) = \preceq(\sigma) \wedge (\Sigma^* \vee P) = (\preceq(\sigma) \wedge \Sigma^*) \vee (\preceq(\sigma) \wedge P) \leq \Sigma^* \vee \mathit{live}(Q) = \mathit{live}(Q)$, i. e., $\sigma \in \mathit{live}(P) \Rightarrow \mathit{live}(Q)$ by Proposition 5.51. □

5.2.4 Assume-Guarantee Specifications

So far, our temporal logic has not made use of any temporal operators. Now, we are going to introduce the only temporal operator that is required for circular assume-guarantee reasoning, namely an operator that combines two properties into an assume-guarantee specification.

Informally, an assume-guarantee specification with assumption P and guarantee Q holds true of all behaviors that satisfy the guaranteed property Q at least one step longer than the assumed property P. This is formalized by the following definition.

Definition 5.62 (Assume-Guarantee Specifications).
Given two properties P and Q, we define the *assume-guarantee specification* (or just *A-G spec*) consisting of *assumption* P and *guarantee* Q, denoted by $P \xrightarrow{+} Q$, as the property $\{\sigma \in \Sigma^\infty \mid \forall \tau \in \Sigma^* \text{ with } \tau \preceq \sigma : \prec(\tau) \subseteq P \text{ implies } \tau \in Q\}$. By $P \overset{+}{\Rightarrow} Q$, we denote the *strong* A-G spec which is defined as the property $(P \xrightarrow{+} Q) \wedge (P \Rightarrow Q)$.

It is straightforward to prove that the set $P \xrightarrow{+} Q$ is a non-empty order ideal in $\langle \Sigma^\infty, \preceq \rangle$, i. e., $P \xrightarrow{+} Q$ and $P \overset{+}{\Rightarrow} Q$ are in fact properties. Note that the set $\prec(\tau)$ occuring in the above definition need not be a property because $\prec(\epsilon) = \emptyset$. For this reason, we had to use the inclusion $\prec(\tau) \subseteq P$ instead of entailment in the above definition.

Notation. We assume that conjunction and disjunction bind tighter than the operator $\xrightarrow{+}$, i. e., given the properties P, Q, R and S, we may omit parentheses in the expression $(P \wedge Q) \xrightarrow{+} (R \vee S)$. The same applies to the operator $\overset{+}{\Rightarrow}$.

The operator $\xrightarrow{+}$ satisfies similar monotonicity properties than implication, i. e., it is also monotone in the second argument and anti-tone in the first.

Proposition 5.63. *Let $P, Q, R \in \mathcal{L}_{\mathrm{AG}}$. Obviously, the statements below are true.*

1. *If $Q \leq R$ then $P \xrightarrow{+} Q \leq P \xrightarrow{+} R$.*
2. *If $P \leq Q$ then $Q \xrightarrow{+} R \leq P \xrightarrow{+} R$.*
3. *$P \xrightarrow{+} \Sigma^* = \Sigma^\infty$.*

The operator $\xrightarrow{+}$ supports a variant of Modus Ponens restricted to finite behaviors.

Proposition 5.64. *Let $P, Q \in \mathcal{L}_{\mathrm{AG}}$. Then $\Sigma^* \wedge P \wedge (P \xrightarrow{+} Q) \leq Q$.*

Proof. Let $\sigma \in \Sigma^\infty$ with $\sigma \in \Sigma^* \wedge P \wedge (P \xrightarrow{+} Q)$. Then $\prec(\sigma) \subseteq \preceq(\sigma) \subseteq P$, so $\sigma \in Q$ follows by the definition of A-G specs. □

Note that $\xrightarrow{+}$ does not support full Modus Ponens reasoning. As a counter-example, take for instance $\Sigma^\infty \wedge (\Sigma^\infty \xrightarrow{+} \Sigma^*) \not\leq \Sigma^*$.

The following proposition shows that A-G specs are safety properties that essentially ignore the liveness parts of their arguments and depend on the safety closures only. Furthermore, an A-G spec entails the corresponding implication if the guarantee is a safety property.

Proposition 5.65. *Let $P, Q \in \mathcal{L}_{\mathrm{AG}}$. The following statements are true.*

1. *$P \xrightarrow{+} Q$ is a safety property.*
2. *$P \xrightarrow{+} Q = \mathit{safe}(P) \xrightarrow{+} Q = P \xrightarrow{+} \mathit{safe}(Q)$.*
3. *$P \xrightarrow{+} Q \leq P \Rightarrow \mathit{safe}(Q)$.*

Proof. We show the three claims one by one.

1. To show that $P \xrightarrow{+} Q$ is a safety property, let $\sigma \in \Sigma^\omega$ with $\preceq(\sigma) \wedge \Sigma^* \leq P \xrightarrow{+} Q$. We have to show that $\sigma \in P \xrightarrow{+} Q$, so choose any $\tau \in \Sigma^*$ with $\tau \preceq \sigma$. Then $\tau \in \preceq(\sigma) \wedge \Sigma^*$, so $\tau \in P \xrightarrow{+} Q$. By definition of A-G specs, we find that $\prec(\tau) \subseteq P$ implies $\tau \in Q$. Thus, we conclude $\sigma \in P \xrightarrow{+} Q$ by definition of A-G specs again.

2. For proving the first equality, note that by monotonicity, $P \leq \mathit{safe}(P)$ implies $\mathit{safe}(P) \xrightarrow{+} Q \leq P \xrightarrow{+} Q$. To show the reverse entailment, let $\sigma \in \Sigma^\infty$ with $\sigma \in P \xrightarrow{+} Q$. We have to show that $\sigma \in \mathit{safe}(P) \xrightarrow{+} Q$, so choose any $\tau \in \Sigma^*$ with $\tau \preceq \sigma$ and $\prec(\tau) \subseteq \mathit{safe}(P)$. Then we have $\prec(\tau) \subseteq \Sigma^* \wedge \mathit{safe}(P)$, and as $\Sigma^* \wedge \mathit{safe}(P) = \Sigma^* \wedge (\Sigma^* \Rightarrow P) \leq P$, we get $\prec(\tau) \subseteq P$. Because $\sigma \in P \xrightarrow{+} Q$, this implies $\tau \in Q$ by definition of A-G specs. Thus, we conclude $\sigma \in \mathit{safe}(P) \xrightarrow{+} Q$ by definition of A-G specs again.

   Instead of the second equality, we are going to prove $P \xrightarrow{+} Q = P \xrightarrow{+} \mathit{safe}(Q)$. Note that by monotonicity, $Q \leq \mathit{safe}(Q)$ implies $P \xrightarrow{+} Q \leq P \xrightarrow{+} \mathit{safe}(Q)$. To show the reverse entailment, let $\sigma \in \Sigma^\infty$ with $\sigma \in P \xrightarrow{+} \mathit{safe}(Q)$. We have to show that $\sigma \in P \xrightarrow{+} Q$, so choose any $\tau \in \Sigma^*$ with $\tau \preceq \sigma$ and $\prec(\tau) \subseteq P$. By definition of A-G specs, we get $\tau \in \mathit{safe}(Q)$. So $\tau \in \Sigma^* \wedge \mathit{safe}(Q)$, and as $\Sigma^* \wedge \mathit{safe}(Q) \leq Q$ by Modus Ponens, $\tau \in Q$. Thus, we conclude $\sigma \in P \xrightarrow{+} Q$ by definition of A-G specs.

3. By 2 and the restricted variant of Modus Ponens, we have $\Sigma^* \wedge P \wedge (P \xrightarrow{+} Q) = \Sigma^* \wedge P \wedge \big(P \xrightarrow{+} \mathit{safe}(Q)\big) \leq \mathit{safe}(Q)$. By Proposition 5.55, this is equivalent to $P \wedge (P \xrightarrow{+} Q) \leq \mathit{safe}(Q)$, which is equivalent to $P \xrightarrow{+} Q \leq P \Rightarrow \mathit{safe}(Q)$. □

Note that unless the guarantee is a safety property, the operator $\stackrel{+}{\Rightarrow}$ does not construct safety properties in general. As a counter-example, take for instance, $\Sigma^\infty \stackrel{+}{\Rightarrow} \Sigma^* = (\Sigma^\infty \stackrel{+}{\rightarrow} \Sigma^*) \wedge (\Sigma^\infty \Rightarrow \Sigma^*) = \Sigma^\infty \wedge \Sigma^* = \Sigma^*$, which is not a safety property.

We end this section with a proposition which connects assume-guarantee specifications to our central notion of non-blockingness. In fact, we show that under certain conditions, if a system satisfies an A-G spec then the assumption and the guarantee do not block each other relative to the system. Note that a system is nothing else than a property, so by 'a system satisfies an A-G spec' we mean that the system entails the A-G spec.

Proposition 5.66. *Let $S, P_1, P_2 \in \mathcal{L}_{\mathrm{AG}}$. If at least one of the P_i is a safety property then $S \leq P_1 \stackrel{+}{\rightarrow} P_2$ implies $\langle S, P_1, P_2\rangle \in \mathrm{NB}$, i. e., P_1 and P_2 do not block each other relative to S.*

Proof. Assume that there is $i \in \{1, 2\}$ such that P_i is a safety property. Moreover, assume that $S \leq P_1 \stackrel{+}{\rightarrow} P_2$. We have to show $\langle S, P_1, P_2\rangle \in \mathrm{NB}$ using Proposition 4.8. Recall that by Lemma 5.47, $\mathcal{J}(\boldsymbol{\mathcal{L}}_{\mathbf{AG}})$ is the set of those principal ideals in $\langle \Sigma^\infty, \preceq\rangle$ which are generated by non-empty behaviors. We choose an arbitrary $\sigma \in \Sigma^\infty \setminus \{\epsilon\}$. Assume that $\preceq(\sigma) \leq S$ and for all $\tau \in \Sigma^\infty \setminus \{\epsilon\}$, $\preceq(\tau) < \preceq(\sigma)$ implies $\preceq(\tau) \leq P_1 \wedge P_2$. We have to show that $\preceq(\sigma) \leq P_1$ or $\preceq(\sigma) \leq P_2$. Actually, we can simplify the first assumption to $\sigma \in S$ and the second one to $\tau \in P_1 \wedge P_2$ for all $\tau \in \Sigma^\infty \setminus \{\epsilon\}$ with $\tau \prec \sigma$. Furthermore, we can strengthen the second assumption to $\tau \in P_1 \wedge P_2$ for all $\tau \in \Sigma^\infty$ with $\tau \prec \sigma$, because $\epsilon \in P_1 \wedge P_2$ trivially. Now, we have to show that $\sigma \in P_1$ or $\sigma \in P_2$. We distinguish two cases.

- $\sigma \in \Sigma^* \setminus \{\epsilon\}$. Then there exists $\tau \in \Sigma^\infty$ with $\tau \prec \sigma$ such that $\prec(\sigma) = \preceq(\tau)$. By the first assumption, we get $\sigma \in P_1 \stackrel{+}{\rightarrow} P_2$. By the second assumption, we get $\preceq(\tau) \leq P_1 \wedge P_2 \leq P_1$. Thus, we have $\prec(\sigma) = \preceq(\tau) \subseteq P_1$, which by definition of A-G specs implies $\sigma \in P_2$.

- $\sigma \in \Sigma^\omega$. Then $\preceq(\sigma) \wedge \Sigma^* \leq P_1 \wedge P_2$ by the second assumption since $\tau \prec \sigma$ for all $\tau \in \Sigma^\infty$ with $\tau \in \preceq(\sigma) \wedge \Sigma^*$. Thus, we have $\preceq(\sigma) \wedge \Sigma^* \leq P_1 \wedge P_2 \leq P_i$, which implies $\sigma \in P_i$ because P_i is a safety property. □

5.2.5 Assume-Guarantee Rules

Now, we are ready to instantiate the family of A-G rules $\{\mathrm{AG}'_n \mid n \in \mathbb{N}\}$ from Section 4.5 to our lattice of properties $\boldsymbol{\mathcal{L}}_{\mathbf{AG}}$ in order to derive A-G rules for reasoning about assume-guarantee specifications. To this end, we introduce the ternary relation $\mathrm{NB}^{\mathrm{AG}}$, which will function as the side condition of the derived rules.

Definition 5.67 (Entailment of Non-Blocking A-G Specs).
Given three properties S, E and M, we say that S *entails the non-blocking A-G spec* $E \xrightarrow{+} M$ if and only if

- $S \leq E \xrightarrow{+} M$ and
- E is a safety property or M is a safety property.

We define the ternary relation $\mathrm{NB}^{\mathrm{AG}}$ on $\mathcal{L}_{\mathrm{AG}}$ by $\langle S, E, M\rangle \in \mathrm{NB}^{\mathrm{AG}}$ if and only if S entails the non-blocking A-G spec $E \xrightarrow{+} M$.

Given $n \in \mathbb{N}$, we derive soundness of the instantiated rules $\mathrm{AG}_n^{\mathrm{AG}}$ from soundness of the stronger rule AG'_n. Recall the variables $\mathsf{S}, \mathsf{A}_1, \ldots, \mathsf{A}_n, \mathsf{G}_1, \ldots, \mathsf{G}_n \in \mathcal{V}$ from Section 4.5, and remember that $\bigsqcap_i^n \mathsf{A}_i$ and $\bigsqcap_i^n \mathsf{G}_i$ are abbreviations for $\mathsf{A}_1 \sqcap \ldots \sqcap \mathsf{A}_n$ and $\mathsf{G}_1 \sqcap \ldots \sqcap \mathsf{G}_n$, respectively.

Theorem 5.68. *For all $n \in \mathbb{N}$, the A-G rule $\mathrm{AG}_n^{\mathrm{AG}}$ is weaker than AG'_n and sound, where*

$$\mathrm{AG}_n^{\mathrm{AG}} : \frac{\mathsf{A}_1 \sqcap \mathsf{S} \sqsubseteq \mathsf{G}_1 \;\ldots\; \mathsf{A}_n \sqcap \mathsf{S} \sqsubseteq \mathsf{G}_n \qquad \bigsqcap_i^n \mathsf{G}_i \sqcap \mathsf{S} \sqsubseteq \bigsqcap_i^n \mathsf{A}_i}{\mathsf{S} \sqsubseteq \bigsqcap_i^n \mathsf{G}_i} \text{ if } \bigwedge_{1 \leq i \leq n} \mathrm{NB}^{\mathrm{AG}}[\mathsf{S}, \mathsf{A}_i, \mathsf{G}_i].$$

Proof. Let $n \in \mathbb{N}$. That $\mathrm{AG}_n^{\mathrm{AG}}$ is an A-G rule[6] is obvious. We show that $\mathrm{AG}_n^{\mathrm{AG}}$ is weaker than AG'_n using Corollary 3.36. As substitution we take $\mathrm{id}|_{var(\mathrm{AG}'_n)}$. Then the conditions 1 and 2 are trivially true. Condition 3 is also true, as for all $i \in \{1, \ldots, n\}$, $\mathrm{NB}^{\mathrm{AG}}[\mathsf{S}, \mathsf{A}_i, \mathsf{G}_i] \models^{\boldsymbol{\mathcal{L}}_{\mathbf{AG}}} \mathrm{NB}[\mathsf{S}, \mathsf{A}_i, \mathsf{G}_i]$ by definition of the ternary relation $\mathrm{NB}^{\mathrm{AG}}$ and Proposition 5.66.

By Proposition 3.39, soundness of $\mathrm{AG}_n^{\mathrm{AG}}$ follows from soundness of AG'_n, see Theorem 4.34 and Proposition 3.43. Note that soundness of AG'_n requires $\boldsymbol{\mathcal{L}}_{\mathbf{AG}}$ to be well-approximable, which it is according to Proposition 5.48. □

The above A-G rules $\mathrm{AG}_n^{\mathrm{AG}}$ do not use all of the structure that $\boldsymbol{\mathcal{L}}_{\mathbf{AG}}$ offers; in particular, implication is not used. The following corollary essentially reformulates the rules $\mathrm{AG}_n^{\mathrm{AG}}$ into a more convenient form, using implication in premises and conclusion, while still retaining the side condition.

Corollary 5.69. *Let $n \in \mathbb{N}$, $\sigma \in \Sigma^\infty$ and $E, E_1, \ldots, E_n, M, M_1, \ldots, M_n \in \mathcal{L}_{\mathrm{AG}}$. If for all $i \in \{1, \ldots, n\}$, $\sigma \in E_i \xrightarrow{+} M_i$ and at least one out of $\{E_i, M_i\}$ is a safety property then* (5.3) *is sound.*

$$\frac{\sigma \in E_1 \Rightarrow M_1 \;\ldots\; \sigma \in E_n \Rightarrow M_n \qquad \sigma \in E \wedge \bigwedge_{i=1}^n M_i \Rightarrow \bigwedge_{i=1}^n E_i \wedge M}{\sigma \in E \Rightarrow M} \tag{5.3}$$

[6] Concerning the last premise, which is not in strong normal form, see the comment following the Definition of AG'_n in Section 4.5.

Proof. Assume for all $i \in \{1,\dots,n\}$, $\sigma \in E_i \xrightarrow{+} M_i$ and there is $P \in \{E_i, M_i\}$ such that $P = \mathit{safe}(P)$. We will reduce soundness of (5.3) to soundness of $\mathrm{AG}_n^{\mathrm{AG}}$ (Theorem 5.68). Assume that the premises of (5.3) hold. For every $i \in \{1,\dots,n\}$, by monotonicity $E_i \Rightarrow M_i \leq E_i \wedge E \Rightarrow M_i$, so the i-th premise implies $\preceq(\sigma) \leq E_i \wedge E \Rightarrow M_i$, which is equivalent to $E_i \wedge (\preceq(\sigma) \wedge E) \leq M_i$ by Proposition 5.52(1). Similarly, the last premise implies $\bigwedge_{i=1}^n M_i \wedge (\preceq(\sigma) \wedge E) \leq \bigwedge_{i=1}^n E_i \wedge M \leq \bigwedge_{i=1}^n E_i$ by Proposition 5.52(1) and monotonicity. Moreover for every $i \in \{1,\dots,n\}$, we have $\langle \preceq(\sigma) \wedge E, E_i, M_i \rangle \in \mathrm{NB}^{\mathrm{AG}}$ because $\sigma \in E_i \xrightarrow{+} M_i$ implies $\preceq(\sigma) \wedge E \leq E_i \xrightarrow{+} M_i$.

We define the valuation $\alpha \in V(\boldsymbol{\mathcal{L}}_{\mathbf{AG}})$ with $\mathit{dom}(\alpha) = \mathit{var}(\mathrm{AG}_n^{\mathrm{AG}})$ by $\alpha(\mathsf{S}) = \preceq(\sigma) \wedge E$ and for all $i \in \{1,\dots,n\}$, $\alpha(\mathsf{A}_i) = E_i$ and $\alpha(\mathsf{G}_i) = M_i$. Then $\boldsymbol{\mathcal{L}}_{\mathbf{AG}}, \alpha \models \bigsqcap_i^n \mathsf{G}_i \sqcap \mathsf{S} \sqsubseteq \bigsqcap_i^n \mathsf{A}_i$ and for all $i \in \{1,\dots,n\}$, $\boldsymbol{\mathcal{L}}_{\mathbf{AG}}, \alpha \models \mathsf{A}_i \sqcap \mathsf{S} \sqsubseteq \mathsf{G}_i$ and $\boldsymbol{\mathcal{L}}_{\mathbf{AG}}, \alpha \models \mathrm{NB}^{\mathrm{AG}}[\mathsf{S}, \mathsf{A}_i, \mathsf{G}_i]$, i. e., premises and side condition of $\mathrm{AG}_n^{\mathrm{AG}}$ are true under α. By soundness of $\mathrm{AG}_n^{\mathrm{AG}}$, we infer $\boldsymbol{\mathcal{L}}_{\mathbf{AG}}, \alpha \models \mathsf{S} \sqsubseteq \bigsqcap_i^n \mathsf{G}_i$, so $\preceq(\sigma) \wedge E \leq \bigwedge_{i=1}^n M_i$, which is equivalent to $\preceq(\sigma) \leq E \Rightarrow \bigwedge_{i=1}^n M_i$ by Proposition 5.52(1).

The last premise of (5.3) implies $\preceq(\sigma) \leq E \wedge \bigwedge_{i=1}^n M_i \Rightarrow \bigwedge_{i=1}^n E_i \wedge M \leq E \wedge \bigwedge_{i=1}^n M_i \Rightarrow M = \bigwedge_{i=1}^n M_i \Rightarrow (E \Rightarrow M)$ by monotonicity and Proposition 5.52(7). Together with the entailment that originates from the conclusion of $\mathrm{AG}_n^{\mathrm{AG}}$, by Cut we get $\preceq(\sigma) \leq (E \Rightarrow \bigwedge_{i=1}^n M_i) \wedge \big(\bigwedge_{i=1}^n M_i \Rightarrow (E \Rightarrow M)\big) \leq E \Rightarrow (E \Rightarrow M) = E \wedge E \Rightarrow M = E \Rightarrow M$. This is equivalent to the conclusion of (5.3). □

Note that the above corollary restricts circular reasoning to safety properties only. Circular reasoning with liveness properties is not possible because for each A-G spec, either the assumption or the guarantee (or both) must be a safety property. Contrary to the literature [AL95, JT96, VV01], however, we do not a priori confine the assumptions to be safety properties.

Subsequently, we focus on variants of (5.3) specifically tailored for composing assume-guarantee specifications. These rules need not employ a side condition because their premises express all relevant constraints in our logic using (strong) assume-guarantee specifications and safety resp. liveness closures. Still, soundness of these variants is derived via Corollary 5.69 from soundness of (5.3).

Corollary 5.70. *Let $n \in \mathbb{N}$ and $E, E_1, \dots, E_n, M, M_1, \dots, M_n \in \mathcal{L}_{\mathrm{AG}}$. Then* (5.4) *is sound.*

$$\frac{E \wedge \bigwedge_{i=1}^n \mathit{safe}(M_i) \leq \bigwedge_{i=1}^n E_i \qquad \Sigma^\infty \leq E \xrightarrow{+} (\bigwedge_{i=1}^n M_i \Rightarrow M)}{\bigwedge_{i=1}^n (E_i \xrightarrow{+} M_i) \leq E \xrightarrow{+} M} \tag{5.4}$$

Proof. Assume that the premises of (5.4) hold. We have to show that the conjunction of A-G specs $\bigwedge_{i=1}^n (E_i \xrightarrow{+} M_i)$ entails the A-G spec $E \xrightarrow{+} M$, so choose an arbitrary $\sigma \in \Sigma^\infty$ with $\sigma \in E_i \xrightarrow{+} M_i$ for all $i \in \{1,\dots,n\}$. Then we have to show that $\sigma \in E \xrightarrow{+} M$.

For every $i \in \{1,\dots,n\}$, $E_i \xrightarrow{+} M_i = E_i \xrightarrow{+} \mathit{safe}(M_i) \leq E_i \Rightarrow \mathit{safe}(M_i)$ by Proposition 5.65, so we have $\sigma \in E_i \Rightarrow \mathit{safe}(M_i)$. Moreover by the first premise, $\preceq(\sigma) \wedge E \wedge \bigwedge_{i=1}^n \mathit{safe}(M_i) \leq \bigwedge_{i=1}^n E_i \wedge \big(\bigwedge_{i=1}^n \mathit{safe}(M_i)\big)$, so by Proposition 5.51 we get $\sigma \in E \wedge \bigwedge_{i=1}^n \mathit{safe}(M_i) \Rightarrow \bigwedge_{i=1}^n E_i \wedge \big(\bigwedge_{i=1}^n \mathit{safe}(M_i)\big)$. Thus, the premises of (5.3) hold. By Corollary 5.69, (5.3) is sound since $\sigma \in E_i \xrightarrow{+} \mathit{safe}(M_i)$ for all $i \in \{1,\dots,n\}$. Hence, we get $\sigma \in E \Rightarrow \bigwedge_{i=1}^n \mathit{safe}(M_i)$, which is equivalent to $\preceq(\sigma) \wedge E \leq \bigwedge_{i=1}^n \mathit{safe}(M_i)$ by Proposition 5.51. For this reason, $\preceq(\sigma) \wedge E = \preceq(\sigma) \wedge E \wedge \bigwedge_{i=1}^n \mathit{safe}(M_i) \leq \bigwedge_{i=1}^n E_i$ by the first premise.

To prove that $\sigma \in E \xrightarrow{+} M$, choose any $\tau \in \Sigma^*$ with $\tau \preceq \sigma$ and $\prec(\tau) \subseteq E$. Then we have $\prec(\tau) \subseteq \preceq(\sigma) \wedge E \subseteq \bigwedge_{i=1}^n E_i$. So for every $i \in \{1,\dots,n\}$, we have $\prec(\tau) \subseteq E_i$, and as $\sigma \in E_i \xrightarrow{+} M_i$, we get $\tau \in M_i$ by the definition of A-G specs. Thus, we have $\tau \in \bigwedge_{i=1}^n M_i$. As $\sigma \in E \xrightarrow{+} (\bigwedge_{i=1}^n M_i \Rightarrow M)$ by the second premise, we also have $\tau \in \bigwedge_{i=1}^n M_i \Rightarrow M$. By Modus Ponens, $(\bigwedge_{i=1}^n M_i) \wedge (\bigwedge_{i=1}^n M_i \Rightarrow M) \leq M$, so we get $\tau \in M$. Thus, we conclude $\sigma \in E \xrightarrow{+} M$ by the definition of A-G specs. □

Note the second premise of (5.4). Σ^∞ on the left-hand side of the entailment means that the property $E \xrightarrow{+} (\bigwedge_{i=1}^n M_i \Rightarrow M)$ on the right-hand side is true of all behaviors, i. e., it is equivalent to the logical constant *true*. Also note that (5.4) essentially confines the guarantees M_i to safety properties, as the first premise only takes into account their safety closure. For the other occurrences of the M_i, it follows from the propositions 5.65 and 5.61 that their liveness part is irrelevant anyway.

Rule (5.4) admits to derive an A-G spec from the conjunction of n A-G specs. Now, we show that a slight strengthening of the premises enables us to derive even a strong A-G spec from the same composition.

Corollary 5.71. *Let $n \in \mathbb{N}$ and $E, E_1, \dots, E_n, M, M_1, \dots, M_n \in \mathcal{L}_{\mathrm{AG}}$. Then* (5.5) *is sound.*

$$\frac{E \wedge \bigwedge_{i=1}^n \mathit{safe}(M_i) \leq \bigwedge_{i=1}^n E_i \wedge \mathit{live}(M) \qquad \Sigma^\infty \leq E \xrightarrow{+} (\bigwedge_{i=1}^n M_i \Rightarrow M)}{\bigwedge_{i=1}^n (E_i \xrightarrow{+} M_i) \leq E \overset{+}{\Rightarrow} M} \tag{5.5}$$

Proof. Assume that the premises of (5.5) hold. We have to show that the conjunction of A-G specs $\bigwedge_{i=1}^n (E_i \xrightarrow{+} M_i)$ entails the strong A-G spec $E \overset{+}{\Rightarrow} M$, so choose an arbitrary $\sigma \in \Sigma^\infty$ with $\sigma \in E_i \xrightarrow{+} M_i$ for all $i \in \{1,\dots,n\}$. Then we have to show that $\sigma \in E \xrightarrow{+} M$ and $\sigma \in E \Rightarrow M$.

By the first premise, we have $E \wedge \bigwedge_{i=1}^n \mathit{safe}(M_i) \leq \bigwedge_{i=1}^n E_i$, so from this and the second premise, we obtain $\bigwedge_{i=1}^n (E_i \xrightarrow{+} M_i) \leq E \xrightarrow{+} M$ by (5.4). Thus, we have proven $\sigma \in E \xrightarrow{+} M$ and also, by Proposition 5.65, $\sigma \in E \Rightarrow \mathit{safe}(M)$.

For every $i \in \{1,\dots,n\}$, $E_i \xrightarrow{+} M_i = E_i \xrightarrow{+} \mathit{safe}(M_i) \leq E_i \Rightarrow \mathit{safe}(M_i)$ by Proposition 5.65, so $\sigma \in E_i \Rightarrow \mathit{safe}(M_i)$. Moreover, by the first premise we have

$\preceq(\sigma) \wedge E \wedge \bigwedge_{i=1}^{n} \mathit{safe}(M_i) \leq \bigwedge_{i=1}^{n} E_i \wedge \mathit{live}(M)$, which, by Proposition 5.52(1), is equivalent to $\sigma \in E \wedge \bigwedge_{i=1}^{n} \mathit{safe}(M_i) \Rightarrow \bigwedge_{i=1}^{n} E_i \wedge \mathit{live}(M)$. Thus, the premises of (5.3) hold. By Corollary 5.69, (5.3) is sound because $\sigma \in E_i \xrightarrow{+} \mathit{safe}(M_i)$ for all $i \in \{1, \ldots, n\}$. Hence, the conclusion of (5.3) yields $\sigma \in E \Rightarrow \mathit{live}(M)$. Together with $\sigma \in E \Rightarrow \mathit{safe}(M)$, this implies $\sigma \in E \Rightarrow M$ for by the propositions 5.52(6) and 5.60, $\big(E \Rightarrow \mathit{safe}(M)\big) \wedge \big(E \Rightarrow \mathit{live}(M)\big) = E \Rightarrow \mathit{safe}(M) \wedge \mathit{live}(M) = E \Rightarrow M$. □

As strong A-G specs are stronger than A-G specs, the soundness of a variant of (5.5) for composing strong A-G specs is an immediate consequence of Corollary 5.71.

Corollary 5.72. *Let $n \in \mathbb{N}$ and $E, E_1, \ldots, E_n, M, M_1, \ldots, M_n \in \mathcal{L}_{\mathrm{AG}}$. Then* (5.6) *is sound.*

$$\frac{E \wedge \bigwedge_{i=1}^{n} \mathit{safe}(M_i) \leq \bigwedge_{i=1}^{n} E_i \wedge \mathit{live}(M) \qquad \Sigma^{\infty} \leq E \overset{+}{\Rightarrow} (\bigwedge_{i=1}^{n} M_i \Rightarrow M)}{\bigwedge_{i=1}^{n} (E_i \overset{+}{\Rightarrow} M_i) \leq E \overset{+}{\Rightarrow} M} \tag{5.6}$$

Proof. Soundness of (5.6) follows from soundness of (5.5) because of the entailments $E \overset{+}{\Rightarrow} (\bigwedge_{i=1}^{n} M_i \Rightarrow M) \leq E \xrightarrow{+} (\bigwedge_{i=1}^{n} M_i \Rightarrow M)$ and $E_i \overset{+}{\Rightarrow} M_i \leq E_i \xrightarrow{+} M_i$ for all $i \in \{1, \ldots, n\}$. □

To end this section, we present a rule for composing strong A-G specs which confines the assumptions to safety properties instead of the guarantees. In large parts, the following proof of soundness will be very similar to the proof of Corollary 5.70.

Corollary 5.73. *Let $n \in \mathbb{N}$ and $E, E_1, \ldots, E_n, M, M_1, \ldots, M_n \in \mathcal{L}_{\mathrm{AG}}$. Then* (5.7) *is sound.*

$$\frac{E \wedge \bigwedge_{i=1}^{n} M_i \leq \bigwedge_{i=1}^{n} \mathit{safe}(E_i) \qquad \Sigma^{\infty} \leq E \overset{+}{\Rightarrow} (\bigwedge_{i=1}^{n} M_i \Rightarrow M)}{\bigwedge_{i=1}^{n} (\mathit{safe}(E_i) \overset{+}{\Rightarrow} M_i) \leq E \overset{+}{\Rightarrow} M} \tag{5.7}$$

Proof. Assume that the premises of (5.7) hold. We have to show that the conjunction of strong A-G specs $\bigwedge_{i=1}^{n} (\mathit{safe}(E_i) \overset{+}{\Rightarrow} M_i)$ entails the strong A-G spec $E \overset{+}{\Rightarrow} M$, so choose an arbitrary $\sigma \in \Sigma^{\infty}$ with $\sigma \in \mathit{safe}(E_i) \overset{+}{\Rightarrow} M_i$ for all $i \in \{1, \ldots, n\}$. Then we have to show that $\sigma \in E \xrightarrow{+} M$ and $\sigma \in E \Rightarrow M$.

For every $i \in \{1, \ldots, n\}$, $\mathit{safe}(E_i) \overset{+}{\Rightarrow} M_i = (\mathit{safe}(E_i) \xrightarrow{+} M_i) \wedge (\mathit{safe}(E_i) \Rightarrow M_i)$, so we have $\sigma \in E_i \Rightarrow \mathit{safe}(M_i)$. Moreover by the first premise, $\preceq(\sigma) \wedge E \wedge \bigwedge_{i=1}^{n} M_i \leq \bigwedge_{i=1}^{n} \mathit{safe}(E_i) \wedge (\bigwedge_{i=1}^{n} M_i)$, so we get $\sigma \in E \wedge \bigwedge_{i=1}^{n} M_i \Rightarrow \bigwedge_{i=1}^{n} \mathit{safe}(E_i) \wedge (\bigwedge_{i=1}^{n} M_i)$ by Proposition 5.51. Thus, the premises of (5.3) hold. By Corollary 5.69, (5.3) is sound since $\sigma \in \mathit{safe}(E_i) \xrightarrow{+} M_i$ for all $i \in \{1, \ldots, n\}$. Therefore, we get $\sigma \in E \Rightarrow \bigwedge_{i=1}^{n} M_i$, which is equivalent to $\preceq(\sigma) \wedge E \leq \bigwedge_{i=1}^{n} M_i$ by Proposition 5.51. For this reason, $\preceq(\sigma) \wedge E = \preceq(\sigma) \wedge E \wedge \bigwedge_{i=1}^{n} M_i \leq \bigwedge_{i=1}^{n} \mathit{safe}(E_i)$ by the first premise.

To prove that $\sigma \in E \xrightarrow{+} M$, choose any $\tau \in \Sigma^*$ with $\tau \preceq \sigma$ and $\prec(\tau) \subseteq E$. Then we have $\prec(\tau) \subseteq \preceq(\sigma) \wedge E \subseteq \bigwedge_{i=1}^{n} \mathit{safe}(E_i)$. So for every $i \in \{1, \ldots, n\}$, we have $\prec(\tau) \subseteq \mathit{safe}(E_i)$, and as $\sigma \in \mathit{safe}(E_i) \xrightarrow{+} M_i$, we get $\tau \in M_i$ by the definition of A-G specs. Thus, we have $\tau \in \bigwedge_{i=1}^{n} M_i$. As $E \overset{+}{\Rightarrow} (\bigwedge_{i=1}^{n} M_i \Rightarrow M) \leq E \xrightarrow{+} (\bigwedge_{i=1}^{n} M_i \Rightarrow M)$, the second premise yields $\sigma \in E \xrightarrow{+} (\bigwedge_{i=1}^{n} M_i \Rightarrow M)$, and so $\prec(\tau) \subseteq E$ implies $\tau \in \bigwedge_{i=1}^{n} M_i \Rightarrow M$. By Modus Ponens, $(\bigwedge_{i=1}^{n} M_i) \wedge (\bigwedge_{i=1}^{n} M_i \Rightarrow M) \leq M$, so we get $\tau \in M$. Thus, we conclude $\sigma \in E \xrightarrow{+} M$ by the definition of A-G specs.

To prove $\sigma \in E \Rightarrow M$, note that we obtain $\sigma \in \bigwedge_{i=1}^{n} M_i \Rightarrow (E \Rightarrow M)$ from the second premise as $E \overset{+}{\Rightarrow} (\bigwedge_{i=1}^{n} M_i \Rightarrow M) \leq E \Rightarrow (\bigwedge_{i=1}^{n} M_i \Rightarrow M) = \bigwedge_{i=1}^{n} M_i \Rightarrow (E \Rightarrow M)$ by Proposition 5.52(7). Recall that we have already proven $\sigma \in E \Rightarrow \bigwedge_{i=1}^{n} M_i$. Hence, $\sigma \in E \Rightarrow M$ follows since by Cut, $\big(E \Rightarrow \bigwedge_{i=1}^{n} M_i\big) \wedge \big(\bigwedge_{i=1}^{n} M_i \Rightarrow (E \Rightarrow M)\big) \leq E \Rightarrow (E \Rightarrow M) = E \wedge E \Rightarrow M = E \Rightarrow M$. □

5.2.6 Comparison to Other Work

Quite some amount of work has been devoted to the development of rules or proof systems that deal with composing assume-guarantee specifications. Some of these rules look strikingly similar to the rules from the corollaries 5.70 to 5.73. Yet, subtle differences exist.

Close to ours comes the work of Abadi and Plotkin. Like we do, they develop a composition principle for assume-guarantee specifications in an intuitionistic temporal logic [AP93, Theorem 1], which looks very similar to a variant of (5.4).[7] However, they interpret their logic over finite behaviors only, so they confine their whole work to safety properties. Also, their semantics is an interleaving semantics, which explicitly records which steps are executed by the system and which by the environment. In this setting, presuming that an assumption E constrains only the steps of the environment whereas a guarantee M constrains only the steps of the system, the A-G spec $E \xrightarrow{+} M$ is equivalent to the implication $E \Rightarrow M$. For this reason, [AP93] adopts intuitionistic implication as their operator for constructing assume-guarantee specifications.

Abadi and Lamport present the composition principle in classical temporal logics, where formulas are interpreted over infinite behaviors. In [AL93] they use an interleaving semantics similar to the one in [AP93], whereas [AL95] establishes the composition principle for TLA, Lamport's Temporal Logic of Actions [Lam94], which does not force interleaving semantics. In this later work, they define an operator $\overset{+}{\rightarrow}$ for constructing assume-guarantee specifications in essentially the same way than we have defined our operator $\overset{+}{\Rightarrow}$ except they use classical instead of intuitionistic implication. Still, their composition rule [AL95, Theorem 3] resembles closely (5.7). To see how closely, note that Abadi and Lamport presume the assumptions E_i always to be safety properties. Thus, the

[7] Drop the second premise of (5.4) and replace M in the conclusion by $\bigwedge_{i=1}^{n} M_i$.

first premise of (5.7), which by the propositions 5.55 and 5.61 is equivalent to $safe(E) \wedge \bigwedge_{i=1}^{n} safe(M_i) \leq \bigwedge_{i=1}^{n} safe(E_i)$, corresponds to condition 1 of [AL95, Theorem 3]. Furthermore, the second premise of (5.7) can be split into two premises that correspond to the conditions 2(a) and 2(b) there, because

$$E \stackrel{+}{\Rightarrow} (\bigwedge_{i=1}^{n} M_i \Rightarrow M) = \Big(safe(E) \stackrel{+}{\Rightarrow} (\bigwedge_{i=1}^{n} safe(M_i) \Rightarrow safe(M)) \Big) \wedge \Big(E \Rightarrow (\bigwedge_{i=1}^{n} M_i \Rightarrow M) \Big)$$

by the propositions 5.65 and 5.61. Interestingly, Abadi and Merz [AM95] re-prove soundness of the composition principle in TLA by embedding TLA into an intuitionistic logic interpreted over well-founded Kripke frames. In that logic, they define their assume-guarantee operator exactly like we have defined $\stackrel{+}{\rightarrow}$, and they reduce soundness of the TLA composition principle for safety properties [AM95, Theorem 4] to soundness of a variant of (5.4).

Similar to [AL95] is the work of Jonsson and Tsay [JT96, Tsa00], only they do not work in TLA but in linear-time temporal logic (LTL) with past temporal operators [MP92]. This enables them to give a syntactic definition of Abadi's operator $\stackrel{+}{\triangleright}$ (which they denote by $\triangleright$) for constructing A-G specs instead of a semantic one. Just like [AL95], Jonsson and Tsay confine assumptions to safety properties, and not surprisingly, their composition rule [JT96, Theorem 11] is in direct analogy to the one of Abadi and Lamport (even the structure of the premises is the same) and hence also to our rule (5.7) — except for the use of classical instead of intuitionistic implication, again.

The Viswanathans [VV01] generalize assume-guarantee specifications. As properties they adopt least and greatest fix points of ω-continuous and ω-co-continuous functions on properties, respectively, where properties are just sets of computations (and computations can be anything — sequences, trees, et cetera). They define the assume-guarantee operator via the chain of iterative approximations that converges to the fix point; notice that (co-)continuity ensures convergence within at most ω steps. More precisely, they say that a computation satisfies an assume-guarantee specification if for all $k \geq 0$, whenever σ satisfies the k-th approximation of the assumption then it must satisfy some strictly later approximation of the guarantee. If the guarantee is a greatest fix point then the above definition can actually be strengthened to: σ satisfies the A-G spec if for all $k \geq 0$, whenever σ satisfies the k-th approximation of the assumption then it must satisfy the $(k+1)$-st approximation of the guarantee. In this setting, the authors present a number of generic rules for composing A-G specs, whereby the premises of the rules depend on whether assumptions resp. guarantees are least or greatest fix points — truly circular rules are possible only when the assumptions are confined to greatest fix points. They show the generality of their rules by proving several known assume-guarantee rules, one for A-G specs in LTL [McM99] and one for trace tree containment of Moore machines [HQRT02], to be special instances of their rules.

McMillan [McM99] presents an assume-guarantee rule for establishing LTL formulas of the form $\mathbf{G}\varphi$, with φ being an arbitrary LTL formula. The premises of his rule are assume-guarantee specifications constructed via negation and the until operator $\mathbf{U}$. In fact, as [VV01] observes, in LTL the formulas $\neg(\varphi\ \mathbf{U}\ \neg\psi)$ and $\mathbf{G}\varphi \triangleright \mathbf{G}\psi$ are equivalent, where $\triangleright$ denotes the LTL-equivalent to our assume-guarantee operator $\xrightarrow{+}$. Thus, McMillan has found one more way to define the assume-guarantee operator syntactically, even without recurring to past operators. Moreover, his rule does not confine assumptions or guarantees to safety properties, i. e., liveness properties are admitted everywhere. The same applies to the rules in [VV01] as the authors prove McMillan's rule to be an instance of their rules. Whether our assume-guarantee framework can also be instantiated to McMillan's rule needs to be investigated still.

Finally, it should be noted that assume-guarantee specifications do not require full temporal logic. A number of proof systems for proving (partial) correctness of concurrent or distributed programs are built around so called *assumption-commitment* or *rely-guarantee* specifications. These are triples consisting of a program (fragment) P, an assumption A about P's environment and P's guarantee G, frequently written in a notation like $\{A\}P\{G\}$ which resembles Hoare triples [Hoa69] for specifying pre- and post-conditions of sequential programs. (In deed, one may think of A and G as pre- and post-condition, respectively, of each atomic step of the program P.) The underlying model of concurrency of such approaches is interleaving execution of program and environment. Thus, the assumption A typically is a predicate over the state space of the system expressing in which states the program P expects to find the system (after a step of the environment). The guarantee G is a predicate over the possible transitions of P, so the reading of $\{A\}P\{G\}$ is: Whenever P finds the system in a state that satisfies A then the atomic step which P (maybe non-deterministically) chooses to execute must satisfy G. A detailed account on these formalisms (along with a very extensive list of references) can be found in the recent book by de Roever et al. [dRdBH$^+$00], which also includes several proof systems that admit circular reasoning with assumption-commitment resp. rely-guarantee specifications.

Bibliography

[AG00] Rajeev Alur and Radu Grosu. Modular refinement of hierarchic reactive machines. In *Proceedings of the 27th ACM Symposium on Principles of Programming Languages (POPL)*, pages 390–402. ACM Press, 2000.

[AH99] Rajeev Alur and Thomas A. Henzinger. Reactive modules. *Formal Methods in System Design*, 15(1):7–48, 1999. A preliminary version appears in *11th IEEE Symposium on Logic in Computer Science (LICS)*, 1996.

[AL93] Martín Abadi and Leslie Lamport. Composing specifications. *ACM Transactions on Programming Languages and Systems*, 15(1):73–132, 1993.

[AL95] Martín Abadi and Leslie Lamport. Conjoining specifications. *ACM Transactions on Programming Languages and Systems*, 17(3):507–534, 1995.

[AM95] Martín Abadi and Stephan Merz. An abstract account of composition. In *20th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, LNCS 969, pages 499–508. Springer, 1995.

[AP93] Martín Abadi and Gordon D. Plotkin. A logical view of composition. *Theoretical Computer Science*, 114(1):3–30, 1993.

[AS85] Bowen Alpern and Fred B. Schneider. Defining liveness. *Information Processing Letters*, 21:181–185, 1985.

[BRJ98] Grady Booch, James Rumbaugh, and Ivar Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, 1998.

[CES86] Edmund M. Clarke, E. Allen Emerson, and A. Prasad Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, 1986.

[CLM89] Edmund M. Clarke, David E. Long, and Ken L. McMillan. Compositional model checking. In *4th Symposium on Logic in Computer Science (LICS)*, pages 353–362. IEEE Computer Society, 1989.

[Dav93] Brian A. Davey. Duality theory on ten dollars a day. In I. G. Rosenberg and G. Sabidussi, editors, *Algebras and Orders*, volume C 389 of *NATO ASI Series*, pages 71–111. Kluwer Academic Publishers, 1993.

[DG84] William F. Dowling and Jean H. Gallier. Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *Journal of Logic Programming*, 1(3):267–284, 1984.

[DP90] Brian A. Davey and Hilary A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 1990.

[DP99] Giorgio Delzanno and Andreas Podelski. Model checking in CLP. In *5th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS)*, LNCS 1579, pages 223–239. Springer, 1999.

[DP01] Giorgio Delzanno and Andreas Podelski. Constraint-based deductive model checking. *International Journal on Software Tools for Technology Transfer (STTT)*, 3(3):250–270, 2001.

[dRdBH+00] Willem-Paul de Roever, Frank de Boer, Ulrich Hannemann, Jozef Hooman, Yassine Lakhnech, Mannes Poel, and Job Zwiers. *Concurrency Verification: Introduction to Compositional and Noncompositional Methods*. Cambridge University Press, 2000.

[FFQ02] Cormac Flangan, Stephen N. Freund, and Shaz Qadeer. Thread-modular verification for shared-memory programs. In *11th European Symposium on Programming (ESOP)*, LNCS 2305, pages 262–277. Springer, 2002.

[GL94] Orna Grumberg and David E. Long. Model checking and modular verification. *ACM Transactions on Programming Languages and Systems*, 16(3):843–871, 1994.

[Gol84] Robert Goldblatt. *Topoi: The Categorical Analysis of Logic*, volume 98 of *Studies in logic and the foundations of mathematics*. North-Holland, 1984.

[Har87] David Harel. Statecharts: A visual formulation for complex systems. *Science of Computer Programming*, 8(3):231–274, 1987.

[HLQR99] Thomas A. Henzinger, Xiaojun Liu, Shaz Qadeer, and Sriram K. Rajamani. Formal specification and verification of a dataflow processor array. In *International Conference on Computer-Aided Design (ICCAD)*, pages 494–499. IEEE Computer Society Press, 1999.

[HMP01] Thomas A. Henzinger, Marius Minea, and Vinayak Prabhu. Assume-guarantee reasoning for hierarchical hybrid systems. In *4th International Workshop on Hybrid Systems: Computation and Control (HSCC)*, LNCS 2034, pages 275–290. Springer, 2001.

[Hoa69] C. A. R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12(10):576–580, 1969.

[HQR98] Thomas A. Henzinger, Shaz Qadeer, and Sriram K. Rajamani. You assume, we guarantee: Methodology and case studies. In *10th International Conference on Computer Aided Verification (CAV)*, LNCS 1427, pages 440–451. Springer, 1998.

[HQR00] Thomas A. Henzinger, Shaz Qadeer, and Sriram K. Rajamani. Decomposing refinement proofs using assume-guarantee reasoning. In *International Conference on Computer-Aided Design (ICCAD)*, pages 245–252. IEEE Computer Society Press, 2000.

[HQRT02] Thomas A. Henzinger, Shaz Qadeer, Sriram K. Rajamani, and Serdar Tasiran. An assume-guarantee rule for checking simulation. *ACM Transactions on Programming Languages and Systems*, 24(1):51–64, 2002.

[Joh82] Peter T. Johnstone. *Stone spaces*, volume 3 of *Cambridge studies in advanced mathematics*. Cambridge University Press, 1982.

[Jon81] Cliff B. Jones. *Development Methods for Computer Programs Including a Notion of Interference*. PhD thesis, Oxford University, 1981.

[JT96] Bengt Jonsson and Yih-Kuen Tsay. Assumption/guarantee specifications in linear-time temporal logic. *Theoretical Computer Science*, 167(1–2):47–72, 1996.

[Kur94] Robert P. Kurshan. *Computer-aided Verification of Coordinating Processes*. Princeton University Press, 1994.

[KV00] Orna Kupferman and Moshe Y. Vardi. An automata-theoretic approach to modular model checking. *ACM Transactions on Programming Languages and Systems*, 22(1):87–128, 2000.

[Lam94] Leslie Lamport. The temporal logic of actions. *ACM Transactions on Programming Languages and Systems*, 16(3):872–923, 1994.

[LP85] Orna Lichtenstein and Amir Pnueli. Checking that finite state concurrent programs satisfy their linear specification. In *Proceedings of the 12th ACM Symposium on Principles of Programming Languages (POPL)*, pages 97–107, 1985.

[Mai01] Patrick Maier. A set-theoretic framework for assume-guarantee reasoning. In *28th International Colloquium on Automata, Languages and Programming (ICALP)*, LNCS 2076, pages 821–834. Springer, 2001.

[Mai02] Patrick Maier. A framework for circular assume-guarantee rules. In *Symposium on the Effectiveness of Logic in Computer Science*, Saarbrücken, Germany, March 2002. `http://www.mpi-sb.mpg.de/conferences/elics02/index.html`.

[Mai03] Patrick Maier. Compositional circular assume-guarantee rules cannot be sound and complete. In *6th International Conference on Foundations of Software Science and Computation Structures (FOSSACS)*, LNCS 2620, pages 343–357. Springer, 2003.

[MC81] Jayadev Misra and K. Mani Chandy. Proofs of networks of processes. *IEEE Transactions on Software Engineering*, 7(4):417–426, 1981.

[McM97] Ken L. McMillan. A compositional rule for hardware design refinement. In *9th International Conference on Computer Aided Verification (CAV)*, LNCS 1254, pages 207–218. Springer, 1997.

[McM98] Ken L. McMillan. Verification of an implementation of Tomasulo's algorithm by compositional model checking. In *10th International Conference on Computer Aided Verification (CAV)*, LNCS 1427, pages 110–121. Springer, 1998.

[McM99] Ken L. McMillan. Circular compositional reasoning about liveness. In *10th IFIP WG 10.5 Advanced Research Working Conference on Correct Hardware Design and Verification Methods (CHARME)*, LNCS 1703, pages 342–345. Springer, 1999.

[Mel93] Thomas F. Melham. *Higher Order Logic and Hardware Verification*, volume 31 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1993.

[Mil71] Robin Milner. An algebraic definition of simulation between programs. In *Proceedings of the 2nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 481–489. William Kaufmann, 1971.

[Mil89] Robin Milner. *Communication and Concurrency*. Prentice Hall international series in computer science. Prentice Hall, 1989.

[MP92] Zohar Manna and Amir Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. Springer, 1992.

[MP95] Zohar Manna and Amir Pnueli. *Temporal Verification of Reactive Systems: Safety*. Springer, 1995.

[NT00] Kedar S. Namjoshi and Richard J. Trefler. On the completeness of compositional reasoning. In *12th International Conference on Computer Aided Verification (CAV)*, LNCS 1855, pages 139–153. Springer, 2000.

[Pnu85] Amir Pnueli. In transition from global to modular temporal reasoning about programs. In K. R. Apt, editor, *Logics and Models of Concurrent Systems*, volume F 13 of *NATO ASI Series*, pages 123–144. Springer, 1985.

[Pod00] Andreas Podelski. Model checking as constraint solving. In *7th International Symposium on Static Analysis (SAS)*, LNCS 1824, pages 22–37. Springer, 2000.

[QS82] Jean-Pierre Queille and Joseph Sifakis. Specification and verification of concurrent systems in CESAR. In *5th International Symposium on Programming*, LNCS 137, pages 337–351. Springer, 1982.

[Raj99] Sriram K. Rajamani. *New Directions in Refinement Checking*. PhD thesis, University of California, Berkeley, 1999.

[Rog87] Hartley Rogers. *Theory of Recursive Functions and Effective Computability*. MIT Press, 1987.

[RR01] Sriram K. Rajamani and Jakob Rehof. A behavioral module system for the pi-calculus. In *8th International Symposium on Static Analysis (SAS)*, LNCS 2126, pages 375–394. Springer, 2001.

[SS03] Viorica Sofronie-Stokkermans. Representation theorems and the semantics of non-classical logics, and applications to automated theorem proving. In M. Fitting and E. Orlowska, editors, *Theory and Applications of Multiple-Valued Logic*, Studies in Fuzziness and Soft Computing 114, pages 59–100. Springer, 2003.

[TAKB96] Serdar Tasiran, Rajeev Alur, Robert P. Kurshan, and Robert K. Brayton. Verifying abstractions of timed systems. In *7th International Conference on Concurrency Theory (CONCUR)*, LNCS 1119, pages 546–562. Springer, 1996.

[TB97] Serdar Tasiran and Robert K. Brayton. Stari: A case study in compositional and hierarchical timing verification. In *9th International Conference on Computer Aided Verification (CAV)*, LNCS 1254, pages 191–201. Springer, 1997.

[Tsa00] Yih-Kuen Tsay. Compositional verification in linear-time temporal logic. In *3rd International Conference on Foundations of Software Science and Computation Structures (FOSSACS)*, LNCS 1784, pages 344–358. Springer, 2000.

[Val98] Antti Valmari. The state explosion problem. In W. Reisig and G. Rozenberg, editors, *Lectures on Petri Nets I: Basic Models*, LNCS 1491, pages 429–528. Springer, 1998.

[Var95] Moshe Y. Vardi. On the complexity of modular model checking. In *10th IEEE Symposium on Logic in Computer Science (LICS)*, pages 101–111. IEEE Computer Society, 1995.

[VV01] Mahesh Viswanathan and Ramesh Viswanathan. Foundations for circular compositional reasoning. In *28th International Colloquium on Automata, Languages and Programming (ICALP)*, LNCS 2076, pages 835–847. Springer, 2001.

[VW86] Moshe Y. Vardi and Pierre Wolper. An automata-theoretic approach to automatic program verification. In *Proceedings of the Symposium on Logic in Computer Science (LICS)*, pages 332–344. IEEE Computer Society, 1986.

List of Symbols

Index