PhD-FSTM-2020-46
The Faculty of Science, Technology and Medicine

DISSERTATION

Defence held on 01/10/2020 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

Ali FARJAMI
Born on 23 February 1989 in Tehran (IRAN)

DISCURSIVE INPUT/OUTPUT LOGIC: DEONTIC MODALS, AND COMPUTATION

Dissertation defence committee

Dr. Leon van der Torre, dissertation supervisor
Professor, Université du Luxembourg

Dr. Pierre Kelsen, Chairman
Professor, Université du Luxembourg

Dr. Jan Broersen, Vice Chairman
Professor, Utrecht University

Dr. Dov Gabbay
Professor, King's College London (emeritus)

Dr. Christian Straßer
Professor, Ruhr-University Bochum

Discursive Input/Output Logic: Deontic Modals, and Computation

A thesis submitted in fulfilment of the requirements for the degree of Ph.D.

by

Ali Farjami

Department of Computer Science

University of Luxembourg

Abstract

The thesis investigates logical and computational aspects of normative reasoning using deontic logic and theorem provers. This interdisciplinary study draws inspiration from logic-based knowledge representation in Artificial Intelligence, deontic modality in Linguistics, and Philosophical Logic. The modal logic and norm-based paradigms in deontic logic investigate the logical relations among normative concepts such as obligation, permission, and prohibition. The thesis unifies these two paradigms by introducing an algebraic framework, called discursive input/output logic, in which deontic modals are evaluated with reference both to a set of possible worlds and a set of norms. The distinctive feature of the new framework is the non-adjunctive definition of input/output operations. Non-adjunctive logical systems are those where deriving the conjunctive formula $\varphi \wedge \psi$ from the set $\{\varphi, \psi\}$ fails. These systems are especially suited for modeling discursive reasoning. Moreover, the thesis presents a new compositional theory of conditional obligation and permission, which separates the contributions of "if" and "ought" ("may"). We propose to combine input/output logic as a logical theory about deontic modals with Hansson and Lewis's conditional theory simultaneously. In addition, we provide a dataset of semantic embeddings of deontic logics in Isabelle/HOL. The dataset can be used for ethical and legal reasoning tasks.

Acknowledgements

First of all, I would like to express my gratitude to my supervisor, Leon van der Torre. I have learned much from his comments on my talks and writing, clarification of conceptual ideas, and his academic life attitude. I am very grateful to Xavier Parent and Christoph Benzmüller. Thanks for the advice, encouragement, and criticism. I thank Jan Broersen for all the attention that he has paid to me during these years. I would like to thank Christian Straßer and Pierre Kelsen for their willingness to assess my dissertation. The technical reports of Dov Gabbay and Christian Straßer about my Ph.D. thesis gave me much food for thought and improving this thesis.

A special thanks to Dov Gabbay, Emil Weydert, and Shahid Rahman for all the interesting discussions, many of which influenced the ideas that lead to this thesis. A particular thanks to Majid Alizadeh, who, during my visit to the University of Tehran in the summer of 2019, dedicated much time and attention to my work and gave me essential advice. I surely need also to thank Seyed Mohammad Yarandi for our joint readings and critical discussions. A special thanks to Paul Meder for our joint work. I must thank all my colleagues in the ICR group: Alexander Steen, Réka Markovich, Tomer Libal, Giovanni Casini, Livio Robaldo, Jérémie Dauphin, Shohreh Haddadan, and David Streit, among others for our fruitful discussions and their help during my Ph.D. studies.

I want to thank my parents and my two brothers for their support, as well as my friends in Tehran (Iravani mosque) and Belval (omelet parties). Finally, I would like to thank Mahsa (and her family) for her love, support, and patience.

Ali Farjami
Grand Duchy of Luxembourg, 2020

Contents

List of Figures

To my wife, to my parents,
and to the memory of Seyed Aliasghar Mirpenhan

Chapter 1

Introduction: Deontic Logic, and Computation

The thesis introduces a new logical framework for reasoning about permission and obligation. It is a non-adjunctive variant of input/output logic. The new framework works for discursive normative reasoning and has strong connections to the classical semantics for deontic modals and conditionals. Moreover, the thesis uses artificial intelligence (AI) techniques, namely automated deduction, to the study of normative reasoning in deontic logic. The results may be used in AI, trustworthy and responsible AI, and in other domains where deontic logic is used. The aim of mechanizing deontic logic is to experiment, study, and change the existing deontic logics.

1.1 Normative Reasoning in Deontic Logic

Normative reasoning is mainly about "ought to be" and "ought to do" statements. We use these statements to *describe* or *prescribe* actions (or ideal actions) of an individual or a group of persons for ethical or legal tasks. Deontic logic investigates the logical relations among normative concepts such as obligation, permission, and prohibition [147]. We use the following symbols for representing conditional and unconditional obligations.

- $\bigcirc\varphi :=$ It ought to be the case that φ; φ ought to be done; It is obligatory that φ.
- $\bigcirc(\psi/\varphi) :=$ It ought to be the case that ψ, given φ; ψ ought to be done if φ is done; It is obligatory that ψ, given φ.

1.2 Automated Theorem Prover: A Reasoner Engine Tool

A deduction system can be characterized by a set of logical axioms schema and inference rules, and a formula deduction is a logical derivation. An automated deduction system is obtained by implementing a deduction system in a computer programming language. Automated theorem provers (ATP) are automated deduction systems used to prove or disprove a formula conjecture (deduction) in the deduction system. ATP systems are powerful computer programs for solving complex computational problems. Some ATP systems, *interactive theorem provers* (ITP), are not completely automated and need an expert guide for intermediate steps to solve the conjectures [137, 82]. Some ATP systems include *HOL*, *Isabelle*, *Coq*, and *Agda*.

1.3 Normative Reasoning and Logic Engineering

Artificial intelligence (AI) is concerned with studying intelligent behaviors. Symbolic logic is one of the most important bases of the mathematics of AI [96]. The logic-based AI methodology, so-called knowledge representation and reasoning (KR), is based on the interaction between a knowledge base and an inference mechanism. A knowledge-based agent stores sentences about the world in its knowledge-base and within using the inference mechanism decides by following new sentence derivation [185].

Deontic logic is expressive enough for representing ethical and legal reasoning tasks. Legal and ethical reasoning have special requirements due to their normative and conflict resolution roles. We have the family of traditional deontic logics, which includes standard deontic logic (SDL), a modal logic of type **KD**, and dyadic deontic logic (DDL) [12, 93]. On the other hand, we have so-called "norm-based" deontic logics [108]. The deontic operators are evaluated not with reference to a set of possible worlds but with reference to a set of norms. A particular framework that falls within this category is called input/output (I/O) logic [140].

Modal logic approach The classic semantics for deontic modality was developed as a branch of modal logic in variants by Danielsson [80], Hansson [109], Føllesdal, Hilpinen [86], van Fraassen [216, 217], and Lewis [131, 132], among others. However, its most developed formulation is the Kratzerian framework [128, 129]. For Kratzer, the semantics of deontic modals has two contextual components: a set of accessible worlds and an ordering of those worlds. In the Kratzerian framework, each contextual component is given as a set of propositions. Formally these are both functions, called *conversational backgrounds*, from evaluation worlds to sets of propositions. The *modal base* determines the set of accessible worlds and the *ordering source* induces the ordering on worlds [128, 218]. Conversational background functions uniform informational and motivational modalities, such as knowledge, beliefs, relevant facts, desires, and plans.

Norm-based approach Van Fraassen [217] and Makinson [138], among others, drew attention to a semantics, so-called norm-based [108], for obligations and permissions, where deontic operators are evaluated not with reference to a set of possible worlds but with reference to a set of norms. This set of norms cannot meaningfully be termed true or false. The logic developed by Makinson and van der Torre [140, 164] is known as input/output (I/O) logic. I/O logic is a fruitful framework for the theoretical study of deontic reasoning [141, 166] and has strong connections to nonmonotonic logic [141], the other main method for normative reasoning [113, 157]. More examples of norm-based logics include theory of reasons [114], which is based on Reiter's default logic, and logic for prioritized conditional imperatives [107].

Inference engine for family of logics Simple type theory developed by Church [75], aka classical higher-order logic (HOL), is an expressive language for representing mathematical structures. The syntax and semantics of HOL are well understood [26, 25]. It has roots in Begriffsschrift [89] by Frege and ramified theory of types [184] by Russell. Simple type theory can help to study and understand semantical issues. The so-called *shallow semantical embedding* approach is developed by Benzmüller [24] for translating (the semantics of) classical and non-classical logics into HOL. For example, by encoding Kripke style semantics (possible world semantics), many propositional and quantified modal logics can be embedded in Church simple type theory. Examples include propositional and quantified multimodal logics [40, 22], intuitionistic logics [39], epistemic and doxastic logics [21], access control logics [20]. Some advantages of this methodology are as follows:

- A crucial advantage of this embedding approach is that powerful proof assistance and automated theorem provers for HOL already exist. Isabelle/HOL [156] is one of the important automated proof tools for simple type theory. Isabelle/HOL is a useful tool for automated reasoning and studying computational aspects of logics.
- Classical higher-order logic is expressive enough, as unifying meta-logic, to model explicitly the syntax and semantics of varying other logics and flexibly combine them.

1.3.1 Research questions

Norms, semantics, and deontic modals The first main research question of this thesis is mainly about unifying the norm-based deontic logic, input/output logic, and the classic semantics for deontic modals[1] and integrating it with a conditional theory for resolving normative conflicts.

Norms, and modalities *How can we integrate the norm-based approach in the sense of Makinson [138] to the classic semantics in the sense of the Kratzerian framework [128]?* We introduce a semantics for deriving deontic modals based on bringing the core semantical elements of both approaches in a single unit. We use input/output logic for normative reasoning and the Kratzerian framework for representing different sets of information and motivation. The question here is relevant to the question that is addressed by Leon van der Torre and Jan Broersen [62] in *Ten Problems of Deontic Logic and Normative Reasoning in Computer Science*: "How do norms interact with informational modalities such as beliefs and knowledge, and motivational modalities such as intentions and desires?". More generally, each modal logic and norm-based approach has its advantages. An advantage of the modal logic approach is the capability to extend with other modalities such as epistemic or temporal operators. The norm-based approach's advantages include the ability to explicitly represent normative codes such as legal systems and using non-monotonic logic techniques of common sense reasoning. Unifying these two approaches will provide us with a framework with all of these advantages simultaneously.

This research question is based on three more concrete questions or requirements:

- *How can we use an algebraic setting such as Boolean algebras instead of a logical setting for building input/output logic on top of it?* This requirement is important for making connections to algebraic models of modal logic approach such as Boolean algebras. Gabbay, Parent, and van der Torre [94] gave a proposal for building I/O framework on top of lattices. They have result only for the simple-minded output operation. We show that for an input set A by using *upward-closed set of A* operator instead of *upward-closed set of the infimum of A* [94], we can build many new and old derivation systems over Boolean algebras, Heyting algebras, and generally any abstract logic. We use Stone's representation theorem for Boolean algebras

[1] "Deontic modality is a kind of modality which has to do with what is necessary or possible according to various rules, such as the norms of morality, the principles of practical rationality or the laws of some country." (Routledge Encyclopedia of Philosophy)

for integrating input/output logic with possible world semantics. Another possible worlds semantics of I/O logic is studied by Bochman [47] for causal reasoning. It has no direct connection to the operational semantics (see Subsection 2.4, [164]). The algebrization of the I/O framework shows more similarity with the theory of joining-systems [136] that is an algebraic approach for study normative-systems over Boolean algebras. We can say that norms in the I/O framework play the same role of joining in the theory of Lindahl and Odelstal [136, 200]. Sun [200] built Boolean joining systems that characterize I/O logic in a sense that a norm is derivable from a set of norms if and only if it is in the set of norms algebraically generated in the Lindenbaum-Tarski algebra for propositional logic. The work of Sun [200], similar to the Bochman approach [47], has no direct connection to output operations. Here, we build algebraic I/O operations directly over Boolean algebras and, more generally, abstract logics.

- *How can we introduce two groups of I/O operations similar to syntactical characterization of box and diamond in modal logic?* This requirement is important for building primitive deontic modalities similar to modal logic. We need to define two groups of operations similar to the possible world semantics characterization of box and diamond, where box is closed under AND, $((\Box\varphi\wedge\Box\psi) \to \Box(\varphi\wedge\psi))$, and diamond not.

  – *Derivations systems that do not admit AND rule*: In the main literature of input/output logic developed by Makinson and van der Torre [140], Parent, Gabbay, and van der Torre [163], Parent and van der Torre [165, 167, 169], and Stolpe [192, 193, 196], at least one form of AND inference rule is present (see the related table in Subsection 2.2.2). Sun [199] analyzed norms derivations rules of input/output logic in isolation. Still, it is not clear how we can combine them and build new logical systems, specifically systems that do not admit the rule of AND. We show how we can remove AND rule from the proof system and build new I/O operations to produce permissible propositions. Comparing to minimal deontic logics [71, 102], and similar approaches, such as Ciabattoni et al. [76], that do not have deontic aggregation principles, our approach validates deontic and factual detachment.

  – *Derivations systems that admit AND rule*: According to the reversibility of inference rules in the I/O proof systems, we show that how it is possible to add AND and other rules, required for obligation [140], to the proof systems, and find I/O operations for them.

There are other abstract approaches: I/O operations over semigroups [208], which does not admit AND; and a detachment mechanism over an arbitrary set [3], which admits a kind of AND, cumulative aggregation. However, it is not clear how we can use these approaches for logical purposes.

- *How can we integrate conversational backgrounds, from the Kratzerian framework, into input/output logic framework to build a more fruitful unified semantics for deontic modals?* Horty [112] raised the issue of unification for deontic logic. We are still missing a semantics based on both the modal logic and the norm-based approaches. We define a semantics that unifies both approaches. We employ the contextual components, the *modal base* and *ordering source* functions, from the Kratzerian framework [129] and the *detachment* approach [164] from I/O semantical framework, instead of quantification, for deriving deontic modals. As an advantage of the detachment approach, we can characterize derivation systems that do not admit, for example, weakening of the output (WO) or strengthening of the input (SI). There are other frameworks, such as adaptive logic [197, 198], that combine norm-based and modal logic approaches. The novelty of our approach is *semantical unification.* The unification is based on bringing the core semantical elements of both approaches into a single unit.

Norms, and conditionals *How can we integrate input/output logic with Hansson and Lewis's conditional theory for building a new compositional theory about conditional deontic modals?* We can consider two motivations for this research question:

- *How can we use Hansson and Lewis's conditional theory within input/output logic for resolving contrary-to-duty problems?* It is open whether by using a conditional theory, obtained by a preference relation, inside input/output logic, it can support contrary-to-duty scenarios. We combine input/output logic with Hansson and Lewis's conditional logic [109, 132]. The new framework can be used for resolving contrary-to-duty problems. Constrained I/O logic [141] was introduced for reasoning about contrary-to-duty problems. There are syntactical ([141], Section 6) and proof theoretical ([198], Section 3) characterizations for constrained I/O logic. Here, constraints are preferences. In this sense, we present a semantical characterization for constrained I/O logic.

- *How can we use a suitable non-monotonic defeat mechanism within Hansson and Lewis's conditional theory in the face of dilemma problems?* Hansson and Lewis's conditional theory [109, 132] is too weak to represent inconsistency in deontic dilemmas [174, 213]. We can combine input/output logic as a non-monotonic defeat mechanism with Hansson and Lewis's conditional theory [109, 132] to detect these dilemmas. Prohairetic deontic logic (PDL) was investigated by van der Torre and Tan [213] to formalize contrary-to-duty conditionals [109, 132] and detect deontic dilemmas. It is based on combining two dyadic deontic operators for each task. Their formalization is in terms of monadic deontic logic and a deontic betterness relation. They take the axiomatization of the underlying modal logic, which provides a uniform semantically framework for the both operators [203, 214, 213]. Our axiomatization comes directly from the chosen conditional logic and the derivation system of input/output logic. It is flexible to add or remove inference rules such as weakening of the output (WO), reasoning by cases (OR), cumulative transitivity (CT), and more interestingly rules with consistency check. In this sense, our approach provides a uniform syntactical way to combine Hansson and Lewis's conditional theory [109, 132] with a non-monotonic mechanism. A problem in most solutions for detecting dilemmas is that the set of formulas $N = \{\bigcirc(\neg\psi/\varphi_1), \bigcirc(\psi/\varphi_2)\}$ is inconsistent or $\varphi_1 \wedge \varphi_2$ is impossible [213]. In our setting the set of formulas N is not necessary inconsistent, given $\{\varphi_1, \varphi_2\}$ consistent. We detect dilemmas by using detachment in the output operations.

Deontic modals, semantics, and computation The second main research question of this thesis is about automating some popular deontic logics in higher-order theorem provers. As far as we know, deontic logic is not studied with computational tools very well, and the implementation of deontic logic in the computational tools is not apparent. We consider the following requirements for this purpose:

Faithful Embedding of some deontic logics in HOL The first and most important requirement of our second research question is providing a (faithful) embedding of some well-known deontic logics in HOL as the target logic.

- The deontic logic by Carmo and Jones [66] that is addressed in the handbook of philosophical logic and has a complex neighborhood semantics is faithfully translated into HOL.[2] It is reported as a joint work with Benzmüller and Parent in the

[2] The faithfulness is jointly proven by Benzmüller, Farjami, and Parent. The implementation of the logical system in Isabelle/HOL is given by Benzmüller.

conference paper [28] as well the slightly reworked version in Chapter 7 with some experiments within the Kratzerian framework.

- The dyadic deontic logic introduced by Hansson [109] and developed by Åqvist [12] with a preference semantics by Parent [161], which is well-known as one of the main semantics for deontic modals, is faithfully embedded in HOL.[3] It has published as a joint work with Benzmüller and Parent in the journal paper [31] as well the slightly reworked version in Chapter 6 with a characterization of the dyadic deontic system within the Kratzerian framework.

- Input/output logic [140] faithful embedding, as a norm-based semantics embedding of deontic logic, is studied in Chapter 5. An indirect approach is devised to embed two I/O operations in modal logic and consequently into HOL,[4] which is published as a journal paper jointly with Benzmüller, Meder, and Parent [28]. As one advantage of the building I/O framework over Boolean algebras, which is addressed as our first research question, is a direct embedding of the proposed I/O framework in HOL, studied in Chapter 5.

Implementing the embedded logics in a well-known higher-order theorem prover Our second requirement is encoding the logical embeddings in Isabelle/HOL, which turns this system into a proof assistant for deontic logic reasoning. The experiments with this environment should provide evidence that these logical *implementations* fruitfully enable interactive and automated reasoning at the meta-level and the object-level.

1.3.2 Methodology

Normative reasoning

Norms, and modalities To achieve a more uniform semantics [112, 92, 166] for deontic modals, we build I/O operations on top of Boolean algebras for deriving permissions and obligations. The approach is close to the work is done by Gabbay, Parent, and van der Torre: a geometrical view of I/O logic [94]. For defining the I/O framework over an algebraic setting, they use the algebraic counterpart, *upward-closed set of the infimum*

[3] The faithfulness is jointly proven by Farjami and Benzmüller. The implementation of the logical system in Isabelle/HOL is given by Benzmüller and Farjami.

[4] The faithfulness is proven by Farjami. The GDPR and Moral Luck formalization and implementation were done by Farjami and Meder [84, 28].

of A, for *the propositional logic consequence relation* ("$Cn(A)$"), within lattices. They have characterized only the simple-minded output operation.[5] We show that by choosing the "Up" operator,[6] *upward-closed set*, as the algebraic counterpart of the "Cn" operator and by using the reversibility of inference rules in the I/O proof system, we can characterize all the previously studied I/O systems and find many more new logical systems. This suggested framework has a significant difference from other types of input/output logics. In contrast to the earlier input/output logics, we define non-adjunctive input/output operations. Non-adjunctive logical systems are those where deriving the conjunctive formula $\varphi \wedge \psi$ from the set $\{\varphi, \psi\}$ fails [77, 78]. These systems are especially suited for modeling discursive reasoning. This is why we call the proposed framework *discursive input/output logic*. We build two groups of I/O operations for deriving permissions and obligations over Boolean algebras. The main difference between the two operations is similar to the possible world semantics characterization of box and diamond, where box is closed under AND, $((\Box\varphi \wedge \Box\psi) \rightarrow \Box(\varphi \wedge \psi))$, and diamond not. Moreover, we use Stone's representation theorem for Boolean algebras for integrating input/output logic with possible world semantics.

Norms, and conditionals We propose to combine input/output logic [140] as a logical theory about deontic modals with Hansson and Lewis's conditional logic [109, 132] simultaneously. We use the valuation functions from a set of propositional symbols into the class of Boolean algebras for defining our conditional theory. It is a well-known fact that Boolean algebras are the algebraic models of (classical) propositional logic. In this thesis, we have developed I/O mechanism over every Boolean algebra. So the natural question would be extending algebraic models of propositional logic along with I/O operations over Boolean algebras. We show that the extension of propositional logic with a set of conditional norms is sound and complete respect to the class of Boolean algebras that the corresponding I/O operation holds through all of them. In fact, the valuation functions play the role of possible worlds, and the order over them supports the theory of conditionals. The proposed I/O framework is expressive enough to represent preferential conditionals.

Inference engine To achieve ambitious goals, such as designing ethical and legal machines and responsible systems, Benzmüller, Parent and van der Torre [37] introduced

[5] In their approach, the background algebra should be complete and compact.

[6] Thanks Majid Alizadeh for this suggestion.

LogiKEy methodology based on the semantical embedding of deontic logics. The engineering methodology addresses three layers: *Logics and logic combinations* (L1), *Ethico-legal domain theories* (L2) and *Applications* (L3). The work was done (in Chapters 5, 6 and 7) in this thesis is partly match for the Layer 1.[7]

Our methodology for engineering family of deontic logics is the *shallow semantical embedding* (SSE) approach developed by Benzmüller [24], which is based on three main ingredients:

Semantical embedding approach First, we can semantically characterize the source logic using set theory. So for each logical constant, we have a corresponding equation. Second, since HOL is expressive enough, we can represent the equations that characterize semantics of source logic and translate the syntax of source logic employing these semantical representations into HOL as the target logic.

Theorem prover tool support The semantical embedding of a logic in HOL could be implemented in higher-order theorem provers. In this thesis, we use the powerful and up-to-date higher-order theorem prover Isabelle/HOL that is supported by the University of Cambridge and Technical University of Munich. Practically, the equational theory, which semantically embeds source logics in the target logic HOL, could be encoded in Isabelle/HOL.

Universal logical reasoning approach HOL is an expressive logical language. HOL as unifying meta-logic encodes (combinations of) a wide range of classical and non-classical logics [24]. This quality gives us the chance to compare and combine different deontic logics or other intentional logics such as epistemic, temporal, and action logic in HOL.

1.4 Success Criteria

Normative reasoning We have provided a couple of soundness and completeness results for building I/O frameworks on top of Boolean algebras. One immediate benefit of moving to an algebraic setting for building I/O operations is the ability to construct I/O

[7] Our study in this thesis is simpler than the steps mentioned in the Layer 1 [37]. We do not discuss the logic combination in this thesis. Only some well-known logics, with specific semantics, in the literature are chosen, the semantical embeddings in HOL are devised, the embeddings are implemented in Isabelle/HOL and tested by some deontic puzzles mainly contrary-to-duty examples.

logic over any consequence relation. Moreover, we have introduced a new semantics by integrating norm-based and classical semantics for deontic modals to unify both approaches. Also, we have combined conditional logic tradition and norm-based deontic logic for normative reasoning. The results wish to bring different approaches for deontic logic under the same umbrella. For deriving (monadic) deontic modals the fragment is input/output logic based on the interaction between normative reasoning and informational and motivational modalities. We use the Kratzerian framework for representing different sets of information and motivation. For dyadic obligations and permissions, we add another fragment besides input/output logic based on the basic idea in classical semantics (or the Kratzerian framework): a preference ordering on possible worlds, and what ought to be the case, is determined by what is the case in all the best of the accessible worlds. We introduce a compositional theory of conditional obligation and permission. The framework is capable of solving deontic paradoxes.

Inference engine We have shown the faithfulness of some logical embeddings in higher-order logics. We encoded the logical translations in Isabelle/HOL, which turns this system into a proof assistant for deontic logic reasoning. The experiments with this environment provide evidence that these logical implementations fruitfully enables interactive and automated reasoning at the meta-level and the object-level, which developed to LogiKEy framework. This thesis contribution to LogiKEy is several semantical embeddings, which are reported in [29, 31, 28] and in Chapters 5, 6, and 7 as well as in the Data in Brief article [27] accompanying Benzmüller et al. Artificial Intelligence Journal paper [37].

1.5 Interdisciplinary Aspects

Deontic logic is useful in many areas. For example, philosophers are interested in the semantics of norms and moral action theory, linguists to deontic paradoxes, lawyers to norm compliance [212]. Moreover, the study of norms is related to a bunch of computer science communities. In 1991 Meijer and Wieringa founded Deontic Logic in Computer Science (DEON) conferences for applying deontic logics in computer problems. Some of the computer science communities interested in normative concepts include normative multi-agent systems [73, 170], security [53], rational architecture [60, 61, 50], and artificial intelligence [37]. For example, one of the main aims of artificial intelligence is to provide a formal model of ethical agents. Machine ethics is a subfield of artificial intelligence focusing on the task of ensuring the ethical, legal, and social behavior of artificial agents. Engineering of deontic logic, which is one of our goals in this thesis, is a potential proposal for designing these machines [37]. Machine ethics and machine law are needed for developing trustworthy and responsible AI. We can study how an agent should reason to behave responsibly in a normative (multi-agent) environment through deontic logic. In Section 8.6, we discuss a couple of related interdisciplinary domains, how and why this thesis can contribute to them.

1.6 Thesis Structure

We present the formal system and our methodology requirements in Chapter 2. In Chapter 3, we introduce discursive input/output logic and its integration into the Kratzerain framework. Chapter 4 presents our new compositional conditional theory. Chapter 5 is dedicated to engineering input/output logic. Chapter 6 presents the semantical embedding of Åqvist dyadic deontic logic **E** in HOL and study some meta-theoretical properties of the system. Chapter 7 discusses the implementation and automation of dyadic deontic logic proposed by Carmo and Jones in Isabelle/HOL. Finally, in Chapter 8, we give a summary and briefly discuss further possible work.

Chapter 2

Formal Framework, Methodology, and Tool Engine

The formal framework in this thesis is logic. More specifically, we use a logical framework, namely input/output logic, for reasoning about permissions and obligations. We also study normative reasoning within deontic logic by using automated deduction systems for ethical and legal reasoning tasks. We do this indirectly. We translate some deontic logics into higher-order logic. Higher-order logic (HOL) is our computational formal framework, and translating a target logic into HOL is by using our methodology, shallow semantical embedding. There are robust automated deduction systems for HOL, which we can automate our deontic translations into HOL with them. So higher-order theorem provers are our tool engines for automating normative reasoning. In this chapter, we give a summary about deontic logic and higher-order logic (as formal frameworks), shallow semantical embedding (as logical translation methodology), and theorem provers (as tool engines).

2.1 Formal Framework: Logic

Logic has applications in mathematics, computer science, artificial intelligence, philosophy, and many other disciplines. We briefly mention three foremost of them, mainly we discuss using logic in the area of artificial intelligence.

Logic in mathematics Whitehead and Russell [225] introduced logic as a foundation for mathematics. Hilbert's Program aims for a formalization of all of mathematics in a consistent axiomatic form [226]. Gödel second theorem was a negative answer to Hilbert's idea. However, developing theorem provers for deriving true statements for a mathematical theory is an active domain.

Logic in computer science This area is based on the interaction between logic and computing, which covers many aspects of information technology, from software engineering and hardware to programming and artificial intelligence. Natural language processing, program control specification, artificial intelligence, logic programming, imperative vs. declarative languages, database theory, and complexity theory are some areas that different domains of logic such as temporal, modal, non-monotonic, and intuitionist logic have contributions.

Logic for artificial intelligence: Knowledge representation Artificial intelligence (AI) is concerned with studying intelligent behaviors. Symbolic logic is one of the most important bases of the mathematics of AI [96]. Logic develops concepts and formal systems for intelligent behavior. The logic-based AI methodology, called knowledge representation and reasoning (KR), is based on the interaction between a knowledge base and an inference mechanism. A knowledge-based agent stores sentences about the world in its knowledge base and within the inference mechanism decides by following new sentence derivation [185].

2.2 Deontic Logic: A Framework for Normative Reasoning

Deontic logic is mainly about normative concepts. The development of modern deontic logic is based on the following two main approaches.

2.2.1 Modal logic approach

In the modal-based approach [43] for deontic reasoning, deontic statements are evaluated with reference to a set of possible worlds.

Standard deontic logic (SDL)

The starting point of deontic logic is based on the seminal paper of G. H. von Wright's classic paper "Deontic Logic" [220]. The significant analogy between obligation and permission with modal operators is the core of von Wright's proposal.

Necessity $\Box$	**Obligation** $\bigcirc$
Possibility $\Diamond$	**Permission** P

The standard system of deontic logic is known as **KD** in the literature [86]. It is the extension of modal logic **K** with the axiom **D**:

$$\Box\varphi \to \Diamond\varphi$$

The language of **K** is obtained by supplementing the language of propositional logic (PL) with a modal operator $\Box$. It is generated as follows:

$$\varphi ::= p|\neg\varphi|\varphi \vee \varphi|\Box\varphi$$

where p denotes an atomic formula. Other logical connectives such as $\wedge$, $\to$ and $\Diamond$, are defined in the usual way. The axioms of system **K** consist of those of PL plus $\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$, called axiom K. The rules of **K** are *Modus ponens* (from φ and $\varphi \to \psi$ infer ψ) and *Necessitation* (from φ infer $\Box\varphi$).

A Kripke model for **K** is a triple $M = \langle W, R, V \rangle$, where W is a non-empty set of possible worlds, R is a binary relation on W, called accessibility relation, and V is a function assigning a set of worlds to each atomic formula, that is, $V(p) \subseteq W$.

Truth of a formula φ in a model $M = \langle W, R, V \rangle$ and a world $s \in W$ is written as $M, s \models \varphi$. We define $V(\varphi) = \{s \in W | M, s \models \varphi\}$. The relation $\models$ is defined as follows:

$$\begin{array}{lll}
M, s \models\ p & \text{if and only if} & s \in V(p) \\
M, s \models\ \neg\varphi & \text{if and only if} & M, s \not\models \varphi \,\text{(that is, not } M, s \models \varphi) \\
M, s \models\ \varphi \vee \psi & \text{if and only if} & M, s \models \varphi \text{ or } M, s \models \psi \\
M, s \models\ \Box\varphi & \text{if and only if} & \text{for every } t \in W\text{such that } sRt,\ M, t \models \varphi
\end{array}$$

As usual, a modal formula φ is *true in a Kripke model* $M = \langle W, R, V \rangle$, i.e., $M \models \varphi$, if and only if for all worlds $s \in W$, we have $M, s \models \varphi$. A formula φ is *valid in a class* $\mathcal{C}$ *of Kripke models*, denoted as $\models_{\mathcal{C}} \varphi$, if and only if it is true in every model in class $\mathcal{C}$.

System **K** is determined by (i.e., is sound and complete with respect to) the class of all Kripke models. System **KD** is obtained from system **K** by adding the schema $\mathbf{D} : \Box\varphi \rightarrow \Diamond\varphi$ as an axiom. System **KD** is determined by the class of all Kripke models in which R is serial.[1] We denote the class of all Kripke models and the class of Kripke models where R is serial as $\mathcal{C}_K$ and $\mathcal{C}_{KD}$, respectively. The following table demonstrates the proof theory of traditional monadic deontic logic type of **KD**.

Axiom schema	
All instance of tautologies	(PL)
$\bigcirc(\varphi \rightarrow \psi) \rightarrow (\bigcirc\varphi \rightarrow \bigcirc\psi)$	(K)
$\bigcirc\varphi \rightarrow P\varphi$	(D)
Rules	
$\dfrac{\varphi}{\bigcirc\varphi}$	$\bigcirc$- necessitation
$\dfrac{\varphi, \varphi \rightarrow \psi}{\psi}$	Modus pones (MP)

Dyadic deontic logic (DDL)

A landmark and historically important family of modal-based deontic logic so-called dyadic deontic logics has been developed mainly by Rescher [178], von Wright [223], Danielsson [80], Hansson [109], van Fraassen [216, 217], Lewis [132], Spohn [188], Åqvist [9, 10, 12], Goble [100, 101], and Parent [158, 159, 161]. The framework was motivated by the well-known paradoxes of *contrary-to-duty* (CTD) reasoning like Chisholm [72]'s paradox. They come with a preference semantics, in which a binary preference relation ranks the possible words in terms of betterness [161].

A preference model is a structure $M = \langle W, \succeq, V \rangle$ where

- W is a non-empty set of items called possible worlds;
- $\succeq \subseteq W \times W$ (intuitively, $\succeq$ is a betterness or comparative goodness relation);
- V is a function assigning to each atomic sentence a set of worlds (i.e $V(p) \subseteq W$).

(Satisfaction) $M, s \models \ \bigcirc(\psi/\varphi)$ if and only if $Best(V(\varphi)) \subseteq V(\psi)$

[1] Other axiom schemas that can be added to **K** are $\mathrm{T} : \Box\varphi \rightarrow \varphi$, $4 : \Box\varphi \rightarrow \Box\Box\varphi$ and $5 : \Diamond\varphi \rightarrow \Box\Diamond\varphi$. For instance, **K45** is an extension of **K** obtained by adding 4 and 5 as axioms. The schemas T, 4 and 5 are valid if R is *reflexive*, *transitive* and *euclidean*, respectively. We use these systems in Chapter 5.

2.2.2 Norm-based approach

Van Fraassen [217] and Makinson [138], among others, drew attention to a semantics for obligations and permissions, based on the distinction between norms and propositions about norms. SDL and DDL deontic logics do not explicitly represent a distinction between norms and deontic modals. The proposed norm-based semantics analysis obligations and permission operators not with reference to a set of possible worlds but with reference to a set of norms. The semantics is motivated by Jørgensen's dilemma. This is a puzzling situation about truth and normative language.

> "So we have the following puzzle: According to a generally accepted definition of logical inference only sentences which are capable of being true or false can function as premisses or conclusions in an inference; nevertheless it seems evident that a conclusion in the imperative mood may be drawn from two premises one of which or both of which are in the imperative mood. How is this puzzle to be dealt with?" (Jørgensen [121])

Norm-based paradigm is a recent robust program to provide a logical framework for norms or imperatives as non-truth-evaluable items [93]. Examples include input/output (I/O) logic [140], Horty's theory of reasons [114], and Hansen's logic of imperatives. In this thesis, we focus on input/output framework.

Unconstrained input/output logic

Input/output logic was initially introduced by Makinson and van der Torre [140]. There are various I/O operations. The general theory of input/output logic is about operations resembling inference, where inputs need not be included among outputs, and outputs need not be reusable as inputs [140]. The input/output logic is inspired by a view of logic as "secretarial assistant" rather than logic as an "inference motor".

> "logic is often seen as an 'inference motor', with premises as inputs and conclusions as outputs. But it may also be seen in another role, as 'secretarial assistant' to some other, perhaps non-logical, transformation engine. From this point of view, the task of logic is one of preparing inputs before they go into the machine, unpacking outputs as they emerge and, less obviously,

> coordinating the two. The process as a whole is one of ‘logically assisted transformation’, and is an inference only when the central transformation is so.” (Makinson and van der Torre [140])

Makinson and van der Torre [140] exemplified two main kinds of transformation engines, or black boxes, which could assisted by input/output logic .

> “The box may stop some inputs, while letting others through, perhaps in modified form. [...] Inputs might be facts about the performance of the stock-market today, and outputs an analyst’s commentary; or facts about your date and place of birth, with output your horoscope readings. In these examples, the outputs express some kind of belief or expectation.
>
> Again, inputs may be conditions, with outputs expressing what is deemed desirable in those conditions. The desiderata may be obligations of a normative system, ideals, goals, intentions or preferences. In general a fact entertained as a condition may itself be far from desirable, so that inputs are not always outputs; and as is widely recognised, contraposition is inappropriate for conditional goals.”(Makinson and van der Torre [140])

Syntax

$N \subseteq \mathcal{L} \times \mathcal{L}$ is called a normative system, with $\mathcal{L}$ representing the set of all the formulas of propositional logic. A pair $(a, x) \in N$ is referred to as a conditional norm or obligation, where a and x are formulas of propositional logic. The pair (a, x) is read as “given a, it is obligatory that x”. a is called the body and represents some situation or condition, whereas x is called the head and represents what is obligatory or desirable in that situation.

Semantics

For a set of formulas A, we define $N(A) = \{x \mid (a, x) \in N \text{ for some } a \in A\}$ and $Cn(A) = \{x \mid A \vdash x\}$ with $\vdash$ denoting the classical consequence relation. A set of formulas V is *maximal consistent* if it is consistent, and no proper extension of V is consistent. A set of formulas V is said to be *complete* if it is either *maximal consistent* or equal to $\mathcal{L}$.

Definition 2.1 (Output operation)**.** Given a set of conditional norms N and an input set A of propositional formulas,

- Simple-Minded Output:

$$out_1(N, A) = Cn(N(Cn(A)))$$

- Basic Output:

$$out_2(N, A) = \bigcap\{Cn(N(V)) \mid A \subseteq V, V \text{ complete}\}$$

- Simple-Minded Reusable Output:

$$out_3(N, A) = \bigcap\{Cn(N(B)) \mid A \subseteq B = Cn(B) \supseteq N(V)\}$$

- Basic Reusable Output:

$$out_4(N, A) = \bigcap\{Cn(N(V)) \mid A \subseteq V \supseteq N(V), V\text{complete}\}$$

Proof system

The proof system of an I/O logic is specified via a number of derivation rules acting on pairs (a, x) of formulas. Given a set N of pairs, we write $(a, x) \in deriv_i(N)$ to say that (a, x) can be derived from N using those rules.

- $(\top)$ Tautology: infer every $(\top, \top)$
- (SI) Strengthening of the input: from (a, x) and $\vdash b \to a$, infer (b, x)
- (WO) Weakening of the output: from (a, x) and $\vdash x \to y$, infer (a, y)
- (AND) Conjunction of the output: from (a, x) and (a, y), infer $(a, x \wedge y)$
- (OR) Disjunction of the input:[2] from (a, x) and (b, x), infer $(a \vee b, x)$
- (CT) Cumulative transitivity: from (a, x) and $(a \wedge x, y)$, infer (a, y)

Each output is syntactically characterized by $deriv_i(N)$ that is closed under rules SI, WO, AND, OR and CT as follows:

[2] Sometimes we call this rule: *Reasoning by cases.*

$derive_i(N)$	Rules
$derive_1(N)$	$\{\top$, SI, WO, AND$\}$
$derive_2(N)$	$\{\top$, SI, WO, OR, AND$\}$
$derive_3(N)$	$\{\top$, SI, WO, CT, AND$\}$
$derive_4(N)$	$\{\top$, SI, WO, OR, CT, AND$\}$

Input/output logic is flexible enough to include other conditional rules as well. Following table [169] has summarized the main recognized I/O systems in the literature "+" denotes presence of a rule and "−" its absence.

EQO	SI	WO	R-AND	R-AND'	AND	OR	R-ACT	ACT	MCT	CT	ID	References
-	+	-	-	-	-	-	-	-	-	-	-	[199]
-	-	+	-	-	-	-	-	-	-	-	-	[199]
-	-	-	-	-	+	-	-	-	-	-	-	[199]
-	-	-	-	-	-	+	-	-	-	-	-	[199]
-	-	-	-	-	-	-	-	-	-	+	-	[199]
+	+	-	+	-	-	-	-	-	-	-	-	[167]
+	+	-	+	-	-	-	+	-	-	-	-	[169]
+	+	-	+	+	-	-	-	-	-	-	-	[167]
+	+	-	+	+	+	-	-	-	-	-	-	[192, 193, 165]
+	+	-	+	+	+	-	+	+	-	-	-	[165]
+	+	-	+	+	+	-	+	+	+	+	-	[192, 193]
+	+	-	+	+	+	+	-	-	-	-	-	[165]
+	+	+	+	+	+	-	-	-	-	-	-	[140]
+	+	+	+	+	+	-	-	-	-	-	+	[140]
+	+	+	+	+	+	-	+	+	+	+	-	[140, 163]
+	+	+	+	+	+	-	+	+	+	+	+	[140]
+	+	+	+	+	+	+	-	-	-	-	-	[140, 163, 196]
+	+	+	+	+	+	+	-	-	-	-	+	[140]
+	+	+	+	+	+	+	+	+	+	+	-	[140]
+	+	+	+	+	+	+	+	+	+	+	+	[140]

- (EQO) Equivalence of the output: from (a, x), $x \vdash y$ and $x \vdash y$ infer (a, y)
- (R-AND) Restricted AND: from (a, x) and (a, y) and $a \wedge x \wedge y$ consistent, infer $(a, x \wedge y)$
- (R-AND') from (a, x), (a, y), $a \wedge x$ consistent and $a \wedge y$ consistent, infer $(a, x \wedge y)$
- (R-ACT) restricted ACT': from (a, x), $(a \wedge x, y)$ and $a \wedge x \wedge y$ consistent, infer $(a, x \wedge y)$
- (ACT) Aggregative CT: from (a, x) and $(a \wedge x, y)$, infer $(a, x \wedge y)$
- (MCT) Mediated CT: from $(a, x^{'})$, $(a \wedge x, y)$, $x^{'} \vdash x$, infer (a, y)
- (ID) Identity: infer (a, a)

2.2.3 Benchmark problems

The history of deontic logic evaluated through some common sense problems. We categorize these problems into four main categories: Martial implication, Contrary-to-duty, Exception, and Dilemma problems. In deontic logic literature, these problems called a paradox, since they are contrary to our intuition. For more details see [93, 166].

Martial implication problems: Ross paradox

The meaning of implication in monadic deontic logic (SDL) is the same as the material implication of propositional logic. Consequently, the paradoxes of material implications could make arise in deontic logic. Ross's paradox [182] is one of these paradoxes that is based on the monotonicity of material implication.

1. You should mail the letter. $\bigcirc m$
2. If you should mail the letter, then you should mail or burn the letter. $\bigcirc(m \vee b)$

This problem is rooted in the rule schemata (if $\vdash \varphi \rightarrow \psi$ then $\vdash \bigcirc\varphi \rightarrow \bigcirc\psi$) which holds in SDL. In this case, from $\vdash m \rightarrow m \vee b$ we have $\vdash \bigcirc m \rightarrow \bigcirc(m \vee b)$. It is odd to think that by brunning the letter, which is forbidden, I fulfilled my obligation to mail or burn it. The Good Samaritan paradox [175] is another problematic scenario in this category.

Deontic conditionals: Chisholm's contrary-to-duty paradox

Contrary-to-duty (or CTD for short) has been an important benchmark problem of studying deontic logic. This structure appears naturally in legal and ethical scenarios. Formalizing CTD scenarios in deontic logic was an important step and initially started by Chisholm [72].

1. It ought to be that Jones goes to assist his neighbors. $\bigcirc g$
2. It ought to be that if Jones goes, then he tells them he is coming.
3. If Jones does not go, then he ought not tell them he is coming.
4. Jones does not go. $\neg g$

For instance, in SDL, there are two possibilities to interpret sentences number 2 and 3: $\{\bigcirc(g \rightarrow t), g \rightarrow \bigcirc t\}$ and $\{\bigcirc(\neg g \rightarrow \neg t), \neg g \rightarrow \bigcirc \neg t\}$. Any four ways to interpret 1-4 yields a contradiction, or we lose logical independence. For example, in the most natural interpretation set $\{\bigcirc g, \bigcirc(g \rightarrow t), \neg g \rightarrow \bigcirc \neg t, \neg g \rightarrow \bigcirc \neg t\}$ the results of $\bigcirc g$ and $\bigcirc(g \rightarrow t)$, known as *deontic detachment*, is $\bigcirc t$ and the result of $\neg g \rightarrow \bigcirc \neg t$ and $\neg g$, known as *factual detachment*, is $\bigcirc \neg t$ which is inconsistent with $\bigcirc t$. The same problem happens in the "Gentle Murder paradox" [88].

1. It is obligatory that Smith not kill Jones.
2. If Smith does kill Jones, then it is obligatory that Smith kill him gently.
3. Smith does kill Jones.
4. That Smith kills Jones gently implies that Smith does kill Jones.

These examples illustrate that combining factual with deontic detachment besides deriving unconditional obligations is problematic [166].

Exception problems: Cottage regulations

Prakken and Sergot [174] based on cottage regulations argued that temporal and action logic are not capable of representing CTD problems. Cottage regulations light on the expressive power of deontic logic for defeasible reasoning also.

1. There must be no fence. $\bigcirc(\neg f/\top)$
2. If there is a fence, then it must be a white fence. $\bigcirc(w \wedge f/f)$
3. If there is a dog, then there must be a white fence.[3] $\bigcirc(f \wedge w/d)$

In the case that there is a fence without dog (f), the obligation $\bigcirc(\neg f/\top)$ is violated. For the case that there is a dog and fence $(f \wedge d)$, the obligation is overridden. The obligation $\bigcirc(\psi/\varphi)$ is overridden by obligation $\bigcirc(\psi_1/\varphi_1)$ if ψ and ψ_1 are inconsistent and φ_1 is more specific than φ. This example illustrates the distinction between contrary-to-duty and defeasible statements based on exceptional situations [166].

[3] It is "If the cottage is by the sea, then there may be a fence." in the original text [174]

Dilemma problems: Möbius strip

The requirement of avoiding conflict first is addressed by van Wright [220] as the axiom $\neg(\bigcirc\varphi \wedge \bigcirc\neg\varphi)$ in his system. Resolving norm conflicts [2] is an ongoing research domain employing modern tools such as argumentation theory and default logic besides deontic logic. Möbius strip represents how norm conflicts [138] and deontic detachment leads to new challenges.

1. ψ is obligatory given φ.
2. χ is obligatory given ψ.
3. $\neg\varphi$ is obligatory given χ.

In the case that φ is true, inspired by the maxi choice in AGM theory of change, there are three possibilities: both ψ and χ are obligatory, only ψ is obligatory, neither of ψ and χ is obligatory. Möbius strip example illustrates the dilemma problem among these three alternatives [166].

2.3 Deontic Logic for Knowledge Representation

Deontic logic is an expressive language for knowledge representation in ethical and legal scenarios. Legal and ethical reasoning have special requirements due to their normative and conflict resolution roles. The requirement of deontic logic first is addressed by Jones and Sergot [117, 119] for building a knowledge representation of the Imperial College Library Regulations. We need the regulations as a specification of how the library system should be and how the computer system should be.

The requirement for violation detection is the main reason why we can use deontic logic for system specification. The contrary-to-duty structure can be expressed in library specification as follows [117]:

- The agent X shall return the book Y by date due.
- If the agent X returns the book Y by date due then disciplinary action shall not be taken against the agent X.
- If the agent X does not return the book Y by date due then disciplinary action shall taken against the agent X.

- The agent X does not return the book Y by date due.

The language we use to formulate library specification must allow for the consistent expression of contrary-to-duty scenarios.

2.3.1 Legal knowledge representation

To begin with, McCarty based on TAXMAN project discussed why semantical accounts of *permission* and *obligation* will be required in an adequate knowledge representation language for the legal domain [143, 144]. The modern style interaction of law and logic is promoted by Stig Kanger [122, 123, 124], Ingmar Pörn [172, 173], Lars Lindahl (Position and Change [135]), and C. E. Alchourrón and E. Bulygin (Normative Systems [1]). The concept of a legal or generally normative system is developed and pursued by a considerable number of publications dealing with the law's logical aspects [152]. Legal norms set what is lawfully obligated or permitted and what is not. So norms are action-guiding and regulate our actions [221]. Normative systems as coding systems are used to supporting obligations and permissions [1]. Input/output logic [140] and the theory of joining-systems [136] are two of the approaches that in recent years have taken up and developed the normative systems approach to the analysis of (legal) norms.

2.3.2 Machine ethics knowledge representation

Machine ethics is concerned with giving machines ethical principles or a procedure for discovering a way to resolve the ethical dilemmas they might encounter, enabling them to function in an ethically responsible manner through their own ethical decision making [4]. There are two main approaches to developing these machines: *the bottom-up approaches* by using a large amount of data and *the top-down approaches*, a set of rules is the basis for evaluating the morality of a decision [224]. Bringsjord [59] argues that deontic logic is a useful knowledge representation for engineering ethically correct robots. Deontic logic can be used as a top-down approach for coding ethical theories [122, 145, 146, 11, 151]. The LogiKEy framework [37] is a package for designing and implementing ethical theories based on translating an established deontic logic into HOL.

2.4 Classical Higher-Order Logic

In this section we introduce classical higher-order logic (HOL). The presentation adapted from [28].

HOL is based on simple typed λ-calculus. We assume that the set $\mathcal{T}$ of simple types is freely generated from a set of basic types $\{o, i\}$ using the function type constructor $\rightarrow$. Type o denotes the set of Booleans whereas type i refers to a non-empty set of individuals.

For $\alpha, \beta, o \in \mathcal{T}$, the *language of HOL* is generated as follows:

$$s, t ::= p_\alpha | X_\alpha | (\lambda X_\alpha s_\beta)_{\alpha\rightarrow\beta} | (s_{\alpha\rightarrow\beta}\, t_\alpha)_\beta$$

where p_α represents a typed constant symbol (from a possibly infinite set $\mathcal{P}_\alpha$ of such constant symbols) and X_α represents a typed variable symbol (from a possibly infinite set $\mathcal{V}_\alpha$ of such symbols). $(\lambda X_\alpha s_\beta)_{\alpha\rightarrow\beta}$ and $(s_{\alpha\rightarrow\beta}\, t_\alpha)_\beta$ are called *abstraction* and *application*, respectively. HOL is a logic of terms in the sense that the *formulas of HOL* are given as terms of type o. Moreover, we require a sufficient number of primitive logical connectives in the signature of HOL, i.e., these logical connectives must be contained in the sets $\mathcal{P}_\alpha$ of constant symbols. The primitive logical connectives of choice in this thesis are $\neg_{o\rightarrow o}$, $\vee_{o\rightarrow o\rightarrow o}$, $\Pi_{(\alpha\rightarrow o)\rightarrow o}$ and $=_{\alpha\rightarrow\alpha\rightarrow o}$. The symbols $\Pi_{(\alpha\rightarrow o)\rightarrow o}$ and $=_{\alpha\rightarrow\alpha\rightarrow o}$ generally assumed for each type $\alpha \in \mathcal{T}$. From the selected set of primitive connectives, other logical connectives can be introduced as abbreviations. Type information as well as brackets may be omitted if obvious from the context, and we may also use infix notation to improve readability. For example, we may write $(s \vee t)$ instead of $((\vee_{o\rightarrow o\rightarrow o}\, s_o)\, t_o)_o$. We often write $\forall X_\alpha s_o$ as syntactic sugar for $\Pi_{(\alpha\rightarrow o)\rightarrow o}(\lambda X_\alpha s_o)$.

The notions of *free variables*, *α-conversion*, *$\beta\eta$-equality* and *substitution* of a term s_α for a variable X_α in a term t_β, denoted as $[s/X]t$, are defined as usual.

The semantics of HOL are well understood and thoroughly documented [26]. In the remainder, the semantics of choice is Henkin's general models [110].

A *frame* D is a collection $\{D_\alpha\}_{\alpha\in\mathcal{T}}$ of nonempty sets D_α, such that $D_o = \{T, F\}$, denoting truth and falsehood, respectively. $D_{\alpha\rightarrow\beta}$ represents a collection of functions mapping D_α into D_β.

A *model* for HOL is a tuple $M = \langle D, I\rangle$, where D is a frame and I is a family of typed interpretation functions mapping constant symbols p_α to appropriate elements of D_α, called the *denotation of* p_α. The logical connectives $\neg$, $\vee$, Π and $=$ are always given in their expected standard denotations. A *variable assignment* g maps variables X_α to elements in D_α. $g[d/W]$ denotes the assignment that is identical to g, except for the variable W, which is now mapped to d. The *denotation* $\|s_\alpha\|^{M,g}$ of a HOL term s_α on a model $M = \langle D, I\rangle$ under assignment g is an element $d \in D_\alpha$ defined in the following way:

$$\begin{array}{rcl}
\|p_\alpha\|^{M,g} & = & I(p_\alpha) \\
\|X_\alpha\|^{M,g} & = & g(X_\alpha) \\
\|(s_{\alpha\to\beta}\, t_\alpha)_\beta\|^{M,g} & = & \|s_{\alpha\to\beta}\|^{M,g}(\|t_\alpha\|^{M,g}) \\
\|(\lambda X_\alpha s_\beta)_{\alpha\to\beta}\|^{M,g} & = & \text{the function } f \text{ from } D_\alpha \text{ to } D_\beta \text{ such that} \\
 & & f(d) = \|s_\beta\|^{M,g[d/X_\alpha]} \text{ for all } d \in D_\alpha
\end{array}$$

Since $I(\neg_{o\to o})$, $I(\vee_{o\to o\to o})$, $I(\Pi_{(\alpha\to o)\to o})$ and $I(=_{\alpha\to\alpha\to o})$ always denote the standard truth functions, we have:

1. $\|(\neg_{o\to o}\, s_o)_o\|^{M,g} = T$ iff $\|s_o\|^{M,g} = F$.
2. $\|((\vee_{o\to o\to o}\, s_o)\, t_o)_o\|^{M,g} = T$ iff $\|s_o\|^{M,g} = T$ or $\|t_o\|^{M,g} = T$.
3. $\|(\forall X_\alpha s_o)_o\|^{M,g} = \|(\Pi_{(\alpha\to o)\to o}(\lambda X_\alpha s_o))_o\|^{M,g} = T$ iff for all $d \in D_\alpha$ we have $\|s_o\|^{M,g[d/X_\alpha]} = T$.
4. $\|((=_{\alpha\to\alpha\to o}\ s_\alpha)\, t_\alpha)_o\|^{M,g} = T$ iff $\|s_\alpha\|^{M,g} = \|t_\alpha\|^{M,g}$.

A HOL formula s_o is *true* in a Henkin model M under the assignment g if and only if $\|s_o\|^{M,g} = T$. This is also expressed by the notation $M, g \models^{HOL} s_o$. A HOL formula s_o is called *valid* in M, denoted as $M \models^{HOL} s_o$, if and only if $M, g \models^{HOL} s_o$ for all assignments g. Moreover, a formula s_o is called *valid*, denoted as $\models^{HOL} s_o$, if and only if s_o is valid in all Henkin models M. Finally, we define $\Sigma \models^{HOL} s_o$ for a set of HOL formulas Σ if and only if $M \models^{HOL} s_o$ for all Henkin models M with $M \models^{HOL} t_o$ for all $t_o \in \Sigma$.

2.5 Shallow Semantical Embedding

The so-called *shallow semantical embedding* approach is developed by Benzmüller [24] for translating (the semantics of) classical and non-classical logics into HOL. The following is a mathematical description of this semantical embedding.

Algebraically, we can based on a language $\mathbb{L}$ define a logic Γ as a triple $\Gamma = \langle \mathbb{L}, \vdash, \vDash \rangle$ where $\vdash \subseteq \mathbb{L} \times \mathbb{L}$ and $\vDash \subseteq \mathbb{L} \times \mathbb{L}$ ($\vdash \subseteq 2^{\mathbb{L}} \times \mathbb{L}$ and $\vDash \subseteq 2^{\mathbb{L}} \times \mathbb{L}$) axiomatize the syntax and semantics of logic Γ such that are sound and complete respect to each other in the weak or strong sense.

$$\text{Weak sense:: } \forall \varphi, \psi \in \mathbb{L} : \quad (\top, \psi) \in \vdash \quad \text{iff} \quad (\top, \psi) \in \vDash$$

$$\text{Strong sense:: } \forall \Gamma \subseteq \mathbb{L}, \psi \in \mathbb{L} : \quad (\Gamma, \psi) \in \vdash \quad \text{iff} \quad (\Gamma, \psi) \in \vDash$$

A logical embedding $f : \mathbb{L}_s \rightarrow \mathbb{L}_t$ is a translation from a source logic to a target logic that keeps the logical properties of the source logic in the target logic. Suppose $\Gamma_s = \langle \mathbb{L}_s, \vdash_s, \vDash_s \rangle$ and $\Gamma_t = \langle \mathbb{L}_t, \vdash_t, \vDash_t \rangle$ are two logics.

Shallow semantical embedding A function $f : \mathbb{L}_s \rightarrow \mathbb{L}_t$ is faithful shallow semantical embedding as one of the following cases:

- $\forall \varphi, \psi \in \mathbb{L}_s \quad (\top, \psi) \in \vDash_s$ if and only if $(f(\top), f(\psi)) \in \vDash_t$.
- $\forall \Gamma \subseteq \mathbb{L}_s, \psi \in \mathbb{L}_s \quad (\Gamma, \psi) \in \vDash_s$ if and only if $(f(\Gamma), f(\psi)) \in \vDash_t$.[4]

Sometimes technically, it is useful to add internal functions $g_1 : \mathbb{L}_s \rightarrow \mathbb{L}_s$ or $g_2 : \mathbb{L}_t \rightarrow \mathbb{L}_t$ for having faithfulness of the embedding. For instance we have:[5]

$$\forall \varphi, \psi \in \mathbb{L}s \quad (g_1(\top), g_1(\psi)) \in \vDash_s \text{ if and only if } (g_2(f(\top)), g_2(f(\psi))) \in \vDash_t.$$

Specifically, when the target logic is higher-order logic, our shallow semantical embedding could be based on an efficient algorithm that translates the language $\mathbb{L}_s$ into $\mathbb{L}_t$.

First, we should semantically characterize the source logic using set theory. So for each logical constant, we have a corresponding equation. For example in Kripke semantics we can define the main logical constants as follows:

- $\neg\varphi$ in the model $\langle W, R, V \rangle$ is the set of all the states that does not satisfies φ.
- $\varphi \vee \psi$ in the model $\langle W, R, V \rangle$ is the set of all the states that satisfies φ or ψ.

[4] $f(\Gamma) = \{f(\varphi) \,|\, \varphi \in \Gamma\}$

[5] This maybe allows for trivializations: e.g., take $g_1 : \phi \rightarrow \top$ and $g_2 : \phi \rightarrow \top$.

Second, since HOL is an expressive language we can represent the equations that characterize semantics of the source logic and translate the syntax of source logic employing these semantical representations into HOL as the target logic.

In this thesis, we use an additional function $vld : HOL \to HOL$ with translation function $\lfloor . \rfloor : \mathbb{L}_s \to HOL$. So our shallow semantical embedding is faithful as one of the two following cases:

- $\forall \varphi, \psi \in \mathbb{L}_s \quad (\top, \psi) \in \models_s$ if and only if $(vld(\lfloor \top \rfloor), vld(\lfloor \psi \rfloor)) \in \models_{HOL}$.
- $\forall \Gamma \subseteq \mathbb{L}_s, \psi \in \mathbb{L}_s \quad (\Gamma, \psi) \in \models_s$ if and only if $(vld(\lfloor \Gamma \rfloor), vld(\lfloor \psi \rfloor)) \in \models_{HOL}$.

2.6 Theorem Provers

The main question in theorem provers is as follows:

Given a set of axioms $A = \{\varphi_1, ..., \varphi_m\}$ and a hypothesis H in first-order or higher-order logic. An automated theorem prover tries to answer that the set of axioms implies $A \models H$ hypothesis or not $A \not\models H$. Refutational theorem provers deal with the equivalent problem of showing that the set $A \cup \neg H$ is inconsistent.

First-order theorem provers The landmark paper of Robinson [180] is the main base for building first-order theorem provers. Davis and Putnam [83] developed a resolution rule for propositional logic. Robinson generalized this method for first-order logic. Robinson [180] invented refutational theorem proving in its contemporary form. He introduced resolution calculus, which generally is based on two inference rules:

$$\text{(Binary) Resolution } \frac{\phi \vee \varphi \quad \chi \vee \neg\psi}{(\phi \vee \chi)\sigma} \qquad\qquad \text{(Positive) Factoring } \frac{\phi \vee \varphi \vee \psi}{(\phi \vee \varphi)\sigma}$$

where σ is the most general unifier of the atomic formulas φ and ψ. Resolution is a method for determining whether certain sets of formulas are satisfiable: the contradiction can be derived from any unsatisfiable set. A key point in the resolution is the existence (and uniqueness) of a most general unifier for any two unifiable terms. Unification is an algorithmic procedure of solving equations between symbolic expressions. Unification, as a selection mechanism for inferences, provides an effective way of identification and demonstration of formula unsatisfiability [15]. Substitution is a solution of a such unification problem. For a more detailed discussion see [137, 15, 82].

Higher-order theorem provers Andrews [5] adopts the resolution method for higher-order logic. Andrew's resolution method uses an enumeration of the universe and avoids unification completely. Following are some well know HOL provers family. In this thesis we only use Isabelle/HOL.

HOL HOL88, HOL98, and HOL4 which are members of the HOL family [106].

LEO Benzmüller developed automated theorem provers LEO [33, 19] and LEO-II [41] based on resolution method for Henkin semantics. Leo-III [191] is one of the most effective higher-order automated reasoning systems to date as new LEO member.

Isabelle/HOL Isabelle/HOL is an interactive proof assistant with a sophisticated user-interface and, besides, integrates various state-of-the-art reasoning tools. Isabelle/HOL is integrated with the ATP systems and satisfiability modulo theories (SMT) solvers via the Sledgehammer tool [45]. Some of solver included in Isabelle/HOL are: the equational reasoner *simp*; the untyped tableau prover *blast*; the simplifier and classical reasoners *auto*, *force*, and *fast*; the best-first search procedure *best*. Isabelle/HOL also provides two model finders, Nitpick [44] and Nunchaku [79].

2.6.1 Some experiments in Isabelle/HOL

Theorem provers are AI modern tools for both verification and specification. Since higher-order logic is an expressive language, higher-order theorem provers are capable of specification, especially mathematical objects such as graphs. Common sense concepts include time, space, and belief are the main objects of artificial intelligence. Some of them are mathematical objects that are formalized by mathematical theories such as geometry, and we could implement these theories in the theorem provers. Some of them are intentional concepts that could be represented in a (modal) logical setting. We can specify these intentional concepts by logical implantations of possible worlds semantics in the theorem provers.

Meta-logical properties Theorem provers are logic-based tools for proving mathematical truths of a formal system. For example, Benzmüller [23] proved the cut-elimination property for quantified conditional logic by employing shallow semantical embedding of conditional logic in HOL. Moreover, Halkjær From [90] provided a soundness and completeness proof for modal logic system **K** and epistemic logic in Isabelle/HOL.

Kurt Gödel‘s ontological argument for the existence of God The ontological proof for the existence of God is based on the intentional concepts of "Property", "Positive", "God-like being", "Possess", and "Essence". In addition, two logical concepts of "Necessity" and "Possibility" represented by modal logic ($S5$) are required. The existence of God is based on the specification of mentioned intentional concepts and by using the inference method of modal logic. Benzmüller and Woltzenlogel Paleo [42] implemented this proof in the higher-order theorem provers includes Isabelle/HOL by employing shallow semantical embedding of modal logic into HOL.

The Principia Logico-Metaphysica The scopes of Principia Logico-Metaphysica are metaphysics, mathematics, and the sciences [227]. So the automation of this theory needs a lot of different objects and intentional concepts. Three basic objects of this theory are as follows:

- Basic logical objects: Propositions and properties.
- Mathematical objects and relations: Natural numbers and natural sets.
- Philosophical objects: Platonic forms, situations, worlds, and time.

Kirchner et al. [125] implemented Abstract Object Theory that is the canonical part of Principia Logico-Metaphysica utilizing shallow semantical embedding in Isabelle/HOL.

Alan Gewirth's Proof for the Principle of Generic Consistency Benzmüller et al. [29], see Chapter 5, implemented faithfully a dyadic deontic logic by Carmo and Jones [67] in Isabelle/HOL. Fuenmayor and Benzmüller [91] based on this logical implantation and using the expressivity power of HOL provided a specification for Kaplan's theory in Isabelle/HOl. Kaplan's logical system models context-sensitivity by representing contexts as tuples of features $\langle Agent(c); Position(c); World(c); Time(c)\rangle$. They introduced two more dimensional semantics for a domain of individuals and "contexts of use" of Kaplanian. Also they specified concept objects of "Agent", "Position", "World", and "Time" in HOL. The more complicated specification was done for "Goodness", "Freedom", and "Well-Being".

2.7 Semantical Embedding of KD in HOL

By introducing a new type i to denote possible worlds, the formulas of **K** are identified with certain HOL terms (predicates) of type $i \to o$. The type $i \to o$ is abbreviated as τ in the remainder. This allows us to represent the formulas of **K** as functions from possible worlds to truth values in HOL and therefore the truth of a formula can explicitly be evaluated in a particular world. The HOL signature is assumed to further contain the constant symbol $r_{i \to i \to o}$. Moreover, for each atomic propositional symbol p^j of **K**, the HOL signature must contain the corresponding constant symbol p^j_τ. Without loss of generality, we assume that besides those symbols and the primitive logical connectives of HOL, no other constant symbols are given in the signature of HOL.

The mapping $\lfloor \cdot \rfloor$ translates a formula φ of **K** into a term $\lfloor \varphi \rfloor$ of HOL of type τ. The mapping is defined recursively:

$$\begin{array}{lcl}
\lfloor p^j \rfloor & = & p^j_\tau \\
\lfloor \neg \varphi \rfloor & = & \neg_{\tau \to \tau} \lfloor \varphi \rfloor \\
\lfloor \varphi \vee \psi \rfloor & = & \vee_{\tau \to \tau \to \tau} \lfloor \varphi \rfloor \lfloor \psi \rfloor \\
\lfloor \Box \varphi \rfloor & = & \Box_{\tau \to \tau} \lfloor \varphi \rfloor
\end{array}$$

$\neg_{\tau \to \tau}$, $\vee_{\tau \to \tau \to \tau}$ and $\Box_{\tau \to \tau}$ abbreviate the following terms of HOL:

$$\begin{array}{ll}
\neg_{\tau \to \tau} & = \lambda A_\tau \lambda X_i \neg (A\, X) \\
\vee_{\tau \to \tau \to \tau} & = \lambda A_\tau \lambda B_\tau \lambda X_i (A\, X \vee B\, X) \\
\Box_{\tau \to \tau} & = \lambda A_\tau \lambda X_i \forall Y_i (\neg (r_{i \to i \to o} X\, Y) \vee A\, Y)
\end{array}$$

Analyzing the truth of formula φ, represented by the HOL term $\lfloor \varphi \rfloor$, in a particular world w, represented by the term w_i, corresponds to evaluating the application $(\lfloor \varphi \rfloor\, w_i)$. In line with the previous work [40], we define $vld_{\tau \to o} = \lambda A_\tau \forall S_i (A\, S)$. With this definition, validity of a formula s in **K** corresponds to the validity of the formula $(vld \lfloor \varphi \rfloor)$ in HOL, and vice versa.

To prove the soundness and completeness, that is, faithfulness, of the above embedding, a mapping from Kripke models into Henkin models is employed.

Lemma 2.2 (Kripke models ⇒ Henkin models)**.** *For every Kripke model $M = \langle W, R, V \rangle$ there exists a corresponding Henkin model H^M, such that for all formulas δ of* **K***, all* *assignments g and worlds s it holds:*

$$M, s \models \delta \text{ if and only if } \|\lfloor\delta\rfloor\, S_i\|^{H^M, g[s/S_i]} = T$$

Proof. See [39, 40]. □

Lemma 2.3 (Henkin models ⇒ Kripke models)**.** *For every Henkin model $H = \langle\{D_\alpha\}_{\alpha\in\mathcal{T}}, I\rangle$ there exists a corresponding Kripke model M_H, such that for all formulas δ of* **K***, all assignments g and worlds s it holds:*

$$\|\lfloor\delta\rfloor S_i\|^{H, g[s/S_i]} = T \text{ if and only if } M_H, s \models \delta$$

Proof. See [39, 40].

□

The following table summarizes the alignment of Kripke models and Henkin models. For the class of Kripke models $\langle W, R, V\rangle$ that satisfies some properties, such as *reflexivity*, the corresponding class of Henkin models also needs to satisfy the higher-order counter part of this property. For instance, in system **KD** the class of Kripke models satisfies the property of *seriality*, which corresponds to axiom **D**. The higher-order counter part of this property is represented as $SER : \forall X_i \exists Y_i (r_{i\to i\to o} X_i Y_i)$, where constant symbol $r_{i\to i\to o}$ denotes the accessibility relation.

Kripke model $\langle W, R, V\rangle$	Henkin model $\langle D, I\rangle$
Possible worlds $s \in W$	Set of individuals $s_i \in D_i$
Accessibility relation R sRu	Binary predicates $r_{i\to i\to o}$ $Ir_{i\to i\to o}(s_i, u_i) = \top$
Propositional letters p^j Valuation function $s \in V(p^j)$	Unary predicates $p^j_{i\to o}$ Interpretation function $Ip^j_{i\to o}(s_i) = \top$

These correspondences between Kripke and Henkin models include the assumptions that have been formulated at the beginning of this section.

Theorem 2.4 (Faithfulness of the embedding of system **K** in HOL)**.**

$$\models_{\mathcal{C}_K} \varphi \text{ if and only if } \models^{HOL} vld\,\lfloor\varphi\rfloor$$

Proof. See [39, 40]. □

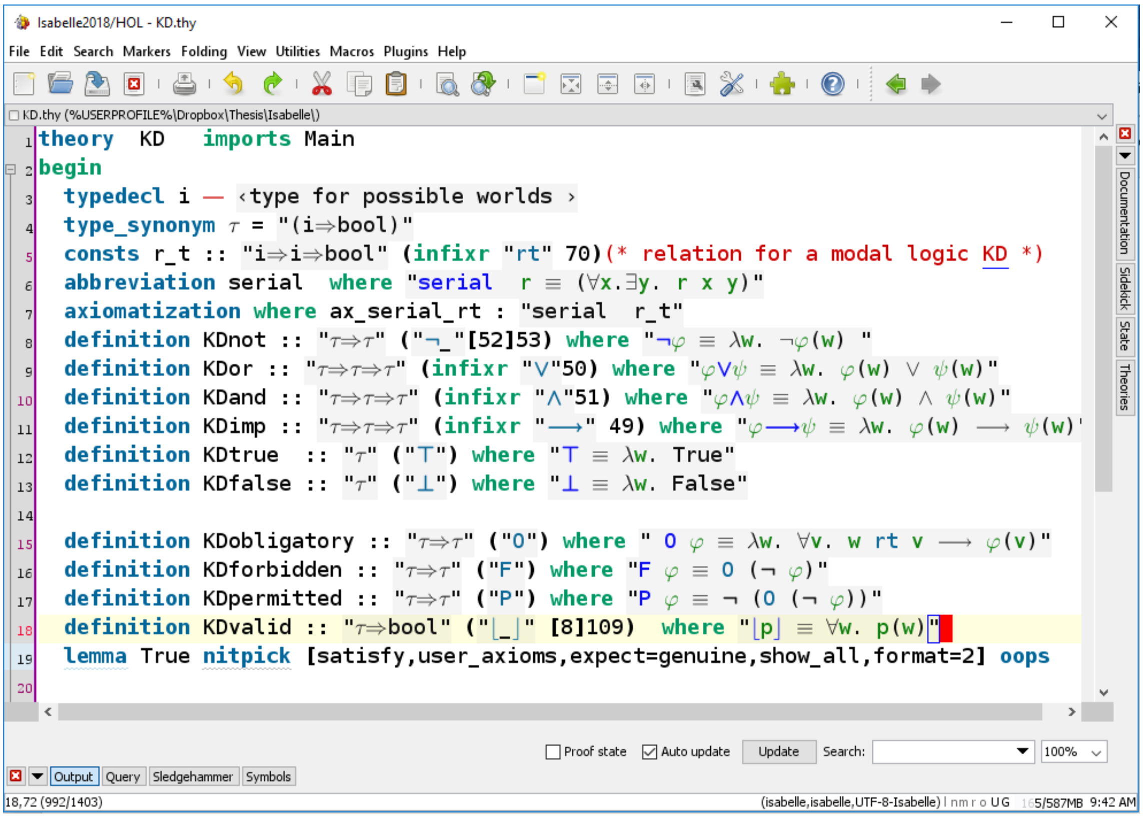
Theorem 2.5 (Faithfulness of the embedding of system **KD** in HOL)**.**

$$\models_{\mathcal{C}_{KD}} \varphi \textit{ if and only if } \{SER\} \models^{HOL} vld\lfloor\varphi\rfloor$$

Proof. See [39, 40] for a proof of the faithfulness of the embedding of modal logic **K** in HOL. The result for logic **KD** follows as a simple corollary; see also Section 3.3 in [35]. □

2.7.1 Implementation of KD in Isabelle/HOL

The semantical embedding of **KD** in HOL has been implemented in the higher-order proof assistant Isabelle/HOL [156]. Figure 2.1 displays the respective encoding. Some explanations are in order:


Isabelle2018/HOL - KD.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
KD.thy (%USERPROFILE%\Dropbox\Thesis\Isabelle\)
Documentation
Sidekick
State
Theories
□ Proof state
☑ Auto update
Update
Search:
100%
Output
Query
Sledgehammer
Symbols
18,72 (992/1403)
(isabelle,isabelle,UTF-8-Isabelle) | nm r o UG 165/587MB 9:42 AM

```
1  theory  KD   imports Main
2  begin
3    typedecl i — ‹type for possible worlds ›
4    type_synonym τ = "(i⇒bool)"
5    consts r_t :: "i⇒i⇒bool" (infixr "rt" 70)(* relation for a modal logic KD *)
6    abbreviation serial  where "serial  r ≡ (∀x.∃y. r x y)"
7    axiomatization where ax_serial_rt : "serial  r_t"
8    definition KDnot :: "τ⇒τ" ("¬_"[52]53) where "¬φ ≡ λw. ¬φ(w) "
9    definition KDor :: "τ⇒τ⇒τ" (infixr "∨"50) where "φ∨ψ ≡ λw. φ(w) ∨ ψ(w)"
10   definition KDand :: "τ⇒τ⇒τ" (infixr "∧"51) where "φ∧ψ ≡ λw. φ(w) ∧ ψ(w)"
11   definition KDimp :: "τ⇒τ⇒τ" (infixr "⟶" 49) where "φ⟶ψ ≡ λw. φ(w) ⟶ ψ(w)'
12   definition KDtrue  :: "τ" ("⊤") where "⊤ ≡ λw. True"
13   definition KDfalse :: "τ" ("⊥") where "⊥ ≡ λw. False"
14
15   definition KDobligatory :: "τ⇒τ" ("O") where " O φ ≡ λw. ∀v. w rt v ⟶ φ(v)"
16   definition KDforbidden :: "τ⇒τ" ("F") where "F φ ≡ O (¬ φ)"
17   definition KDpermitted :: "τ⇒τ" ("P") where "P φ ≡ ¬ (O (¬ φ))"
18   definition KDvalid :: "τ⇒bool" ("⌊_⌋" [8]109)  where "⌊p⌋ ≡ ∀w. p(w)"
19   lemma True nitpick [satisfy,user_axioms,expect=genuine,show_all,format=2] oops
20
```

FIGURE 2.1: Implementation of **KD** in Isabelle/HOL

- On line 3 the primitive type i for possible words is introduced.
- On line 4 the type τ for formulas is introduced.
- On line 5 the constant r_t is introduced. r_t encodes the accessibility relation.
- Line 7 restrict the accessibility relation by seriality property.
- Lines 8–13 define the Boolean logical connectives.

- Lines 15–17 define the monadic deontic operators: obligation, forbidden and permission.
- Line 18 introduce the notion of global validity.
- Line 19 the model finder Nitpick [44] confirms the consistency of the logical system.

Chapter 3

Discursive Input/Output Logic

The chapter introduces a new logical framework for normative reasoning. It is a unification of the two main approaches for deontic logic: *modal logic* and *norm-based* approaches, see Section 1.3 and Section 2.2. Each approach has its advantages. An advantage of the modal logic approach is the capability to extend with other modalities such as epistemic or temporal operators, and advantages of the norm-based approach include the ability to explicitly represent normative codes such as legal systems and using non-monotonic logic techniques of common sense reasoning. Unifying these two approaches will provide us with a framework with all of these advantages simultaneously. For example, we can design a normative temporal system, which changes over time. The temporal reasoning comes from the advantage of the modal logic part and changes operators (expansion, contraction) from the norm-based part. There are other frameworks, such as adaptive logic [197, 198], that combine modal logic and norm-based approaches. The novelty of our approach is *semantical unification*. The unification is based on bringing the core semantical elements of both approaches in a single unit. We introduce a semantics for deriving deontic modals based on the interaction between normative reasoning and informational and motivational modalities. We use input/output logic for normative reasoning and the Kratzerian framework for representing different sets of information and motivation.

The question of this chapter is: *How can we integrate the norm-based approach in the sense of Makinson [138] to the classic semantics in the sense of the Kratzerian framework [128] ?* To achieve a more uniform semantics [112, 92, 166] for deontic modals, we build I/O operations on top of Boolean algebras for deriving permissions and obligations. The approach is close to the work is done by Gabbay, Parent, and van der Torre: a geometrical view of I/O logic [94]. For defining the I/O framework over an algebraic setting, they use the algebraic counterpart, *upward-closed set of the infimum of* A, for *the propositional logic consequence relation* ("$Cn(A)$"), within lattices. They have characterized only the simple-minded output operation. We show that by choosing the "Up" operator, *upward-closed set*, as the algebraic counterpart of the "Cn" operator and by using the reversibility of inference rules in the I/O proof system, we can characterize all the previously studied I/O systems and find many more new logical systems. This suggested framework has a significant difference from other types of input/output logics. In contrast to the earlier input/output logics, we define non-adjunctive input/output operations. Non-adjunctive logical systems are those where deriving the conjunctive formula $\varphi \wedge \psi$ from the set $\{\varphi, \psi\}$ fails [77, 78]. These systems are especially suited for modeling discursive reasoning. In fact, the first non-adjunctive system in the literature was proposed by Jaśkowski [116] for discursive systems.

> "[...] such a system which cannot be said to include theses that express opinions in agreement with one another, be termed a discursive system. To bring out the nature of the theses of such a system it would be proper to precede each thesis by the reservation: "in accordance with the opinion of one of the participants of the discourse"[...]. Hence the joining of a thesis to a discursive system has a different intuitive meaning than has assertion in an ordinary system." (Jaśkowski [116])

We build two groups of I/O operations for deriving permissions and obligations over Boolean algebras. The main difference between the two operations is similar to the possible world semantics characterization of box and diamond, where box is closed under AND, $((\Box\varphi \wedge \Box\psi) \rightarrow \Box(\varphi \wedge \psi))$, and diamond not.[1] For each deontic modal of permission and obligation, a primitive operation[2] is defined in the strong sense [1].[3]

[1] In the main literature of input/output logic developed by Makinson and van der Torre [140], Parent, Gabbay, and van der Torre [163], Parent and van der Torre [165, 167, 169], and Stolpe [192, 193, 196] at least one form of AND inference rule is present (see the related table in Subsection 2.2.2). Sun [199] analyzed norms derivations rules of input/output logic in isolation. Still, it is not clear how we can combine them and build new logical systems, specifically systems that do not admit the rule of AND. For building a primitive operation for producing permissible propositions, we need to remove the AND rule from the proof system.

[2] Von Wright [220] defined permission as the primitive concept and obligation as the dual of it. Later, in the central literature of deontic logic, obligation introduced as the primitive concept and permission defined as the dual concept, as well in the earlier input/output logic for permission [142]. Moreover, in the I/O literature, permission base on derogation is studied by Stolpe [195, 194] and based on constraints by Boella and van der Torre [54].

[3] For example Alchourrón, and Bulygin [1] define strong permission as :"To say the p is strongly permitted in the case q by the system α means that a norm to the effect that p is permitted in q is a consequence of α".

The “Up” operator, for a given set A, sees all the elements that are upper than or equal to the elements of A by usual ordering in lattices. This operator instead of the “Cn” operator is not closed under conjunction so that we do not have $a \wedge \neg a \in Up(a, \neg a)$. Consequently, the new I/O operations defined by the “Up” operator instead of the “Cn” operate on inputs independently and do not derive joint outputs (are not closed under AND). According to the reversibility of inference rules in the I/O proof systems, we show how it is possible to add AND and other rules, required for obligation [140], to the proof systems and find I/O operations for them. The introduced I/O operations admit normative conflicts and could receive technical benefits from the constrained version of I/O logics [141] for resolving normative conflicts. The introduced framework is a form of paraconsistent logic for admitting normative conflicts (see Subsection 6.1, [102]). Moreover, we use Stone’s representation theorem for Boolean algebras for integrating input/output logic with possible world semantics.[4]

The I/O operations presented here are *Tarskain* or *closure operator* over a set of conditional norms so that they can be used as logical operators for reasoning about normative systems. The algebrization of the I/O framework shows more similarity with the theory of joining-systems [136] that is an algebraic approach for study normative-systems over Boolean algebras. We can say that norms in the I/O framework play the same role of joining in the theory of Lindahl and Odelstal [136, 200].

The chapter is structured as follows: Section 3.1 is about integrating the norm-based approach to the standard semantics for deontic modals. Section 3.2 and 3.3 give the soundness and completeness results of I/O operations for deriving permissions and obligations. Section 3.4 generalizes the I/O operations over any abstract logic. Section 3.5 concludes the chapter.

3.1 Norms and Deontic Modals

Before going to our discussion consider following basic logical notions:

- W is the set of possible worlds.
- $P(W)$[5] is the set of (atomic) propositions.[6] A proposition x is true in a world w if and only if $w \in x$.

[4] Another possible worlds semantics of I/O logic is studied by Bochman [47] for causal reasoning. It has no direct connection to the operational semantics (see Subsection 2.4, [164]).

[5] $P(W)$ denotes the power set of W.

[6] Logical connectives can be defined as usual. $\top := W$ and $\bot := \emptyset$.

- If A is a set of propositions,
  - $\bigcap A \neq \emptyset$ means that A is consistent.
  - $\bigcap A \subseteq x$ means that x follows from A.
  - $\bigcap A \cap x \neq \emptyset$ means that x is compatible with A ($A \cup \{x\}$ is consistent).
- f is a function, which is termed the *modal base*, assigns to every possible world (w) a set of propositions ($f(w)$), which is called premise set, that are known in w by us. We use the same formal definition for the *ordering source* function g.
- Normative system N denotes a set of norms (a, x), which the body and head are propositions. More explicitly, N^O denotes a set of obligatory norms and N^P a set of permissive norms. If $(a, x) \in N^O$, it means that "given a, it is obligatory that x" and if $(a, x) \in N^P$, it means that "given a, it is permitted that x."
- $x \in out(N^O, A)$ means given normative system N^O and input set A (state of affairs), x (obligation) is in the output (similar definition works for permission $x \in out(N^P, A)$). The output operations resemble inferences, where inputs need not be included among outputs, and outputs need not be reusable as inputs [140].

In the classic semantics, modals are quantifiers over possible worlds. Deontic modals are quantifiers over the best worlds in the domain of accessible worlds, represented as $\bigcap f(w)$ in the Kratzerian framework: *ought* or *have-to* are modal verbs of necessity that the prejacent (i.e., the proposition under the modal operator) is true in all of the best worlds and *may* or *can* are modal verbs of possibility that the prejacent is true in some of the best worlds [219, 218]. We can define deontic modals in the Kratzerian framework as follows [218]:

$$[[\text{be-allowed-to}]]^{w,f,g} = \lambda x\, (Best_{g(w)}(\bigcap f(w)) \cap x \neq \emptyset)$$

$$[[\text{have-to}]]^{w,f,g} = \lambda x\, (Best_{g(w)}(\bigcap f(w)) \subseteq x)$$

where $Best_{g(w)}(\bigcap f(w))$ is given as follows:

$$\{w^{'} \in \bigcap f(w) : \neg \exists w^{''} \in \bigcap f(w) \text{ such that } \exists y \in g(w) : w^{''} \in y \text{ and } w^{'} \notin y\}$$

In the definition, the domain of quantification is selected by a modal base and an ordering source for deriving deontic modals. Moreover, there are two ways for quantification:

compatibility and entailment. We employ the *modal base* and *ordering source* functions, from the Kratzerian framework [128, 129] and the *detachment* approach [164] from I/O framework, instead of quantification. As an advantage of the detachment approach, we can characterize derivation systems that do not admit, for example, weakening of the output (WO) or strengthening of the input (SI). In Section 3.2 and Section 3.3, we develop various detachment methods are using different I/O operations, in turn, for permission and obligation.

In input/output logic, the main semantical construct for normative propositions is the output operation, which represents the set of normative propositions related to the normative system N, regarding the state of affairs A, namely $out(N, A)$. *Detachment* is the basic idea of the semantics of input/output logic [164]. The interpretation of "x is obligatory if a" is that "x can be detached in context a". In a discourse, the context is represented by a modal base or an ordering source in the Kratzerian framework. To unify the norm-based semantics with the classic semantics, in each world w, we can detach what we allowed to or have to as the output of what we know (as the input set) represented as $\bigcap f(w)$, the intersection of the propositions given by the modal base, and the corresponding normative system N.

Consistent premise sets: Suppose $\bigcap f(w) \neq \emptyset$

$$[[\text{be-allowed-to}]]^{w,f} = \lambda N^P \lambda x \ (x \in out(N^P, \{\bigcap f(w)\}))$$

$$[[\text{have-to}]]^{w,f} = \lambda N^O \lambda x \ (x \in out(N^O, \{\bigcap f(w)\}))$$

In this case, deontic modals are evaluated with reference to a set of propositions given by the modal base and a normative system in each possible world. In the same way, in the world w, we can detach what we allowed to or have to as the output of what we preferred (as the input set) represented as $\bigcap g(w)$ and the corresponding normative system N. The modal bases are always factual. Whenever there are possible inconsistencies, we can take the content as an ordering source [130]. If the set of $g(w)$ is not consistent, we can draw conclusions by looking at maximal consistent subsets.[7]

[7] In the original input/output logic we have $x \in \bigcirc(N, \{a, \neg a\})$ for all x. So when the input set is inconsistent we have explosion in the original input/output logic. Reasoning from an inconsistent premise set which is represented as set of logical formulas is an important issue for deontic modals [127, 129].

Inconsistent premise sets: Suppose $\bigcap g(w) = \emptyset$,
and $\text{Maxfamily}^{\bigcap}(g(w)) = \{\bigcap A | A \subseteq g(w) \text{ and } A \text{ is consistent and maximal}\}$
$[[\text{be-allowed-to}]]^{w,g} = \lambda N^P \lambda x \; (x \in \mathit{out}(N^P, \text{Maxfamily}^{\bigcap}(g(w))))$
$[[\text{have-to}]]^{w,g} = \lambda N^O \lambda x \; (x \in \mathit{out}(N^O, \text{Maxfamily}^{\bigcap}(g(w))))$

Both introduced modals ([[be-allowed-to]] and [[have-to]]) are in the strong sense [1]. For each one, we can define a weak sense of modality using the dual operator, which means $x \in [[\text{be-allowed-to}]]_{Weak-sense}$ if and only if $\neg x \notin [[\text{have-to}]]_{Strong-sense}$; $x \in [[\text{have-to}]]_{Weak-sense}$ if and only if $\neg x \notin [[\text{be-allowed-to}]]_{Strong-sense}$. We have presented a family of *output operations* that derive different sets of permissions in Section 3.2. In Section 3.3, we define more complicated output operations for deriving obligations. As the distinctive feature, the output operations for obligations are closed under AND which means: if $x \in \mathit{out}(N^O, A)$ and $y \in \mathit{out}(N^O, A)$, then $x \wedge y \in \mathit{out}(N^O, A)$.

3.2 Permissive Norms: Input/Output Operations

The term “input/output logic” is used broadly for a family of related systems such as *simple-minded*, *basic*, and *reusable* [168, 140]. In this section, we use a similar terminology and introduce some input/output systems for deriving permissions over Boolean algebras. Each derivation system is closed under a set of rules. Moreover, we define systems that are closed only for weakening of the output (WO) or strengthening of the input (SI). We use a bottom-up approach for characterizing different derivations systems. The rule of AND, for the output, is absent in the derivation systems presented in this section.

Definition 3.1 (Boolean algebra)**.** A structure $\mathcal{B} = \langle B, \wedge, \vee, \neg, 0, 1\rangle$ is a Boolean algebra iff it satisfies following identities:[8]

- $x \vee y \approx y \vee x$, $x \wedge y \approx y \wedge x$
- $x \vee (y \vee z) \approx (x \vee y) \vee z$, $x \wedge (y \wedge z) \approx (x \wedge y) \wedge z$
- $x \vee 0 \approx x$, $x \wedge 1 \approx x$
- $x \vee \neg x \approx 1$, $x \wedge \neg x \approx 0$
- $x \vee (y \wedge z) \approx (x \vee y) \wedge (x \vee z)$, $x \wedge (y \vee z) \approx (x \wedge y) \vee (x \wedge z)$

[8] An equation $t \approx t^{'}$ holds in an algebra $\mathcal{A}$ if its universal closure $\forall x_0 ... x_n t \approx t^{'}$ is a sentence true in $\mathcal{A}$.

For a set of variable X, we denote the set of Boolean-terms over X by $Ter(X)$ defined as follows:

$$Ter(X) = \bigcup_{n \in N} Ter_n(X)$$

where

$$Ter_0(X) = X \cup \{0, 1\}$$
$$Ter_{n+1}(X) = Ter_n(X) \cup \{a \wedge b, a \vee b, \neg a : a, b \in Ter_n(X)\}$$

Given a Boolean algebra $\mathcal{B}$, the elements of $Ter(B)$ are ordered as $a \leq b$ iff $a \wedge b = a$.[9] Since $\leq$ is antisymmetric $a \leq b$ and $b \leq a$ imply $a = b$.

Definition 3.2 (Upward-closed set)**.** Given a Boolean algebra $\mathcal{B}$, a set $A \subseteq Ter(B)$ satisfying the following property is called upward-closed.

$$\text{For all } x, y \in Ter(B), \text{ if } x \leq y \text{ and } x \in A \text{ then } y \in A$$

We denote the least upward-closed set which includes A by $Up(A)$. Up operator satisfies following properties:

- $A \subseteq Up(A)$ (Inclusion)
- $A \subseteq B \Rightarrow Up(A) \subseteq Up(B)$ (Monotony)
- $Up(A) = Up(Up(A))$ (Idempotence)

An operator that satisfies these properties is called closure operator.

Zero Boolean I/O operation

Let $N(A) = \{x \mid (a, x) \in N \text{ for some } a \in A\}$ and in a Boolean algebra $\mathcal{B}$ for $X \subseteq Ter(B)$ we define $Eq(X) = \{x \in Ter(B) | \exists y \in X, x = y\}$.

Definition 3.3 (Semantics)**.** Given a Boolean algebra $\mathcal{B}$, a normative system $N \subseteq Ter(B) \times Ter(B)$ and an input set $A \subseteq Ter(B)$, we define the zero Boolean operation as follows:

$$out_0^{\mathcal{B}}(N, A) = Eq(N(Eq(A)))$$[10]

[9] We use the symbol "=" to express that both sides name the same object. The elements of the variable set (B) that are represented by different letters are supposed to be independent in the algebra ($\mathcal{B}$) w.r.t $\leq$.

[10] Sometimes we write $Up(a, b, ...)(Eq(a, b, ...))$ instead of $Up(\{a, b, ...\})(Eq(\{a, b, ...\}))$ as well $out(N, a)$ ($derive(N, a)$) instead of $out(N, \{a\})$ ($derive(N, \{a\})$).

We put $out_0^{\mathcal{B}}(N) = \{(A, x) : x \in out_0^{\mathcal{B}}(N, A)\}$.

Definition 3.4 (Proof system)**.** Given a Boolean algebra $\mathcal{B}$ and a normative system $N \subseteq Ter(B) \times Ter(B)$, we define $(a, x) \in derive_0^{\mathcal{B}}(N)$ if and only if (a, x) is derivable from N using the rules $\{EQI, EQO\}$.[11]

$$\text{EQI}\ \frac{(a,x) \qquad a = b}{(b,x)} \qquad\qquad \text{EQO}\ \frac{(a,x) \qquad x = y}{(a,y)}$$

Given a set of $A \subseteq Ter(B)$, $(A, x) \in derive_0^{\mathcal{B}}(N)$ whenever $(a, x) \in derive_0^{\mathcal{B}}(N)$ for some[12] $a \in A$. Put $derive_0^{\mathcal{B}}(N, A) = \{x : (A, x) \in derive_0^{\mathcal{B}}(N)\}$.

Outline of proof for soundness: for the input set $A \subseteq Ter(B)$, we show that if $(A, x) \in derive_0^{\mathcal{B}}(N)$, then $x \in out_0^{\mathcal{B}}(N, A)$. By definition $(A, x) \in derive_0^{\mathcal{B}}(N)$ iff $(a, x) \in derive_0^{\mathcal{B}}(N)$ for some $a \in A$. By induction on the length of derivation and the following theorem we have $(a, x) \in derive_0^{\mathcal{B}}(N)$ iff $x \in out_0^{\mathcal{B}}(N, \{a\})$. Then by definition of $out_0^{\mathcal{B}}$ we have $x \in out_0^{\mathcal{B}}(N, A)$. If $A = \{\}$, then by definition $(A, x) \notin derive_0^{\mathcal{B}}(N)$. The outline works for the soundness of other presented systems in this chapter as well.

Theorem 3.5 (Soundness)**.** *$out_0^{\mathcal{B}}(N)$ validates EQI and EQO.*

Proof. EQI: We need to show that

$$\text{EQI}\ \frac{x \in Eq(N(Eq(a))) \qquad a = b}{x \in Eq(N(Eq(b)))}$$

If $x \in Eq(N(Eq(a)))$, then there are t_1 and t_2 such that $t_1 = a$ and $t_2 = x$ and $(t_1, t_2) \in N$. If $a = b$ then $t_1 = b$. Hence, by definition $x \in Eq(N(Eq(b)))$.

EQO: We need to show that

$$\text{EQO}\ \frac{x \in Eq(N(Eq(a))) \qquad x = y}{y \in Eq(N(Eq(a)))}$$

If $x \in Eq(N(Eq(a)))$, then there are t_1 and t_2 such that $t_1 = a$ and $t_2 = x$ and $(t_1, t_2) \in N$. If $x = y$ then $t_2 = y$. Hence, by definition $y \in Eq(N(Eq(a)))$.

□

[11] EQI stands for equivalence of the input and EQO stands for equivalence of the output.

[12] In the original input/output logic [140], it is for some conjunction a of elements in A.

Theorem 3.6 (Completeness)**.** $out_0^{\mathcal{B}}(N) \subseteq derive_0^{\mathcal{B}}(N)$.[13]

Proof. We show that if $x \in Eq(N(Eq(A)))$, then $(A, x) \in derive_0^{\mathcal{B}}(N)$. Suppose $x \in Eq(N(Eq(A)))$, then there are t_1 and t_2 such that $t_1 = a$ and $a \in A$, and $t_2 = x$ such that $(t_1, t_2) \in N$.

$$EQI\;\dfrac{EQO\;\dfrac{(t_1,t_2) \qquad t_2 = x}{(t_1,x)} \qquad t_1 = a}{(a,x)}$$

Thus, $x \in derive_0^{\mathcal{B}}(N, a)$ and then $x \in derive_0^{\mathcal{B}}(N, A)$. □

Two basic subsystems: We can construct two simple subsystems: $out_R^{\mathcal{B}}(N, A) = Eq(N(A))$ and $out_L^{\mathcal{B}}(N, A) = N(Eq(A))$. We define $(a, x) \in derive_R^{\mathcal{B}}(N)$ ($(a, x) \in derive_L^{\mathcal{B}}(N)$) if and only if (a, x) is derivable from N using the rule $\{EQO\}$ ($\{EQI\}$). By rewriting the same definition of $out_0^{\mathcal{B}}(N)$ for $out_R^{\mathcal{B}}(N)$ and $out_L^{\mathcal{B}}(N)$, and the definition of $derive_0^{\mathcal{B}}(N)$ for $derive_R^{\mathcal{B}}(N)$ and $derive_L^{\mathcal{B}}(N)$, we have :

$$out_R^{\mathcal{B}}(N) = derive_R^{\mathcal{B}}(N) \qquad\qquad out_L^{\mathcal{B}}(N) = derive_L^{\mathcal{B}}(N)$$

Simple-I Boolean I/O operation

Definition 3.7 (Semantics)**.** Given a Boolean algebra $\mathcal{B}$, a normative system $N \subseteq Ter(B) \times Ter(B)$ and an input set $A \subseteq Ter(B)$, we define the simple-I Boolean operation as follows:

$$out_I^{\mathcal{B}}(N, A) = Eq(N(Up(A)))$$

We put $out_I^{\mathcal{B}}(N) = \{(A, x) : x \in out_I^{\mathcal{B}}(N, A)\}$.

Definition 3.8 (Proof system)**.** Given a Boolean algebra $\mathcal{B}$ and a normative system $N \subseteq Ter(B) \times Ter(B)$, we define $(a, x) \in derive_I^{\mathcal{B}}(N)$ if and only if (a, x) is derivable from N using the rules $\{SI, EQO\}$.

[13] For the completeness proofs if $A = \{\}$, then by definition of $Eq(\{\}) = \{\}$ and $Up(\{\}) = \{\}$ we have $x \notin out_i^{\mathcal{B}}(N, \{\}) = \{\}$.

$$\text{SI}\ \frac{(a,x) \qquad b \leq a}{(b,x)} \qquad\qquad \text{EQO}\ \frac{(a,x) \qquad x = y}{(a,y)}$$

Given a set of $A \subseteq Ter(B)$, $(A,x) \in derive^{\mathcal{B}}_I(N)$ whenever $(a,x) \in derive^{\mathcal{B}}_I(N)$ for some[14] $a \in A$. Put $derive^{\mathcal{B}}_I(N,A) = \{x : (A,x) \in derive^{\mathcal{B}}_I(N)\}$.

Theorem 3.9 (Soundness). *$out^{\mathcal{B}}_I(N)$ validates SI and EQO.*

Proof. SI: We need to show that

$$\text{SI}\ \frac{x \in Eq(N(Up(a))) \qquad b \leq a}{x \in Eq(N(Up(b)))}$$

If $x \in Eq(N(Up(a)))$, then $\exists t_1$ such that $a \leq t_1$ and $(t_1,x) \in N$ or ($(t_1,y) \in N$ and $y = x$). Hence, if $b \leq a$, we have $b \leq t_1$ and then $x \in Eq(N(Up(b)))$.

EQO: We need to show that

$$\text{EQO}\ \frac{x \in Eq(N(Up(a))) \qquad x = y}{y \in Eq(N(Up(a)))}$$

If $x \in Eq(N(Up(a)))$, then by definition of $Eq(X)$ if $x = y$, we have $y \in Eq(N(Up(a)))$.

□

Theorem 3.10 (Completeness). *$out^{\mathcal{B}}_I(N) \subseteq derive^{\mathcal{B}}_I(N)$*

Proof. We show that if $x \in Eq(N(Up(A)))$, then $(A,x) \in derive^{\mathcal{B}}_I(N)$. Suppose $x \in Eq(N(Up(A)))$, then there is t_1 such that $a \leq t_1$ and $(t_1,x) \in N$ or $((t_1,y) \in N$ and $y = x)$ for $a \in A$. There are two cases:

$$SI\ \frac{(t_1,x) \qquad a \leq t_1}{(a,x)} \qquad\qquad SI\ \frac{EQO\ \dfrac{(t_1,y) \qquad y = x}{(t_1,x)} \qquad a \leq t_1}{(a,x)}$$

Thus, $x \in derive^{\mathcal{B}}_I(N,a)$ and then $x \in derive^{\mathcal{B}}_I(N,A)$. □

Example 3.1. *For the conditionals $N = \{(1,g),(g,t)\}$ and the input set $A = \{\}$ we have $out^{\mathcal{B}}_I(N,A) = \{\}$, and for the input set $C = \{g\}$ we have $out^{\mathcal{B}}_I(N,C) = Eq(g,t)$.*

[14] In the original input/output logic [140], it is for some conjunction a of elements in A.

Simple-II Boolean I/O operation

Definition 3.11 (Semantics)**.** Given a Boolean algebra $\mathcal{B}$, a normative system $N \subseteq Ter(B) \times Ter(B)$ and an input set $A \subseteq Ter(B)$, we define the simple-II Boolean operation as follows:

$$out^{\mathcal{B}}_{II}(N, A) = Up(N(Eq(A)))$$

We put $out^{\mathcal{B}}_{II}(N) = \{(A, x) : x \in out^{\mathcal{B}}_{II}(N, A)\}$.

Definition 3.12 (Proof system)**.** Given a Boolean algebra $\mathcal{B}$ and a normative system $N \subseteq Ter(B) \times Ter(B)$, we define $(a, x) \in derive^{\mathcal{B}}_{II}(N)$ if and only if (a, x) is derivable from N using the rules $\{WO, EQI\}$.

$$\text{WO}\ \frac{(a,x) \qquad x \leq y}{(a,y)} \qquad\qquad \text{EQI}\ \frac{(a,x) \qquad a = b}{(b,x)}$$

Given a set of $A \subseteq Ter(B)$, $(A, x) \in derive^{\mathcal{B}}_{II}(N)$ whenever $(a, x) \in derive^{\mathcal{B}}_{II}(N)$ for some $a \in A$. Put $derive^{\mathcal{B}}_{II}(N, A) = \{x : (A, x) \in derive^{\mathcal{B}}_{II}(N)\}$.

Theorem 3.13 (Soundness)**.** *$out^{B}_{II}(N)$ validates WO and EQI.*

Proof. WO: We need to show that

$$\text{WO}\ \frac{x \in Up(N(Eq(a))) \qquad x \leq y}{y \in Up(N(Eq(a)))}$$

If $x \in Up(N(Eq(a)))$, then there is t_1 such that $t_1 \leq x$ and $(a, t_1) \in N$ or $((b, t_1) \in N$ and $a = b)$. If $x \leq y$, then $t_1 \leq y$ and we have $y \in Up(N(Eq(a)))$.

EQI: We need to show that

$$\text{EQI}\ \frac{x \in Up(N(Eq(a))) \qquad a = b}{x \in Up(N(Eq(b)))}$$

If $x \in Up(N(Eq(a)))$, then there is t_1 such that $t_1 \leq x$ and $(a, t_1) \in N$ or $((c, t_1) \in N$ and $a = c)$. Hence, if $a = b$, then by definition $x \in Up(N(Eq(b)))$.

□

Theorem 3.14 (Completeness)**.** $out^{\mathcal{B}}_{II}(N) \subseteq derive^{\mathcal{B}}_{II}(N)$.

Proof. We show that if $x \in Up(N(Eq(A)))$, then $(A, x) \in derive^{\mathcal{B}}_{II}(N)$. Suppose $x \in Up(N(Eq(A)))$, then there is t_1 such that $t_1 \leq x$ and $(a, t_1) \in N$ or $((b, t_1) \in N$ and $a = b)$ for $a \in A$. There are two cases:

$$WO\,\frac{(a, t_1) \qquad t_1 \leq x}{(a, x)} \qquad\qquad \frac{EQI\,\dfrac{(b, t_1) \qquad a = b}{(a, t_1)} \qquad t_1 \leq x}{(a, x)}\,WO$$

Thus, $x \in derive^{\mathcal{B}}_{II}(N, a)$ and then $x \in derive^{\mathcal{B}}_{II}(N, A)$. □

Example 3.2. *For the conditionals $N = \{(1, g), (g, t)\}$ and the input set $A = \{\}$ we have $out^{\mathcal{B}}_{II}(N, A) = \{\}$, and for the input set $C = \{g\}$ we have $out^{\mathcal{B}}_{II}(N, C) = Up(t)$.*

Simple-minded Boolean I/O operation

Definition 3.15 (Semantics)**.** Given a Boolean algebra $\mathcal{B}$, a normative system $N \subseteq Ter(B) \times Ter(B)$ and an input set $A \subseteq Ter(B)$, we define the simple-minded Boolean operation as follows:

$$out^{\mathcal{B}}_1(N, A) = Up(N(Up(A)))$$

We put $out^{\mathcal{B}}_1(N) = \{(A, x) : x \in out^{\mathcal{B}}_1(N, A)\}$.

Definition 3.16 (Proof system)**.** Given a Boolean algebra $\mathcal{B}$ and a normative system $N \subseteq Ter(B) \times Ter(B)$, we define $(a, x) \in derive^{\mathcal{B}}_1(N)$ if and only if (a, x) is derivable from N using the rules $\{SI, WO\}$.

Given a set of $A \subseteq Ter(B)$, $(A, x) \in derive^{\mathcal{B}}_1(N)$ whenever $(a, x) \in derive^{\mathcal{B}}_1(N)$ for some $a \in A$. Put $derive^{\mathcal{B}}_1(N, A) = \{x : (A, x) \in derive^{\mathcal{B}}_1(N)\}$.

Theorem 3.17 (Soundness)**.** *$out^{\mathcal{B}}_1(N)$ validates SI and WO.*

Proof. SI: We need to show that

$$\text{SI}\,\frac{x \in Up(N(Up(a))) \qquad b \leq a}{x \in Up(N(Up(b)))}$$

Since $b \leq a$ we have $Up(a) \subseteq Up(b)$. Hence, $N(Up(a)) \subseteq N(Up(b))$ and therefore $Up(N(Up(a))) \subseteq Up(N(Up(b)))$.

WO: We need to show that

$$\text{WO}\,\frac{x \in Up(N(Up(a))) \qquad x \leq y}{y \in Up(N(Up(a)))}$$

Since $Up(N(Up(a)))$ is upward-closed and $x \leq y$ we have $y \in Up(N(Up(a)))$.

□

Counter-example for AND: We can show that AND is not valid.

$$\text{AND}\,\frac{(a,x) \qquad (a,y)}{(a, x \wedge y)}$$

Consider the normative system $N = \{(a,x),(a,y)\}$ we have $x \in Up(N(Up(\{a\})))$ and $y \in Up(N(Up(\{a\})))$ but $x \wedge y \notin Up(N(Up(\{a\})))$ by definition of $Up(X)$.

Theorem 3.18 (Completeness). $out_1^{\mathcal{B}}(N) \subseteq derive_1^{\mathcal{B}}(N)$.

Proof. We show that if $x \in Up(N(Up(A)))$, then $(A,x) \in derive_1^{\mathcal{B}}(N)$. Suppose $x \in Up(N(Up(A)))$, then there is y_1 such that $y_1 \in N(Up(A))$, $y_1 \leq x$, and there is t_1 such that $(t_1,y_1) \in N$ and $a \leq t_1$ for $a \in A$.

$$SI\,\frac{a \leq t_1 \qquad WO\,\dfrac{(t_1,y_1) \quad y_1 \leq x}{(t_1,x)}}{(a,x)}$$

Thus, $x \in derive_1^{\mathcal{B}}(N,a)$ and then $x \in derive_1^{\mathcal{B}}(N,A)$. □

Example 3.3. *For the conditionals* $N = \{(g,t),(\neg g,\neg t),(a,b)\}$ *and the input set* $A = \{g,\neg g\}$ *we have* $out_1^{\mathcal{B}}(N,A) = Up(t,\neg t)$.

Basic Boolean I/O operation

Definition 3.19 (Saturated set)**.** A set V is saturated in a Boolean algebra $\mathcal{B}$ iff

- If $a \in V$ and $b \geq a$, then $b \in V$;
- If $a \vee b \in V$, then $a \in V$ or $b \in V$.

Definition 3.20 (Semantics)**.** Given a Boolean algebra $\mathcal{B}$, a normative system $N \subseteq Ter(B) \times Ter(B)$ and an input set $A \subseteq Ter(B)$, we define the basic Boolean operation as follows:

$$out_2^{\mathcal{B}}(N, A) = \bigcap\{Up(N(V)), A \subseteq V, V \text{is saturated}\}$$

We put $out_2^{\mathcal{B}}(N) = \{(A, x) : x \in out_2^{\mathcal{B}}(N, A)\}$.

Definition 3.21 (Proof system)**.** Given a Boolean algebra $\mathcal{B}$ and a normative system $N \subseteq Ter(B) \times Ter(B)$, we define $(a, x) \in derive_2^{\mathcal{B}}(N)$ if and only if (a, x) is derivable from N using the rules of $derive_1^{\mathcal{B}}(N)$ along with OR.

$$\text{OR}\,\frac{(a, x) \qquad\quad (b, x)}{(a \vee b, x)}$$

Given a set of $A \subseteq Ter(B)$, $(A, x) \in derive_2^{\mathcal{B}}(N)$ if $(a, x) \in derive_2^{\mathcal{B}}(N)$ for some $a \in A$. Put $derive_2^{\mathcal{B}}(N, A) = \{x : (A, x) \in derive_2^{\mathcal{B}}(N)\}$.

Theorem 3.22 (Soundness)**.** *$out_2^{\mathcal{B}}(N)$ validates SI, WO and OR.*

Proof. OR: We need to show that

$$\text{OR}\,\frac{x \in out_2^{\mathcal{B}}(N, \{a\}) \qquad\quad x \in out_2^{\mathcal{B}}(N, \{b\})}{x \in out_2^{\mathcal{B}}(N, \{a \vee b\})}$$

Suppose $\{a \vee b\} \subseteq V$, since V is saturated we have $a \in V$ or $b \in V$. Suppose $a \in V$, in this case since $out_2^{\mathcal{B}}(N, \{a\}) \subseteq Up(N(V))$ we have $x \in out_2^{\mathcal{B}}(N, \{a \vee b\})$.

□

Theorem 3.23 (Completeness)**.** *$out_2^{\mathcal{B}}(N) \subseteq derive_2^{\mathcal{B}}(N)$.*

Proof. Suppose $x \notin derive_2^{\mathcal{B}}(N, A)$, then by monotony of derivability operation there is a maximal set V such that $A \subseteq V$ and $x \notin derive_2^{\mathcal{B}}(N, V)$. V is saturated because

(a) Suppose $a \in V$ and $a \leq b$, by definition of V we have $(a, x) \notin derive_2^{\mathcal{B}}(N)$. We need to show that $x \notin derive_2^{\mathcal{B}}(N, b)$ and since V is maximal we have $b \in V$. Suppose $(b, x) \in derive_2^{\mathcal{B}}(N)$. We have

$$\text{SI}\,\frac{(b,x) \qquad a \leq b}{(a,x)}$$

That is contradiction with $(a, x) \notin derive_2^{\mathcal{B}}(N)$.

(b) Suppose $a \vee b \in V$, by definition of V we have $x \notin derive_2^{\mathcal{B}}(N, a \vee b)$. We need to show that $x \notin derive_2^{\mathcal{B}}(N, a)$ or $x \notin derive_2^{\mathcal{B}}(N, b)$. Suppose $x \in derive_2^{\mathcal{B}}(N, a)$ and $x \in derive_2^{\mathcal{B}}(N, b)$, then we have

$$OR\,\frac{(a,x) \qquad (b,x)}{(a \vee b, x)}$$

That is contradiction with $x \notin derive_2^{\mathcal{B}}(N, a \vee b)$.

Therefore, we have $x \notin Up(N(V))$ (that is equal to $x \notin out_1^{\mathcal{B}}(N, V)$) and so $x \notin out_2^{\mathcal{B}}(N, A)$. □

Reusable Boolean I/O operation

Definition 3.24 (Semantics)**.** Given a Boolean algebra $\mathcal{B}$, a normative system $N \subseteq Ter(B) \times Ter(B)$ and an input set $A \subseteq Ter(B)$, we define the reusable Boolean operation as follows:

$$out_3^{\mathcal{B}}(N, A) = \bigcap\{Up(N(V)), A \subseteq V = Up(V) \supseteq N(V)\}$$

We put $out_3^{\mathcal{B}}(N) = \{(A, x) : x \in out_3^{\mathcal{B}}(N, A)\}$.

Definition 3.25 (Proof system)**.** Given a Boolean algebra $\mathcal{B}$ and a normative system $N \subseteq Ter(B) \times Ter(B)$, we define $(a, x) \in derive_3^{\mathcal{B}}(N)$ if and only if (a, x) is derivable from N using the rules of $derive_1^{\mathcal{B}}(N)$ along with T.[15]

[15] T stands for transitivity.

$$\mathrm{T}\,\frac{(a,x) \qquad (x,y)}{(a,y)}$$

Given a set of $A \subseteq Ter(B)$, $(A,x) \in derive_3^{\mathcal{B}}(N)$ if $(a,x) \in derive_3^{\mathcal{B}}(N)$ for some $a \in A$. Put $derive_3^{\mathcal{B}}(N,A) = \{x : (A,x) \in derive_3^{\mathcal{B}}(N)\}$.

Theorem 3.26 (Soundness). *$out_3^{\mathcal{B}}(N)$ validates SI, WO and T.*

Proof. T: We need to show that

$$\mathrm{T}\,\frac{x \in out_3^{\mathcal{B}}(N,\{a\}) \qquad y \in out_3^{\mathcal{B}}(N,\{x\})}{y \in out_3^{\mathcal{B}}(N,\{a\})}$$

Suppose that X is the smallest set such that $\{a\} \subseteq X = Up(X) \supseteq N(X)$. Since $x \in out_3^{\mathcal{B}}(N,\{a\})$ we have $x \in X$ and from $y \in out_3^{\mathcal{B}}(N,\{x\})$ we have $y \in X$. Thus, $y \in out_3^{\mathcal{B}}(N,\{a\})$.

□

Theorem 3.27 (Completeness). *$out_3^{\mathcal{B}}(N) \subseteq derive_3^{\mathcal{B}}(N)$.*

Proof. Suppose $x \notin derive_3^{\mathcal{B}}(N,a)$, we need to find B such that $a \in B = Up(B) \supseteq N(B)$ and $x \notin Up(N(B))$. Put $B = Up(\{a\} \cup derive_3^{\mathcal{B}}(N,a))$. We show that $N(B) \subseteq B$. Suppose $y \in N(B)$, then there is $b \in B$ such that $(b,y) \in N$. We show that $y \in B$. Since $b \in B$ there are two cases:

- $b \geq a$: in this case we have $(a,y) \in derive_3^{\mathcal{B}}(N)$ since $(b,y) \in derive_3^{\mathcal{B}}(N)$ and we have

$$SI\,\frac{(b,y) \qquad a \leq b}{(a,y)}$$

- $\exists z \in derive_3^{\mathcal{B}}(N,a), b \geq z$: in this case we have

$$T\,\frac{(a,z) \qquad SI\,\dfrac{(b,y) \qquad z \leq b}{(z,y)}}{(a,y)}$$

We only need to show that $x \notin Up(N(B)) = out_1^{\mathcal{B}}(N, \{a\} \cup derive_3^{\mathcal{B}}(N,a))$. Suppose $x \in Up(N(B))$, then there is y_1 such that $x \geq y_1$ and $\exists t_1, (t_1,y_1) \in N$ and $t_1 \in Up(\{a\} \cup derive_3^{\mathcal{B}}(N,a))$. There are two cases:

- $t_1 \geq a$: in this case we have

$$WO\,\dfrac{SI\,\dfrac{(t_1, y_1) \qquad a \leq t_1}{(a, y_1)} \qquad y_1 \leq x}{(a, x)}$$

- $\exists z_1 \in derive_3^{\mathcal{B}}(N, a), z_1 \leq t_1$: in this case we have

$$WO\,\dfrac{T\,\dfrac{(a, z_1) \qquad SI\,\dfrac{(t_1, y_1) \qquad z_1 \leq t_1}{(z_1, y_1)}}{(a, y_1)} \qquad y_1 \leq x}{(a, x)}$$

Thus, in both cases $(a, x) \in derive_3^{\mathcal{B}}(N)$ and then $x \in derive_3^{\mathcal{B}}(N, a)$ that is contradiction. □

Example 3.4. *For the conditionals* $N = \{(1, g), (g, t), (\neg g, \neg t), (a, b)\}$ *and the input set* $A = \{\neg g\}$ *we have* $out_3^{\mathcal{B}}(N, A) = Up(g, t, \neg t)$.

3.3 Obligatory Norms: Input/Output Operations

In this section, we add the rule AND and cumulative transitivity (CT) to our introduced derivation systems. We aim to rebuild the derivation systems introduced by Makinson and van der Torre [140] for deriving obligations.

Definition 3.28 (Proof system)**.** Given a Boolean algebra $\mathcal{B}$ and a normative system $N \subseteq Ter(B) \times Ter(B)$, we define $(a, x) \in derive_i^X(N)$ if and only if (a, x) is derivable from N using $EQO, EQI, SI, WO, OR, AND, CT$ as follows:

$derive_i^X$	Rules
$derive_{II}^{AND}$	{WO, EQI, AND}
$derive_1^{AND}$	{SI, WO, AND}
$derive_2^{AND}$	{SI, WO, OR, AND}
$derive_I^{CT}$	{SI, EQO, CT}
$derive_{II}^{CT}$	{WO, EQI, CT}
$derive_1^{CT}$	{SI, WO, CT}
$derive_1^{CT,AND}$	{SI, WO, CT, AND}

$$\text{AND}\,\dfrac{(a, x) \qquad (a, y)}{(a, x \wedge y)}$$

$$\text{CT}\,\dfrac{(a, x) \qquad (a \wedge x, y)}{(a, y)}$$

Given a set of $A \subseteq Ter(B)$, $(A, x) \in derive_i^X(N)$ whenever $(a, x) \in derive_i^X(N)$ for some $a \in A$. Put $derive_i^X(N, A) = \{x : (A, x) \in derive_i^X(N)\}$.

Definition 3.29 (Semantics out_i^{AND})**.** Given a Boolean algebra $\mathcal{B}$, a normative system $N \subseteq Ter(B) \times Ter(B)$ and an input set $A \subseteq Ter(B)$, we define the AND operation as follows:

$$\begin{array}{ll}
out_i^{AND^0}(N,A) & = out_i^{\mathcal{B}}(N,A) \\
out_i^{AND^{n+1}}(N,A) & = out_i^{AND^n}(N,A) \cup \\
 & \{y \wedge z : y, z \in out_i^{AND^n}(N,\{a\}),\ \ a \in A\} \\
out_i^{AND}(N,A) & = \bigcup_{n \in N} out_i^{AND^n}(N,A)
\end{array}$$

We put $out_i^{AND}(N) = \{(A,x) : x \in out_i^{AND}(N,A)\}$.

Definition 3.30 (Semantics out_i^{CT})**.** Given a Boolean algebra $\mathcal{B}$, a normative system $N \subseteq Ter(B) \times Ter(B)$ and an input set $A \subseteq Ter(B)$, we define the CT operation as follows:

$$\begin{array}{ll}
out_i^{CT^0}(N,A) & = out_i^{\mathcal{B}}(N,A) \\
out_i^{CT^{n+1}}(N,A) & = out_i^{CT^n}(N,A) \cup \\
 & \{x : y \in out_i^{CT^n}(N,\{a\}) \,\text{and}\, x \in out_i^{CT^n}(N,\{a \wedge y\}),\ \ a \in A\} \\
out_i^{CT}(N,A) & = \bigcup_{n \in N} out_i^{CT^n}(N,A)
\end{array}$$

We put $out_i^{CT}(N) = \{(A,x) : x \in out_i^{CT}(N,A)\}$.

Definition 3.31 (Semantics $out_i^{CT,AND}$)**.** Given a Boolean algebra $\mathcal{B}$, a normative system $N \subseteq Ter(B) \times Ter(B)$ and an input set $A \subseteq Ter(B)$, we define the CT,AND operation as follows:

$$\begin{array}{ll}
out_i^{CT,AND^0}(N,A) & = out_i^{CT}(N,A) \\
out_i^{CT,AND^{n+1}}(N,A) & = out_i^{CT,AND^n}(N,A) \cup \\
 & \{y \wedge z : y, z \in out_i^{CT,AND^n}(N,\{a\}),\ \ a \in A\} \\
out_i^{CT,AND}(N,A) & = \bigcup_{n \in N} out_i^{CT,AND^n}(N,A)
\end{array}$$

We put $out_i^{CT,AND}(N) = \{(A,x) : x \in out_i^{CT,AND}(N,A)\}$.

Theorem 3.32. *Given a Boolean algebra $\mathcal{B}$, for every normative system $N \subseteq Ter(B) \times Ter(B)$ we have $out_i^{AND}(N) = derive_i^{AND}(N)$, $i \in \{II,1,2\}$; $out_i^{CT}(N) = derive_i^{CT}(N)$, $i \in \{I,II,1\}$ and $out_1^{CT,AND}(N) = derive_1^{CT,AND}(N)$.*

Proof. The proof is based on the reversibility of inference rules. Makinson and van der Torre [140] studied the reversibility of inference rules.

Lemma 3.33. *Let D be any derivation using at most EQI, SI, WO, OR, AND, CT; then there is a derivation $D^{'}$ of the same root from a subset of leaves, that applies AND only at the end.*

Proof. See Observation 18 [140].

The main point of the observation is that we can reverse the order of rules AND, WO to WO, AND; AND, SI to SI, AND; AND, OR to OR, AND and finally AND, CT to SI, CT or CT, AND. Also, we can reverse the order of rules AND and EQI as follows:

$$\text{EQI}\,\dfrac{\text{AND}\,\dfrac{(a,x) \qquad (a,y)}{(a,x\wedge y)} \qquad a=b}{(b,x\wedge y)} \qquad\qquad \text{AND}\,\dfrac{\text{EQI}\,\dfrac{(a,x)\ \ a=b}{(b,x)} \qquad \text{EQI}\,\dfrac{(a,y)\ \ a=b}{(b,y)}}{(b,x\wedge y)}$$

□

Hence, in each system of $\{WO, EQI, AND\}$, $\{SI, WO, AND\}$ and $\{SI, WO, OR, AND\}$ we can apply AND rule just at the end. Thus, we can characterize $deriv_i^{AND}(N)$ using the fact $deriv_i^{\mathcal{B}}(N) = out_i^{\mathcal{B}}(N)$ and the iteration of AND.

It is easy to check that we can reverse CT with SI, EQO, WO, and EQI, by this fact similarly we can characterize $deriv_i^{CT}(N)$.

Finally, since AND can reverse with ST, WO and CT, we can characterize $deriv_1^{CT,AND}(N)$ by applying iteration of AND over $out_1^{CT}(N)$ that means $out_1^{CT,AND}(N)$.

□

Similarly, we can define $out_i^{OR}(N)$ operation and characterize some other proof systems :

$derive_i^X$	Rules
$derive_I^{OR}$	{SI, EQO, OR}
$derive_I^{CT,OR}$	{SI, EQO, CT, OR}
$derive_1^{CT,OR}$	{SI, WO, CT, OR}
$derive_1^{CT,OR,AND}$	{SI, WO, CT, OR, AND}

Four systems $derive_1^{AND}$, $derive_2^{AND}$ (or $derive_1^{OR,AND}$), $derive_1^{CT,AND}$ and $derive_1^{CT,OR,AND}$ are introduced by Makinson and van der Torre [140] for reasoning about obligatory norms.

3.3.1 The miners paradox

To illustrate the advantage and difference of the new proposed semantics with the classic semantics, we focus on the miners paradox. The miners paradox is discussed by Kolodny and MacFarlane [126].

> Ten miners are trapped either in shaft A or in shaft B, but we do not know which. Flood waters threaten to flood the shafts. We have enough sandbags to block one shaft, but not both. If we block one shaft, all the water will go into the other shaft, killing any miners inside it. If we block neither shaft, both shafts will fill halfway with water, and just one miner, the lowest in the shaft, will be killed.

So, in our deliberation, it seems that the followings are true:

1. Either the miners are in shaft A or in shaft B.
2. If the miners are in shaft A, we should block shaft A.
3. If the miners are in shaft B, we should block shaft B.
4. We should block neither shaft.

In principle, it would be best to save all the miners by blocking the right shaft, sentence 2, and 3. Sentence 4 is correct since there is a fifty-fifty chance of blocking the right shaft, given that we do not know the shaft in which the miners are. This sentence guarantees that we save nine of the ten miners according to the scenario [218]. The paradox is: the four sentences jointly are inconsistent in classical logic. Moreover, they are inconsistent in the context of the Kratzerian baseline view [65].

> "The problem is that the ordering of worlds should be sensitive to the shifts that are introduced by conditional supposition. It may be unconditionally best to block neither shaft. But if the miners are in shaft A, it will be best to block A; and if they are in shaft B it will be best to block B. This is not allowed by the baseline algorithm: if w is the best world in some initial information state, then it remains best under any supposition P that is true at w." (Cariani [65])

Here, we analyze this paradox in our setting. Suppose that the set of norms $N = \{(shA, blA), (shB, blB), (\top, \neg blA \wedge \neg blB)\}$ represents the sentences 2–4. We choose one of the output operations for deriving obligation, which satisfies the rule of SI. There are two ways for representing sentence 1. For the case $f(w) = \{shA \vee shB\}$, as a set of factual informations, we have $\neg blA \wedge \neg blB \in out(N^O, \{shA \vee shB\})$. There is another way for representing sentence 1 by means of the ordering source. In the case of $g(w) = \{shA, shB\}$, as a set of possible inconsistent informations, we have $blA, blB, \neg blA \wedge \neg blB \in out(N^O, \{shA, shB\})$. Kolodny and MacFarlane [126] introduced a semantics for deontic modals based on referring to a set of information to analyze miners example. In their approach, modus pones is invalid, but our approach is based on detachment or modus pones.

Moreover, by updating the information, if $f(w) = \{shA\}$ then we have $blA, \neg blA \wedge \neg blB \in out(N^O, \{shA\})$. In this case, we know that miners are in shaft A and consider $C = \{blA\}$ as the constraint; by using constrained I/O logic we have $blA \in out_c(N^O, \{shA\})$, see 4.1 for the required definitions. The constraint here gives an ordering over norms, a distinctive feature for I/O framework over classical semantics. In Chapter 4 we give a semantical characterization for these constraints.

3.4 I/O Mechanism over Abstract Logics

An abstract logic [87] is a pair $\mathcal{A} = \langle \mathcal{L}, C \rangle$ where $\mathcal{L} = \langle L, ... \rangle$ is an algebra and C is a closure operator defined on the power set of its universe, that means for all $A, B \subseteq L$:

- $A \subseteq C(A)$
- $A \subseteq B \Rightarrow C(A) \subseteq C(B)$
- $C(A) = C(C(A))$

The elements of an abstract logic can be ordered as $a \leq b$ if and only if $b \in C(\{a\})$.[16] Without loss of generality, we work with the algebra of formulas (or terms in algebraic context) $\mathbf{Fm}(X) = \langle Fm(X), ... \rangle$ for a set of fixed variable X. Similar to Boolean algebras, we can define Eq and Up operators for $A \subseteq Fm(X)$.

[16] $a = b$ if and only if $a \leq b$ and $b \leq a$.

Definition 3.34 (Semantics)**.** Given an abstract logic $\mathcal{A} = \langle \mathbf{Fm}(X), C\rangle$, a normative system $N \subseteq Fm(X) \times Fm(X)$ and an input set $A \subseteq Fm(X)$, we define the I/O operations as follows:

- $out_0^{\mathcal{A}}(N, A) = Eq(N(Eq(A)))$
- $out_I^{\mathcal{A}}(N, A) = Eq(N(Up(A)))$
- $out_{II}^{\mathcal{A}}(N, A) = Up(N(Eq(A)))$
- $out_1^{\mathcal{A}}(N, A) = Up(N(Up(A)))$
- $out_2^{\mathcal{A}}(N, A) = \bigcap\{Up(N(V)), A \subseteq V, V \text{is saturated}\}$[17]
- $out_3^{\mathcal{A}}(N, A) = \bigcap\{Up(N(V)), A \subseteq V = Up(V) \supseteq N(V)\}$

We put $out_i^{\mathcal{A}}(N) = \{(A, x) : x \in out_i^{\mathcal{A}}(N, A)\}$.

Definition 3.35 (Proof system)**.** Given an abstract logic $\mathcal{A} = \langle \mathbf{Fm}(X), C\rangle$ and a normative system $N \subseteq Fm(X) \times Fm(X)$, we define $(a, x) \in derive_0^{\mathcal{A}}(N)$ ($derive_I^{\mathcal{A}}(N)$, $derive_{II}^{\mathcal{A}}(N)$, $derive_1^{\mathcal{A}}(N)$, $derive_2^{\mathcal{A}}(N)$, $derive_3^{\mathcal{A}}(N)$) if and only if (a, x) is derivable from N using the rules $\{EQI, EQO\}$ ($\{SI, EQO\}$, $\{WO, EQI\}$, $\{SI, WO\}$, $\{SI, WO, OR\}$, $\{SI, WO, T\}$). Put $derive_i^{\mathcal{A}}(N, A) = \{x : (A, x) \in derive_i^{\mathcal{A}}(N)\}$.

Theorem 3.36 (Soundness and Completeness)**.** $out_i^{\mathcal{A}}(N) = derive_i^{\mathcal{A}}(N)$.

Proof. The proofs are same as soundness and completeness theorems in Section 3.2. □

A logical system $\mathbf{L} = \langle L, \vdash_{\mathbf{L}}\rangle$ straightforwardly provides an equivalent abstract logic $\langle \mathbf{Fm}_L, C_{\vdash_L}\rangle$. Therefore, we can build I/O framework over different types of logic include first-order logic, simple type theory, description logic, different kinds of modal logics that are expressive for the intentional concepts such as belief and time.

Example 3.5. *In modal logic system KT, for the conditionals* $N = \{(p, \Box q), (q, r), (s, t)\}$ *and input set* $A = \{p\}$ *we have* $out_3^{KT}(N, A) = Up(\Box q, r)$*.*

Moreover, we can add other rules such as AND and CT to the systems same as Section 3.3. There is a similar result for building the simple-minded I/O operation over Tarskian consequence relations [68] (see the discussion about abstract input/output logic [199]).

[17] For this case, the abstract logic $\mathcal{A} = \langle \mathbf{Fm}(X), C\rangle$ should include $\vee$, that is a binary operation symbol, either primitive or defined by a term and we have $a \vee b, b \vee a \in C(\{a\})$ ($\vee$-Introduction) and if $c \in C(\{a\}) \cap C(\{b\})$ then $c \in C(a \vee b), C(b \vee a)$ ($\vee$-Elimination). Saturated sets are defined similarly, see Definition 3.19.

3.4.1 Nested I/O operations

Theorem 3.37. *Every $out_i^{\mathcal{B}}$, and $out_i^{\mathcal{A}}$ operation is a closure operator.*

Proof. We just look at I/O operations over Boolean algebras, the argument for abstract logics is similar. We need to show that

- $N \subseteq out_i^{\mathcal{B}}(N)$
- $N \subseteq M \Rightarrow out_i^{\mathcal{B}}(N) \subseteq out_i^{\mathcal{B}}(M)$
- $out_i^{\mathcal{B}}(N) = out_i^{\mathcal{B}}(out_i^{\mathcal{B}}(N))$

By soundness and completeness theorems we know that $out_i^{\mathcal{B}}(N) = derive_i^{\mathcal{B}}(N)$. So we study $derive_i^{\mathcal{B}}(N)$ that is more simple than $out_i^{\mathcal{B}}(N)$. The first two properties are clear by definition of $derive_i^{\mathcal{B}}$. For the last one, we need to show that $derive_i^{\mathcal{B}}(N) = derive_i^{\mathcal{B}}(derive_i^{\mathcal{B}}(N))$. We have $derive_i^{\mathcal{B}}(derive_i^{\mathcal{B}}(N)) = derive_i^{\mathcal{B}}(\{(A,x)|(a,x) \in derive_i^{\mathcal{B}}(N)$ for some $a \in A\}) = \{(A,x)|(a,x) \in derive_i^{\mathcal{B}}(N)$ for some $a \in A\}$ since $N \subseteq \{(A,x)|(a,x) \in derive_i^{\mathcal{B}}(N)$ for some $a \in A\}$ and same rules apply over $derive_i^{\mathcal{B}}(N)$. Actually we need to show that if $(a,x) \in derive_i^{\mathcal{B}}(N)$ then $derive_i^{\mathcal{B}}(N) = derive_i^{\mathcal{B}}(N \cup \{(a,x)\})$ that is hold for $derive_i^{\mathcal{B}}$.

□

Since $out_i^{\mathcal{B}}$ is a closure operator so we can define $out_j^{\mathcal{A}}(M, out_i^{\mathcal{B}}(N))$ where $N \subseteq Ter(B) \times Ter(B)$, $M \subseteq (Ter(B) \times Ter(B)) \times (Ter(B) \times Ter(B))$ and in the abstract logic $\mathcal{A}$ we have $L = N \times N$ and $C = out_i^{\mathcal{B}}$. The corresponding characterization is $derive_j^{\mathcal{A}}(M, derive_i^{\mathcal{B}}(N))$. Similarly, we can define nested $out_j^{\mathcal{A}}(M, out_i^{\mathcal{A}}(N))$ operations.

3.5 Conclusion

In summary, we have characterized a class of proof systems over Boolean algebras for a set of explicitly given norms as follows:

$derive_i^{\mathcal{B}}$	Rules
$derive_R^{\mathcal{B}}$	{EQO}
$derive_L^{\mathcal{B}}$	{EQI}
$derive_0^{\mathcal{B}}$	{EQI, EQO}
$derive_I^{\mathcal{B}}$	{SI, EQO}
$derive_{II}^{\mathcal{B}}$	{WO, EQI}
$derive_1^{\mathcal{B}}$	{SI, WO}
$derive_2^{\mathcal{B}}$	{SI, WO, OR}
$derive_3^{\mathcal{B}}$	{SI, WO, T}

$$\text{EQO}\ \frac{(a,x) \qquad x=y}{(a,y)} \qquad \text{WO}\ \frac{(a,x) \qquad x \leq y}{(a,y)}$$

$$\text{EQI}\ \frac{(a,x) \qquad a=b}{(b,x)} \qquad \text{OR}\ \frac{(a,x) \qquad (b,x)}{(a \vee b,x)}$$

$$\text{SI}\ \frac{(a,x) \qquad b \leq a}{(b,x)} \qquad \text{T}\ \frac{(a,x) \qquad (x,y)}{(a,y)}$$

Each proof system is sound and complete for an I/O operation. For each of the introduced I/O operations, we can define a new I/O operation version that allows input reappear as outputs. Let $N^+ = N \cup I$, where I is the set of all pairs (a,a) for $a \in Ter(B)$. We define $out_i^{\mathcal{B}+}(N,A) = out_i^{\mathcal{B}}(N^+,A)$, the characterization is same as $out_i^{\mathcal{B}}$. It is interesting to compare the introduced systems with minimal deontic logics [71, 102], and its similar approaches, such as [76], that do not have deontic aggregation principles. In Section 4.2, we study the neighborhood semantics of input/output operations. Moreover, in this chapter we have shown that how we can add two rules AND and CT to the proof systems and find representation theorems for them.

$derive_i^{X}$	Rules
$derive_{II}^{AND}$	{WO, EQI, AND}
$derive_1^{AND}$	{SI, WO, AND}
$derive_2^{AND}$	{SI, WO, OR, AND}
$derive_I^{CT}$	{SI, EQO, CT}
$derive_{II}^{CT}$	{WO, EQI, CT}
$derive_1^{CT}$	{SI, WO, CT}
$derive_1^{CT,AND}$	{SI, WO, CT, AND}
$derive_I^{OR}$	{SI, EQO, OR}
$derive_I^{CT,OR}$	{SI, EQO, CT, OR}
$derive_1^{CT,OR}$	{SI, WO, CT, OR}
$derive_1^{CT,OR,AND}$	{SI, WO, CT, OR, AND}

$$\text{AND}\ \frac{(a,x) \qquad (a,y)}{(a,x \wedge y)}$$

$$\text{CT}\ \frac{(a,x) \qquad (a \wedge x,y)}{(a,y)}$$

The input/output logic is inspired by a view of logic as "secretarial assistant", to prepare inputs before they go into the motor engine and unpacking outputs, rather than logic as an "inference motor". The only input-output logics investigated so far in the literature are built on top of classical propositional logic and intuitionist propositional logic. The algebraic construction shows how we can build the input/output version of any abstract logic. Furthermore, we can build I/O framework over posts $\langle A, \leq \rangle$ where A is a set and $\leq$ is a reflexive, antisymmetric and transitive binary relation. The monotony property of closure has no rule in the proofs and we can build I/O operations over non-monotonic relations. We pose combining I/O operations with consequence relations that do not satisfy inclusion or idempotence property [139] as a further research question. For example, combining the original I/O framework with a consequence relation that does not satisfy inclusion where the consequence relation is an input/output closure ($A \in out(N, A)$ is not necessary) was explored by Sun and van der Torre [201].

Future research could investigate whether the proposed I/O frameworks already supports non-trivial applications in AI. We could consider the proposed I/O frameworks as the rule-based systems that are knowledge-base friendly in the sense that could deal with inconsistent information and could be applied over different kinds of knowledge bases, represented as set of logical formulas. Moreover, we could present an embedding of new I/O operations in HOL [31, 28, 29], see Chapter 5.

Finally, we have introduced a unification of the classic and norm-based semantics for reasoning about permission and obligation; it is worth to investigate exploring the philosophical and conceptual advantages of integrating the norm-based semantics into the classic semantics [218, 112]. The introduced semantics is based on the interaction between normative reasoning and informational and motivational modalities. There is more complexity in motivational modalities comparing to the informational one. For example, there is a hierarchy in our intentions [58]. How can we represent this hierarchy in the proof system and semantics of I/O systems? As an instance we can look at Anankastic conditionals.[18] The Anankastic conditional "*If want* ψ, *should* φ" is interpreted as a best-means-of relation between ψ and φ. In addition, we can use the introduced I/O logical systems for other conceptual reasoning. There are other domains where dropping reflexivity from logical consequence relation is appropriate such as causality [47] and credulous reasoning along dropping AND rule for belief change [48].

[18] For a full discussion see the ESSLLI 2016 course: Deontic modality: Linguistic and Logical Perspectives on Oughts and Ends.

Chapter 4

Norms, Preferences, and Contrary-to-Duty

The so-called dyadic deontic logic approach developed mainly by Rescher [178], von Wright [223], Danielsson [80], Hansson [109], van Fraassen [216, 217], Lewis [132], Spohn [188], Åqvist [9, 10, 12], Goble [100, 101], and Parent [158, 159, 161] use a primitive binary conditional obligation operator. The dyadic deontic logic approach was motivated by the well-known paradoxes of *contrary-to-duty* (CTD), see Subsection 2.2.3. The approach faces outstanding problems such as detecting *deontic dilemmas* [174, 57, 213] and *violation detection* [210, 166]. Against conditional obligations [57, 205, 206, 207], more sophisticated proposals suggested by synthesizing monadic deontic logic and a conditional theory [149, 155, 57, 213, 214, 181]. The first proposal was given by Mott [149]:

> "I wish to argue that the problems which motivated dyadic deontic logic can be adequately resolved in monadic deontic logic with a stronger conditional than material implication. I shall conclude that the conditional required is that developed by David Lewis..." (Mott [149])

It has suggested that the theory of conditional obligations has two components and is not anymore primitive, as Thomason [204]: "A proper theory of conditional obligation will be the product of two separate components: a theory of the conditional and a theory of obligation." Input/output logic is developed by Makinson and van der Torre for reasoning about (monadic) obligations and permissions [140, 142]. It is promising to combine conditional logic and input/output logic for reasoning about conditional obligation and permission. Moreover, it is open whether by using a conditional theory, obtained by a preference relation, inside input/output logic, it can support contrary-to-duty scenarios. Constrained I/O logic [141] was introduced for reasoning about contrary-to-duty problems. There are syntactical ([141], Section 6) and proof theoretical ([198], Section 3) characterizations for constrained I/O logic. Here, constraints are preferences. In this sense, we present a semantical characterization for constrained I/O logic.

The question of this chapter is: *How can we integrate input/output logic with Hansson and Lewis's conditional theory for building a new compositional theory about conditional deontic modals?* We propose to combine input/output logic [140] as a logical theory about deontic modals with Hansson and Lewis's conditional logic [109, 132]. The semantical construct of conditional logics is defined in terms of possible worlds models by seminal works due to Lewis [131], Stalnaker [189, 190], Pollock [171], Chellas [70], and Burgess [64], among others. The logics comprise a notion of preference or choice among worlds: a conditional $\varphi > \psi$ is true at a world w if ψ is true in all the (best) worlds most good/normal/similar to w in which φ is true [153]. Simply, in the actual world, $\varphi > \psi$ holds if the best φ-worlds are ψ-worlds. The various kinds of choosing the best worlds (among worlds) were explored by Parent [160], which was used to define the dyadic deontic obligations [159].

On the other side, Hansson and Lewis's conditional theory [109, 132] is too weak to represent inconsistency in deontic dilemmas [174, 213]. We can combine input/output logic as a non-monotonic defeat mechanism with Hansson and Lewis's conditional theory [109, 132] to detect these dilemmas. Prohairetic deontic logic (PDL) was investigated by van der Torre and Tan [213] to formalize contrary-to-duty conditionals [109, 132] and detect deontic dilemmas. It is based on combining two dyadic deontic operators for each task. Their formalization is in terms of monadic deontic logic and a deontic betterness relation. They take the axiomatization of the underlying modal logic, which provides a uniform semantically framework for the both operators [203, 214, 213]. Our axiomatization comes directly from the chosen conditional logic and the derivation system of input/output logic. It is flexible to add or remove inference rules such as weakening of the output (WO), reasoning by cases (OR), cumulative transitivity (CT), and more interestingly rules with consistency check. In this sense, our approach provides a uniform syntactical way to combine Hansson and Lewis's conditional theory [109, 132] with a non-monotonic mechanism. A problem in most solutions for detecting dilemmas is that the set of formulas $N = \{\bigcirc(\neg\psi/\varphi_1), \bigcirc(\psi/\varphi_2)\}$ is inconsistent or $\varphi_1 \wedge \varphi_2$ is impossible [213]. In our setting the set of formulas N is not necessary inconsistent, given $\{\varphi_1, \varphi_2\}$ consistent. We detect dilemmas by using detachment in the output operations.

Before introducing our compositional theory of conditional obligations and permissions, we study the neighborhood semantics of input/output operations. Then we use the valuation functions from a set of propositional symbols into the class of Boolean algebras for defining our conditional theory. It is a well-known fact that Boolean algebras are the algebraic models of (classical) propositional logic. In Chapter 3, we developed I/O mechanism over every Boolean algebra. So the natural question would be extending algebraic models of propositional logic along with I/O operations over Boolean algebras. We show that the extension of propositional logic with a set of conditional norms is sound and complete respect to the class of Boolean algebras that the corresponding I/O operation holds through all of them. In fact, the valuation functions play the role of possible worlds, and the order over them supports the theory of conditionals. The proposed I/O framework is expressive enough to represent preferential conditionals. Mainly, it is capable of resolving contrary-to-duty paradoxes along requirements suggested by Carmo and Jones [66].

The chapter is structured as follows: Section 4.1 provides a quick review of constrained I/O logic, and presents our suggested solutions for contrary-to-duty statements. Section 4.2 gives a neighborhood characterization of I/O operations. Section 4.3 integrates a conditional theory into input/output logic. Section 4.4 presents a sequent calculus for input/output logic. Section 4.5 and Section 4.6 will extend our results over propositional intuitionistic logic.

4.1 Norms-based Deontic Reasoning and Contrary-to-Duty

First, we review the approach proposed by Makinson and van der Torre [141], and then we introduce two kinds of solutions for dealing with contrary-to-duties.

Original solution by Makinson and van der Torre In the original approach proposed by Makinson and van der Torre [141], the contrary-to-duty problem is resolved according to the maxfamily of norms in which the output is consistent with some constraint.

Definition 4.1 (Constrained I/O logic)**.** Define[1]

- $maxfamily(N, A, C) = \{N^{'} \subseteq N : out(N^{'}, A) \text{ is consistent with } C\}$
- $outfamily(N, A, C) = \{out(N^{'}, A) : N^{'} \in maxfamily(N, A, C)\}$

[1] See Subsection 2.2.2 for more definitions.

- Constrained, net output: $out_c(N, A) = \bigcap / \bigcup outfamily(N, A, C)$

For example, consider the following scenario of contrary-to-duty problem:

(1) Personal data shall be processed lawfully.

(2) If the personal data have been processed unlawfully, the controller has the obligation to erase the personal data in question.

(3) It is obligatory not erase the personal data provided that it is processed lawfully.

(4) Some data has been processed unlawfully.

We can represent above statements as the set of norms $N = \{(\top, pl), (\neg pl, er), (pl, \neg er)\}$ and the input case $A = \{\neg pl\}$. We can reason as follows:

- For out_1 we have $maxfamily(N, A, A) = \{\{(\neg pl, er), (pl, \neg er)\}\}$ and $outfamily(N, A, A) = \{Cn(er)\}$.
- For out_3 we have $maxfamily(N, A, \{\}) = \{\{(\top, pl), (\neg pl, er)\}, \{(\top, pl), (pl, \neg er)\}, \{(\neg pl, er), (pl, \neg er)\}\}$ and $outfamily(N, A, \{\}) = \{Cn(pl, er), Cn(pl, \neg er), Cn(er)\}$.

The $outfamily$ operations in the constrained version of I/O logic [141] can be used for reasoning about contrary-to-duties. There are syntactical ([141], Section 6) and proof theoretical ([198], Section 3) characterizations for constrained I/O logic.

Initial solution: Removing SI We have introduced two basic systems $\{WO, EQI, AND\}$ and $\{WO, EQI, CT\}$, in which the SI rule is absent (see Section 3.3). For the Chisholm set $\{(1, g), (g, t), (\neg g, \neg t)\}$ we just have $Up(\neg t)$ as the output when the input set is $\{\neg g\}$. The main problem with this approach is that we cannot detach primary obligations from the system. For example, if we add more primary obligations to the Chisholm set, such as the moral norm of being kind with your neighbors represented by $(1, k)$. Contrary to our intuition k is not in the output set when the input is $\{\neg g\}$.

Advanced solution: Combining preferential conditionals and normative reasoning Our putative solution is based on a logical system comprised of input/output logic with Hansson and Lewis's conditional theory. So, a conditional obligation is the combination of a unary obligation and the conditional. Simply, we analyze conditional obligation sentences which have the form $a > \bigcirc x$, where $>$ is a classic conditional connective [131]. Given the set of obligatory norms N^O and suppose $(a, x) \in N^O$, we define the new conditionals as follows:

$$a > \bigcirc x \text{ holds iff } (a, x) \in \mathit{derive}_i(N^O) \text{ and } a > x \text{ holds}$$

where $\mathit{derive}_i(N^O)$ is an appropriate derivation system for obligation. In fact, we consider modal translation of $a \rightarrow \bigcirc x$ for $(a, x) \in \mathit{derive}_i(N^O)$. This makes our definition a compositional definition of monadic obligation and conditionals.

For a given set of permissive norms N^P, by choosing a plausible derivation system for permission, $\mathit{derive}_i(N^P)$, similar to the definition of conditional obligation, we can define the conditional permission:

$$a > Px \text{ holds iff } (a, x) \in \mathit{derive}_i(N^P) \text{ and } \neg(a > \neg x) \text{ holds}$$

where $\neg(a > \neg x)$ is the conditional dual of $a > x$. We denote the set of new conditional obligations by derive_i^O and the set of new conditional permissions by derive_i^P. This chapter does not mention subscripts or superscripts of the normative system or derivation systems if it is clear from the context or does not affect on our discussion.

In the contrary-to-duty examples, the focus is on conditional obligations. derive^O (as well derive^P) has two main differences with the original derivation operation. Given the normative system N,

1. derive^O is not reflexive. If $(a, x) \in N$, it is not necessary that $a > \bigcirc x \in \mathit{deriv}^O(N)$. In the original derivation system we have $N \subseteq \mathit{deriv}(N)$.

2. derive^O dose not validate the rule of SI while the *derive* validate it. If $a > \bigcirc x \in \mathit{derive}^O(N)$, it is not necessary that $a \wedge b > \bigcirc x \in \mathit{derive}^O(N)$.

Carmo and Jones [66] formulate several requirements that a plausible solution to the contrary-to-duty paradox should endorse them all.

- (i) **Consistency of the CTD formalisation** and (ii) **Logical independence** Parent and van der Torre [167] consider these two items "self-explanatory" for I/O operations. They are self-explanatory since the derivation system is reflexive $N \subseteq derive(N)$, and the CTD statements are explicitly represented as norms. While, when we integrate I/O operations with a conditional theory and drop the reflexivity property, it would be challenging whether we can derive all the initial norms ($N \nsubseteq derive^O(N)$). However, the same algorithm of ordering by using the number of violations [109] gives a preference such that we can detach all the CTD statements form the corresponding derivation system.

- (iii) **Uniform representation of logical structures for conditional sentences** Parent and van der Torre [167] interpreted this requirement as "**Uniform representation of norms**" for the norm-based deontic reasoning. It seems that this requirement considers a specific theory of conditionals that support both statements, while it was not explored enough which kind of conditionals are supported by I/O operations. In this thesis, we study the neighborhood characterization of I/O operations, and then we compose a conditional theory built on top of a preference relation inside I/O logic. We use the conditional theory for representing CTD statements.

- (iv) **Applicability to (at least apparently) timeless and actionless CTD examples** This item is motivated by Prakken and Sergot [174]. Our formalism is not dependent on action or time.

- (v) **Capacity to derive (ideal and actual) obligations** The semantical construct of I/O operations allows detaching obligations from an input set.

- (vi) **Capacity to avoid the pragmatic oddity** Following Parent and van der Torre [167] we argue that our system is capable of avoiding pragmatic oddity.

- (viii) **Capacity to represent the fact that a violation of an obligation has occurred** We can simply represent violation as pair of $\alpha \in out(N, A)$ and $\neg\alpha \in Up(A)$.

- "**No 'drowning' effect**" and (ix) "**Ability to allow for a certain amount of agglomeration**" are added by Parent and van der Torre [167]. *No 'drowning' effect* is the ability for showing the violated obligations and the agglomeration is the ability to derive the conjoined obligation, for example, it is reasonable that we derive $\bigcirc(\neg f \wedge s)$ from $\bigcirc(f \vee s)$ and $\bigcirc(\neg f)$ [114]. The proposed system satisfies these two requirements.

Pragmatic oddity

The term "pragmatic oddity" refers to a counter-intuitive situation of representing CTD examples in the SDL, introduced by Prakken and Sergot [174].

> "It is a bit odd to say that in all ideal versions of this world you keep your promise and you apologise for not keeping it. This oddity—we might call it a 'pragmatic oddity'—seems to be absent from the natural language version, which means that the SDL representation is not fully adequate." (Prakken and Sergot [174])

Parent and van der Torre [167] discussed that norm-based approach, input/output logic, can avoid the pragmatic oddity by adding a consistency check to the rule of AND for deriving the conjoined obligation. In the I/O framework proposed in Chapter 3 (see Section 3.3), we can use the same method and design derivation systems that avoid the pragmatic oddity. For example, consider the following rule

$$\text{R-AND}\ \frac{(a,x)\quad (a,y)\quad a\wedge x\wedge y\neq 0}{(a,x\wedge y)}$$

We can revise the rule of R-AND (similar to AND) with EQI, WO, SI, OR and (SI, CT), so we could characterize the following systems as before (see Section 3.3):

$derive_i^{R-AND}$	Rules
$derive_{II}^{R-AND}$	{WO, EQI, R-AND}
$derive_1^{R-AND}$	{SI, WO, R-AND}
$derive_2^{R-AND}$	{SI, WO, OR, R-AND}

$derive_i^{CT,R-AND}$	Rules
$derive_1^{CT,R-AND}$	{SI, WO, CT, R-AND}

Example 4.1. *Let* $N=\{(1,k),(\neg k,a)\}$ *and* $A=\{\neg k\}$*, we have* $k\in derive_1^{R-AND}(N,\{\neg k\})$ *and* $a\in derive_1^{R-AND}(N,\{\neg k\})$ *but* $k\wedge a\notin derive_1^{R-AND}(N,\{\neg k\})$*.*

Ross's paradox

In the norm-based approach, we resolve Ross's paradox (deriving obligation of sending a letter or burning it from the obligation of sending a letter) by removing WO from the proof system. For example in the system of $derive_I^{CT}$ (see Section 3.3) we have $s\in derive_I^{CT}(\{(1,s)\},\{1\})$ but $s\vee b\notin derive_I^{CT}(\{(1,s)\},\{1\})$.

4.2 Neighborhood Characterization of I/O Operations

The language of classical propositional logic consists of the connectives $\mathcal{L}_C = \{\wedge, \vee, \neg, \top, \bot\}$. Let X be a set of variables; as usual we define the set of formulas over X and refer to it by $Fm(X)$.[2] The algebra of formulas over X is the Boolean algebra as follows:

$$\mathbf{Fm}(X) = \langle Fm(X), \wedge^{\mathbf{Fm}(X)}, \vee^{\mathbf{Fm}(X)}, \neg^{\mathbf{Fm}(X)}, \top^{\mathbf{Fm}(X)}, \bot^{\mathbf{Fm}(X)} \rangle$$

where $\wedge^{\mathbf{Fm}(X)}(\varphi, \psi) = (\varphi \wedge \psi)$, $\vee^{\mathbf{Fm}(X)}(\varphi, \psi) = (\varphi \vee \psi)$, $\neg^{\mathbf{Fm}(X)}(\varphi) = \neg\varphi$, $\top^{\mathbf{Fm}(X)} = \top$, and $\bot^{\mathbf{Fm}(X)} = \bot$. We define $\varphi \vdash_C \psi$ if and only if $\varphi \leq \psi$; and $\varphi \dashv\vdash_C \phi$ if and only if $\varphi \leq \psi$ and $\psi \leq \varphi$.

Definition 4.2. Let $N \subseteq Fm(X) \times Fm(X)$ where X is a set of propositional variables. $(\varphi, \psi) \in derive_i^{\mathbf{Fm}(X)}(N)$ if and only if (φ, ψ) is derivable from N using EQO, EQI, SI, WO, OR, T as follows:

$derive_i^{\mathbf{Fm}(X)}$	Rules
$derive_R^{\mathbf{Fm}(X)}$	{EQO}
$derive_L^{\mathbf{Fm}(X)}$	{EQI}
$derive_0^{\mathbf{Fm}(X)}$	{EQI, EQO}
$derive_I^{\mathbf{Fm}(X)}$	{SI, EQO}
$derive_{II}^{\mathbf{Fm}(X)}$	{WO, EQI}
$derive_1^{\mathbf{Fm}(X)}$	{SI, WO}
$derive_2^{\mathbf{Fm}(X)}$	{SI, WO, OR}
$derive_3^{\mathbf{Fm}(X)}$	{SI, WO, T}

$$\text{EQO}\ \frac{(\varphi, \psi) \qquad \psi \dashv\vdash_C \phi}{(\varphi, \phi)} \qquad \text{WO}\ \frac{(\varphi, \psi) \qquad \psi \vdash_C \phi}{(\varphi, \phi)}$$

$$\text{EQI}\ \frac{(\varphi, \psi) \qquad \varphi \dashv\vdash_C \phi}{(\phi, \psi)} \qquad \text{OR}\ \frac{(\varphi, \psi) \qquad (\phi, \psi)}{(\varphi \vee \phi, \psi)}$$

$$\text{SI}\ \frac{(\varphi, \psi) \qquad \phi \vdash_C \varphi}{(\phi, \psi)} \qquad \text{T}\ \frac{(\varphi, \psi) \qquad (\psi, \phi)}{(\varphi, \phi)}$$

We define $(\Gamma, \psi) \in derive_i^{\mathbf{Fm}(X)}(N)$ if $(\varphi, \psi) \in derive_i^{\mathbf{Fm}(X)}(N)$ for some $\varphi \in \Gamma \subseteq Fm(X)$. Put $derive_i^{\mathbf{Fm}(X)}(N, \Gamma) = \{\psi : (\Gamma, \psi) \in derive_i^{\mathbf{Fm}(X)}(N)\}$.

Example 4.2. *For the Chisholm conditional norm set* $N = \{(\top, g), (g, t), (\neg g, \neg t)\}$ *and input set* $A = \{\neg g\}$ *we have* $out_3^{\mathbf{Fm}(X)}(N, A) = Up(t, \neg t, g)$.

Definition 4.3 (Input/Output Neighborhood model)**.** A neighborhood model for input/output logic (ION) is a triple $M = \langle W, f, V \rangle$ where W is a set of possible worlds, V is a valuation function and f is a neighborhood function $f : P(W) \to P(P(W))$ such that

[2] For the precise definition we use auxiliary symbols brackets), (. So except for the use of brackets, the formulas over X are the Boolean-terms over X: $Ter(X)$.

- SI: $Y \in f(X)$ and $Z \subseteq X$ then $Y \in f(Z)$
- WO: $Y \in f(X)$ and $Y \subseteq Z$ then $Z \in f(X)$
- OR: $Z \in f(X)$ and $Z \in f(Y)$ then $Z \in f(X \cup Y)$
- T: $Y \in f(X)$ and $Z \in f(Y)$ then $Z \in f(X)$

Satisfiability of a formula φ for a ION model $M = \langle W, f, V \rangle$ and a world $s \in S$ is expressed by writing that $M, s \models \varphi$ and we define $V^M(\varphi) = \{s \in S \mid M, s \models \varphi\}$.

$$\begin{array}{lll} M, s \models p^j & \text{iff} & s \in V(p^j) \\ M, s \models \neg\varphi & \text{iff} & M, s \not\models \varphi \,(\text{that is, not } M, s \models \varphi) \\ M, s \models \varphi \vee \psi & \text{iff} & M, s \models \varphi \text{ or } M, s \models \psi \\ M, s \models \varphi \Rightarrow \psi & \text{iff} & V(\psi) \in f(V(\varphi)) \end{array}$$

As usual, a formula φ is *valid in a ION model model* $M = \langle W, f, V \rangle$, i.e. $M \models \varphi$, if and only if for all worlds $s \in S$ we have $M, s \models \varphi$. A formula φ is *valid*, denoted $\models \varphi$, if and only if it is valid in every ION model.

Theorem 4.4 (Neighborhood model for simple-minded I/O logic)**.** *Given a set of conditional norms N such that in the ION model $M = \langle W, f, V \rangle$ if $(\varphi, \psi) \in N$ then $V(\psi) \in f(V(\varphi))$ and also f satisfies SI and WO we have*

$$(\varphi, \psi) \in \mathit{derive}_1^{\mathbf{Fm}(X)}(N) \textit{ implies } M \models \varphi \Rightarrow \psi.$$

Proof. Suppose $(\varphi, \psi) \in \mathit{derive}_1^{\mathbf{Fm}(X)}(N)$ then by completeness there is ψ_1 such that $\psi_1 \in N(Up(\varphi))$, $\psi \geq \psi_1$ and there is γ_1 such that $(\gamma_1, \psi_1) \in N$ and $\varphi \leq \gamma_1$. Then, $V(\psi_1) \in f(V(\gamma_1))$ and $V(\varphi) \subseteq V(\gamma_1)$. Since $V(\varphi) \subseteq V(\gamma_1)$ we have $V(\psi_1) \in f(V(\varphi))$ and since $V(\psi) \supseteq V(\psi_1)$ we have $V(\psi) \in f(V(\varphi))$. □

Theorem 4.5 (Neighborhood model for basic I/O logic)**.** *Given a set of conditional norms N such that in the ION model $M = \langle W, f, V \rangle$ if $(\varphi, \psi) \in N$ then $V(\psi) \in f(V(\varphi))$ and also f satisfies SI, WO and OR then we have*

$$(\varphi, \psi) \in \mathit{derive}_2^{\mathbf{Fm}(X)}(N) \textit{ implies } M \models \varphi \Rightarrow \psi.$$

Proof. The proof is by induction on the length of proof $(\varphi, \psi) \in derive_2^{\mathbf{Fm}(X)}(N)$.
Base case: If $(\varphi, \psi) \in N$ then $V(\psi) \in f(V(\varphi))$ by definition.
Inductive step: We show that for $n > 0$ if $V(\psi) \in f(V(\varphi))$ holds for n, then also $V(\psi) \in f(V(\varphi))$ holds for $n+1$.

Suppose that the length of proof $(\varphi, \psi) \in derive_2^{B}(N)$ is $n+1$. There are three possibilities:

- *Using SI in the last step:* there is δ such that $(\delta, \psi) \in derive_2^{\mathbf{Fm}(X)}(N)$ and $\varphi \leq \delta$. In this case, by induction step we have $V(\psi) \in f(V(\delta))$ and by property SI of f we have $V(\psi) \in f(V(\varphi))$.
- *Using WO in the last step:* there is δ such that $(\varphi, \delta) \in derive_2^{\mathbf{Fm}(X)}(N)$ and $\delta \leq \psi$. In this case, by induction step we have $V(\delta) \in f(V(\varphi))$ and by property WO of f we have $V(\psi) \in f(V(\varphi))$.
- *Using OR in the last step:* there is δ_1 and δ_2 such that $(\delta_1, \psi), (\delta_2, \psi) \in derive_2^{\mathbf{Fm}(X)}(N)$ and $\varphi = \delta_1 \vee \delta_2$. In this case, by induction step we have $V(\psi) \in f(V(\delta_1))$ and $V(\psi) \in f(V(\delta_2))$ and then by property OR of f we have $V(\psi) \in f(V(\varphi))$.

□

Theorem 4.6 (Neighborhood model for reusable I/O logic)**.** *Given a set of conditional norms N such that in the ION model $M = \langle W, f, V\rangle$ if $(\varphi, \psi) \in N$ then $V(\psi) \in f(V(\varphi))$ and also f satisfies SI, WO and T then we have*

$$(\varphi, \psi) \in derive_3^{\mathbf{Fm}(X)}(N) \text{ implies } M \models \varphi \Rightarrow \psi.$$

Proof. The proof is similar to Theorem 4.5. We just check the case when T is in the last step of derivation: there is δ such that $(\varphi, \delta), (\delta, \psi) \in derive_3^{\mathbf{Fm}(X)}(N)$. In this case, by induction step we have $V(\delta) \in f(V(\varphi))$ and $V(\psi) \in f(V(\delta))$ and then by property T of f we have $V(\psi) \in f(V(\varphi))$. □

4.3 Synthesizing Normative Reasoning and Preferences

Given $\langle \mathbf{Fm}(X), \vdash_C\rangle$, let $\mathcal{B}$ be a Boolean algebra and X be a set of propositional variables. A valuation on $\mathcal{B}$ is a function from X into the universe of $\mathcal{B}$. Any valuation on $\mathcal{B}$ can be extended in a unique way to a homomorphism from the algebra $\mathbf{Fm}(X)$ into $\mathcal{B}$. A valuation V on $\mathcal{B}$ satisfies a formula if $V(\varphi) = 1_{\mathcal{B}}$ and it satisfies a set of formulas if it satisfies all its elements [115].

Definition 4.7. For any Boolean algebra $\mathcal{B}$, we can define the consequence relation $\vDash_{\mathcal{B}}$ as follows:

$$\mathit{\Gamma} \vDash_{\mathcal{B}} \varphi \text{ if and only if any valuation on } \mathcal{B} \text{ that } V(\mathit{\Gamma}) = 1_{\mathcal{B}} \text{ then } V(\varphi) = 1_{\mathcal{B}}$$

Definition 4.8. Let **BA** be the class of all Boolean algebras. We can define the consequence relation $\vDash_{\mathbf{BA}}$ as follows:

$$\mathit{\Gamma} \vDash_{\mathbf{BA}} \varphi \text{ if and only if for any Boolean algebra } \mathcal{B},\ \mathit{\Gamma} \vDash_{\mathcal{B}} \varphi$$

Theorem 4.9. *For every set of formulas $\mathit{\Gamma}$ and every formula φ,*

$$\mathit{\Gamma} \vDash_{\mathbf{BA}} \varphi \textit{ if and only if } \mathit{\Gamma} \vdash_C \varphi.$$

Proof. See [46, 115]. □

Theorem 4.10. *Let X be a set of propositional variables and $N \subseteq Fm(X) \times Fm(X)$. For a given Boolean algebra $\mathcal{B}$ and a valuation V on $\mathcal{B}$ we define $N^V = \{(V(\varphi), V(\psi)|(\varphi, \psi) \in N\}$. We have*

$$(\varphi, \psi) \in derive_i^{\mathbf{Fm}(X)}(N)$$

if and only if

$$V(\psi) \in out_i^{\mathcal{B}}(N^V, \{V(\varphi)\}) \textit{ for every } \mathcal{B} \in \mathbf{BA}, \textit{ for every valuation } V \textit{ on } \mathcal{B}.$$

Proof. We do the proof for the case $i = 1$.

- Suppose $(\varphi, \psi) \in derive_1^{\mathbf{Fm}(X)}(N)$. For an arbitrary valuation V and arbitrary Boolean algebra $\mathcal{B} \in \mathbf{BA}$ we need to show that $V(\psi) \in out_1^{\mathcal{B}}(N^V, \{V(\varphi)\})$.
  The proof is by induction on the length of the proof $(\varphi, \psi) \in derive_1^{\mathbf{Fm}(X)}(N)$.
  Base case: If $(\varphi, \psi) \in N$ then $(V(\varphi), V(\psi)) \in N^V$ by definition and we have $V(\psi) \in out_1^{\mathcal{B}}(N^V, \{V(\varphi)\})$.
  Inductive step: We show that for $n > 0$ if $V(\psi) \in out_1^{\mathcal{B}}(N^V, \{V(\varphi)\})$ holds for n, then also $V(\psi) \in out_1^{\mathcal{B}}(N^V, \{V(\varphi)\})$ holds for $n + 1$.

  Suppose that the length of proof $(\varphi, \psi) \in derive_1^{\mathbf{Fm}(X)}(N)$ is $n + 1$. There are two possibilities:

- *Using SI in the last step:* there is ϕ such that $(\phi, \psi) \in derive_1^{\mathbf{Fm}(X)}(N)$ and $\varphi \vdash_C \phi$. In this case, by induction step we have $V(\psi) \in out_1^{\mathcal{B}}(N^V, \{V(\phi)\})$ and by Theorem 3.18 we have $(V(\phi), V(\psi)) \in derive_1^{\mathcal{B}}(N)$. Since $\varphi \vdash_C \phi$ then by Theorem 4.9 we have $\varphi \vDash_{\mathbf{BA}} \phi$. So $V(\varphi) \wedge V(\phi) = V(\varphi)$. Then from $(V(\phi), V(\psi)) \in derive_1^{\mathcal{B}}(N)$ and $V(\varphi) \leq V(\phi)$ using rule SI we have $(V(\varphi), V(\psi)) \in derive_1^{\mathcal{B}}(N)$ and by Theorem 3.17 $V(\psi) \in out_1^{\mathcal{B}}(N^V, \{V(\varphi)\})$.
- *Using WO in the last step:* there is ϕ such that $(\varphi, \phi) \in derive_1^{\mathcal{B}}(N)$ and $\phi \vdash_C \psi$. In this case, by induction step we have $V(\phi) \in out_1^{\mathcal{B}}(N^V, \{V(\varphi)\})$ and by Theorem 3.18 we have $(V(\varphi), V(\phi)) \in derive_1^{\mathcal{B}}(N)$. Since $\phi \vdash_C \psi$ then by Theorem 4.9 we have $\phi \vDash_{\mathbf{BA}} \psi$. So $V(\phi) \wedge V(\psi) = V(\phi)$. Then from $(V(\varphi), V(\phi)) \in derive_1^{\mathcal{B}}(N)$ and $V(\phi) \leq V(\psi)$ using rule WO we have $(V(\varphi), V(\psi)) \in derive_1^{\mathcal{B}}(N)$ and by Theorem 3.17 $V(\psi) \in out_1^{\mathcal{B}}(N^V, \{V(\varphi)\})$.

- The proof is by contraposition for the other direction. Suppose that $(\varphi, \psi) \notin derive_1^{\mathbf{Fm}(X)}(N)$, if we take $\mathbf{Fm}(X)$ as a Boolean algebra then by Theorem 3.18 $\psi \notin out_1^{\mathbf{Fm}(X)}(N, \{\varphi\})$, then if we put valuation function as the identity function on the algebra $Fm(X)$ we have $\psi \notin out_1^{\mathcal{B}=\mathbf{Fm}(X)}(N, \{\varphi\})$.

□

The proof for other derivation systems $derive_R^{\mathbf{Fm}(X)}(N)$, $derive_L^{\mathbf{Fm}(X)}(N)$, $derive_I^{\mathbf{Fm}(X)}(N)$, $derive_{II}^{\mathbf{Fm}(X)}(N)$, $derive_2^{\mathbf{Fm}(X)}(N)$, and $derive_3^{Fm(X)}(N)$ is similar. Also we can extend the proof for arbitrary input set $\Gamma \subseteq Fm(X)$. Suppose $(\Gamma, \psi) \in derive_i^{\mathbf{Fm}(X)}(N)$ then $(\varphi, \psi) \in derive_i^{\mathbf{Fm}(X)}(N)$ for $\varphi \in \Gamma$. As above we have $V(\psi) \in out_i^{\mathcal{B}}(N^V, \{V(\varphi)\})$ for every $\mathcal{B} \in \mathbf{BA}$, for every valuation V on $\mathcal{B}$; that by definition of $out_i^{\mathcal{B}}$ we can say $V(\psi) \in out_i^{\mathcal{B}}(N^V, \{V(\varphi) | V(\varphi) \in V(\Gamma)\})$ for every $\mathcal{B} \in \mathbf{BA}$, for every valuation V on $\mathcal{B}$.

Theorem 4.11. *Let X be a set of propositional variables and $N \subseteq Fm(X) \times Fm(X)$. For a given Boolean algebra $\mathcal{B}$ and a valuation V on $\mathcal{B}$ we define $N^V = \{(V(\varphi), V(\psi) | (\varphi, \psi) \in N\}$. We have*

$$(\varphi, \psi) \in derive_i^{AND}(N)$$

if and only if

$$V(\psi) \in out_i^{AND}(N^V, \{V(\varphi)\}) \text{ for every } \mathcal{B} \in \mathbf{BA}, \text{ for every valuation } V \text{ on } \mathcal{B}.$$[3]

[3] $out_i^{AND}(N^V, \{V(\varphi)\})$ is defined for every Boolean algebra $\mathcal{B}$, see Definition 3.29.

Proof.

- The proof from right to left is similar to Theorem 4.10. We just check the case when AND is the last step of derivation: there are δ_1 and δ_2 such that $(\varphi, \delta_1), (\varphi, \delta_2) \in derive_i^{AND}(N)$ and $\psi = \delta_1 \wedge \delta_2$. In this case, by induction step we have $V(\delta_1) \in out_i^{AND}(N^V, \{V(\varphi)\})$ and $V(\delta_2) \in out_i^{AND}(N^V, \{V(\varphi)\})$. By Theorem 3.32 we have $(\varphi, \delta_1) \in derive_i^{AND}(N)$ and $(\varphi, \delta_2) \in derive_i^{AND}(N)$. Then by using rule AND we have $(\varphi, \delta_1 \wedge \delta_2) \in derive_i^{AND}(N)$ and then by Theorem 3.32 $V(\psi) \in out_i^{AND}(N^V, \{V(\varphi)\})$.

- The proof is by contraposition for the other direction. Suppose that $(\varphi, \psi) \notin derive_1^{AND}(N)$, if we take $\mathbf{Fm}(X)$ as a Boolean algebra then by Theorem 3.32 $\psi \notin out_1^{AND}(N, \{\varphi\})$, then if we put valuation function as the identity function on the algebra $Fm(X)$ we have $\psi \notin out_1^{AND}(N, \{\varphi\})$.

Also we can extend the proof for arbitrary input set $\Gamma \subseteq Fm(X)$. We can extend this theorem for other adding rule operators. □

4.3.1 Consistency check

Definition 4.12. Let X be a set of propositional variables and $N \subseteq Fm(X) \times Fm(X)$. Given the constraint C that is a set of formulas $C \subseteq Fm(X)$. We define $(\varphi, \psi) \in derive_i^C(N)$ if and only if

$$(\varphi, \psi) \in derive_i^{\mathbf{Fm}(X)}(N) \text{ and } C, \psi \nvdash_C \bot.$$

Given a set of $\Gamma \subseteq Fm(X)$, we define $(\Gamma, \psi) \in derive_i^C(N)$ if $(\varphi, \psi) \in derive_i^C(N)$ for some $\varphi \in \Gamma$.

Theorem 4.13. *Let X be a set of propositional variables, $N \subseteq Fm(X) \times Fm(X)$, and $C \subseteq Fm(X)$. For a given Boolean algebra $\mathcal{B}$ and a valuation V on $\mathcal{B}$ we define $N^V = \{(V(\varphi), V(\psi)|(\varphi, \psi) \in N\}$. We have*

$$(\varphi, \psi) \in derive_i^C(N)$$

if and only if

$$V(\psi) \in out_i^{\mathcal{B}}(N^V, \{V(\varphi)\}) \textit{ for every } \mathcal{B} \in \mathbf{BA}, \textit{ for every valuation } V \textit{ on } \mathcal{B}$$

and

for some $\mathcal{B} \in \mathbf{BA}$*, there is a valuation* V *on* $\mathcal{B}$ *such that* $\forall \delta \in C$, $V(\delta \wedge \psi) = 1_{\mathcal{B}}$.

Proof.

- From left to right: Suppose $(\varphi, \psi) \in derive_i^C(N)$, then by definition $(\varphi, \psi) \in derive_i^{\mathbf{Fm}(X)}(N)$ and $C, \psi \nvdash_C \bot$. From Theorem 4.10 we have "$V(\psi) \in out_i^{\mathcal{B}}(N^V, \{V(\varphi)\})$ for every $\mathcal{B} \in \mathbf{BA}$, for every valuation V on $\mathcal{B}$" and from Theorem 4.9 there is a Boolean algebra $\mathcal{B}$ such that $C, \psi \nvDash_{\mathcal{B}} \bot$. So there is a valuation V on $\mathcal{B}$ such that $\forall \delta \in C$, $V(\delta \wedge \psi) = 1_{\mathcal{B}}$.
- The proof from right to left is similar.

By definition of $derive_i^C(N)$ we can extend the theorem for the case $(\Gamma, \psi) \in derive_i^C(N)$ where $\Gamma \subseteq Fm(X)$. □

4.3.2 Integrating preferences

For adding preference to the proposed norm-based logic we use the same idea of Hansson [109]. We can evaluate the conditionals by a preference relation over valuations.

> "A valuation for the first dyadic standard deontic logic (DSDL1) is a reflexive relation R defined on the set t of all valuations of the BL for DSDL1. A DSDL1-formula of the type $\bigcirc(f/g)$ is true in the valuation R if and only if f contains all R-maximal elements in g. Other formulas take truth values according to the rules or propositional logic. A DSDL1 formula is valid if and only if it is true in all DSDL1 valuations." (Hansson [109])

Definition 4.14. Let X be a set of propositional variables and $MaxC$ will be the set of all maximal consistent subsets of $Fm(X)$. Let $f \subseteq MaxC \times MaxC$ be a relation over elements of $MaxC$ and $\text{opt}_f(\varphi) = \{M \in MaxC \mid \varphi \in M, \forall K\,(\varphi \in K \rightarrow (M, K) \in f)\}$. We define $\varphi > \bigcirc\psi \in derive_i^{OH}(N)$ if and only if

$$(\varphi, \psi) \in derive_i^{\mathbf{Fm}(X)}(N) \text{ and } \forall M \in \text{opt}_f(\varphi)\,(\psi \in M).$$

Given a set of $\Gamma \subseteq Fm(X)$, we define $\Gamma > \bigcirc\psi \in derive_i^{OH}(N)$ if $\varphi > \bigcirc\psi \in derive_i^{OH}(N)$ for some $\varphi \in \Gamma$.

Definition 4.15. Let X be a set of propositional variables and $f \subseteq MaxC \times MaxC$. A preference Boolean algebra for $\mathbf{Fm}(X)$ is a structure $M = \langle \mathcal{B}, \mathcal{V}, \succeq_f \rangle$,

- $\mathcal{B}$ is a Boolean algebra,
- $\mathcal{V} = \{V_i\}_{i \in I}$ is the set of valuations from $\mathbf{Fm}(X)$ on $\mathcal{B}$,
- $\succeq_f \subseteq \mathcal{V} \times \mathcal{V}$: $\succeq_f$ is a betterness or comparative goodness relation over valuations from $\mathbf{Fm}(X)$ to $\mathcal{B}$ such that $V_i \succeq_f V_j$ iff $(\{\varphi | V_i(\varphi) = 1_{\mathcal{B}}\}, \{\psi | V_j(\psi) = 1_{\mathcal{B}}\}) \in f$.

No specific properties (like reflexivity or transitivity) are required of the betterness relation. For a given preference Boolean algebra $M = \langle \mathcal{B}, V, \succeq_f \rangle$, we define $opt_{\succeq_f}(\varphi) = \{V_i \in \mathcal{V} \mid V_i(\varphi) = 1_{\mathcal{B}}, \forall V_j (V_j(\varphi) = 1_{\mathcal{B}} \to V_i \succeq_f V_j)\}$.

Theorem 4.16. *Let X be a set of propositional variables, $N \subseteq Fm(X) \times Fm(X)$, and $f \subseteq MaxC \times MaxC$. For a given Boolean algebra $\mathcal{B}$ and a valuation V on $\mathcal{B}$ we define $N^V = \{(V(\varphi), V(\psi) | (\varphi, \psi) \in N\}$. We have*

$$\varphi > \bigcirc\psi \in derive_i^{O^H}(N)$$

if and only if

$$V(\psi) \in out_i^{\mathcal{B}}(N^V, \{V(\varphi)\}) \text{ for every } \mathcal{B} \in \mathbf{BA}, \text{ for every valuation } V \text{ on } \mathcal{B}$$

and

$$\text{for every preference Boolean algebra } M = \langle \mathcal{B}, \mathcal{V}, \succeq_f \rangle,$$
$$\text{for every valuation } V_i \in opt_{\succeq_f}(\varphi) \text{ we have } V_i(\psi) = 1_{\mathcal{B}}.$$

Proof.

- From left to right: Suppose $\varphi > \bigcirc\psi \in derive_i^{O^H}(N)$, by definition $(\varphi, \psi) \in derive_i^{\mathbf{Fm}(X)}(N)$ and from Theorem 4.10 we have "$V(\psi) \in out_i^{\mathcal{B}}(N^V, \{V(\varphi)\})$ for every $\mathcal{B} \in \mathbf{BA}$, for every valuation V on $\mathcal{B}$". For the second part we need to notice that every maximal consistent subset defines a valuation and vice versa. So "$\forall M \in \mathrm{opt}_f(\varphi)(\psi \in M)$" is equivalent to that for any valuation $V_i \in opt_{\succeq_f}(\varphi)$ we have $V_i(\psi) = 1_{\mathcal{B}}$ and vice versa.
- From right to left the proof is similar.

By definition of $derive_i^{O^H}(N)$ we can extend the theorem for the case $\Gamma > \bigcirc\psi \in derive_i^{O^H}(N)$ where $\Gamma \subseteq Fm(X)$. □

We can rewrite the theorem as follows also:

$$\varphi > \bigcirc\psi \in \mathit{derive}_i^{OH}(N)$$

if and only if

$$\psi \in \mathit{out}_i^{\mathbf{Fm}(X)}(N, \{\varphi\}) \text{ and in } M = \langle \mathbf{2}, \mathcal{V}, \succeq_f \rangle,$$
$$\text{for every valuation } V_i \in \mathit{opt}_{\succeq_f}(\varphi) \text{ we have } V_i(\psi) = 1_{\mathcal{B}}$$[4]

Example 4.3. *For the Chisholm conditional norm set $N = \{(\top, g), (g, t), (\neg g, \neg t)\}$, we can order the maximal consistent sets as follows: the best maximal consistent sets have both g and t (type s_1); the second best maximal consistence sets are those we have either $\{g, \neg t\}$ (type s_2) or $\{\neg g, \neg t\}$ (type s_3). The worst maximal consistent sets are those we have $\{\neg g, t\}$ (type s_4).*

best $s_1 \bullet g, t$

$- - - - - - - - - - - - - - - -$

2nd best $s_2 \bullet g$ $s_3 \bullet$

$- - - - - - - - - - - - - - - -$

worst $s_4 \bullet t$

Since $\forall M \in \mathrm{opt}_f(\top)\,(g \in M)$, $\forall M \in \mathrm{opt}_f(g)\,(t \in M)$ and $\forall M \in \mathrm{opt}_f(\neg g)\,(\neg t \in M)$ we have $\top > \bigcirc g, g > \bigcirc t, \neg g > \bigcirc\neg t \in \mathit{derive}_i^{OH}(N)$.

Example 4.4 (Dilemma problem)**.** *Consider the norm set of $N = \{(\top, \neg c), (k, c)\}$, as formal representation of these two sentences: a person should not offer someone else a cigarette; an assassin should offer a victim a cigarette, if he kills him. Prakken and Sargot [174] argue that this example represents a dilemma. (k, c) is not a CTD obligation of $(\top, \neg c)$, it is an exception to $(\top, \neg c)$. We need a suitable non-monotonic defeat mechanism to formalize this dilemma [213]. The set of $\{\bigcirc(\top/\neg c), \bigcirc(k/c)\}$ is inconsistent in prohairetic deontic logic (PDL) [213]. PDL detect the dilemma by making the set of formulas inconsistent. In our setting the set of $\{\top > \bigcirc\neg c, k > \bigcirc c\}$ is consistent. We detect the dilemma by using detachment in output operations. We can order the maximal consistent sets as follows: the best maximal consistent sets have both $\neg c$ and $\neg k$ (type s_1); the second best maximal consistence sets are those we have either $\{c, k\}$ (type s_2) or $\{c, \neg k\}$ (type s_3). The worst maximal consistent sets are those we have $\{\neg c, k\}$ (type s_4). Similar to the CTD problems we can derive, $\top > \bigcirc\neg c, k > \bigcirc c \in \mathit{derive}_i^{OH}(N)$. Here, I/O operations can show the inconsistency within the dilemma, the dyadic part is logically too weak for*

[4] If $V_i \in \mathit{opt}_{\succeq_f}(\varphi)$ in $M = \langle \mathbf{2}, \mathcal{V}, \succeq_f \rangle$, then we have $V_i \in \mathit{opt}_{\succeq_f}(\varphi)$ in every preference Boolean algebra $M = \langle \mathcal{B}, \mathcal{V}, \succeq_f \rangle$.

this task. In fact, the set of $out_i^{\mathbf{Fm}(X)}(N,\{\top\}) \cup out_i^{\mathbf{Fm}(X)}(N,\{k\})$ *is not consistent since* $\neg c \in out_i^{\mathbf{Fm}(X)}(N,\{\top\})$ *and* $c \in out_i^{\mathbf{Fm}(X)}(N,\{k\})$*. More interestingly, for the output operation closed under AND, we have* $\bot \in out_i^{AND}(N,\{\top,k\})$*.*

Example 4.5 (Miners paradox)**.** *Consider the miners norm set of* $N = \{(shA, blA), (shB, blB), (\top, \neg blA \wedge \neg blB)\}$*, see Subsection 3.3.1. We can abbreviate six types of worlds or maximal consistent sets as* AA, AB, AN, BA, BB, BN*. The first letter denotes which shaft the miners are in; the second letter denotes which shaft (or neither) we block. A plausible ordering is:* $AN, BN > AA, BB > AB, BA$ *; for more details see [218]. Regarding this ordering we have* $\top > \bigcirc(\neg blA \wedge \neg blB) \in derive_i^{OH}(N)$*.*

4.3.3 Integrating preferences along premise sets

Similar to Lewis [133] and Kratzer [129], we can introduce preference over valuations by means of a premise set. Valuations play the role of possible worlds here.

> "Quite generally, a set of propositions A can induce an ordering $\leq_A$ on W in the following way: [...] For all worlds w and $z \in W$: $w \leq_A z$ iff $\{p : p \in A \,\text{and}\, z \in p\} \subseteq \{p : p \in A \,\text{and}\, w \in p\}$ "(Kratzer [129], p. 39)

Definition 4.17. Let X be a set of propositional variables and $MaxC$ will be the set of all maximal consistent subsets of $Fm(X)$. For $A \subseteq Fm(X)$, let $f^A \subseteq MaxC \times MaxC$ such that $f^A = \{(K,M) | \forall\varphi \in A, (\varphi \in M \to \varphi \in K)\}$ be a relation over elements of $MaxC$. Let $opt_{f^A}(\varphi) = \{M \in MaxC \mid \varphi \in M, \forall K\, (\varphi \in K \to (M,K) \in f^A)\}$. We define $\varphi > \bigcirc\psi \in derive_i^{OK}(N)$ if and only if

$$(\varphi,\psi) \in derive_i^{\mathbf{Fm}(X)}(N) \text{ and } \forall M \in opt_{f^A}(\varphi)\, (\psi \in M).$$

Given a set of $\Gamma \subseteq Fm(X)$, we define $\Gamma > \bigcirc\psi \in derive_i^{OK}(N)$ if $\varphi > \bigcirc\psi \in derive_i^{OK}(N)$ for some $\varphi \in \Gamma$.

Definition 4.18. Let X be a set of propositional variables and $A \subseteq Fm(X)$. A factual-preference Boolean algebra for $\mathbf{Fm}(X)$ is a structure $M = \langle \mathcal{B}, \mathcal{V}, \succeq_A \rangle$,

- $\mathcal{B}$ is a Boolean algebra,
- $\mathcal{V} = \{V_i\}_{i \in I}$ is the set of valuations from $\mathbf{Fm}(X)$ on $\mathcal{B}$,

- $\succeq_A \subseteq \mathcal{V} \times \mathcal{V}$ such that ($V_i \succeq_A V_j$ iff $\forall \varphi \in A \quad (V_j(\varphi) = 1_{\mathcal{B}} \to V_i(\varphi) = 1_{\mathcal{B}})$).

Here, the betterness relation is reflexive or transitive by definition. For a given preference Boolean algebra $M = \langle \mathcal{B}, V, \succeq_A \rangle$, we define $opt_{\succeq_A}(\varphi) = \{V_i \in \mathcal{V} \mid V_i(\varphi) = 1_{\mathcal{B}}, \forall V_j (V_j(\varphi) = 1_{\mathcal{B}} \to V_i \succeq_A V_j)\}$.

Theorem 4.19. *Let X be a set of propositional variables, $N \subseteq Fm(X) \times Fm(X)$, and $A \subseteq Fm(X)$. For a given Boolean algebra $\mathcal{B}$ and a valuation V on $\mathcal{B}$ we define $N^V = \{(V(\varphi), V(\psi)|(\varphi, \psi) \in N\}$. We have*

$$\varphi > \bigcirc\psi \in derive_i^{OK}(N)$$

if and only if

$V(\psi) \in out_i^{\mathcal{B}}(N^V, \{V(\varphi)\})$ *for every* $\mathcal{B} \in \mathbf{BA}$, *for every valuation* V *on* $\mathcal{B}$

and

for every factual-preference Boolean algebra $M = \langle \mathcal{B}, \mathcal{V}, \succeq_A \rangle$,
for every valuation $V_i \in opt_{\succeq_A}(\varphi)$ *we have* $V_i(\psi) = 1_{\mathcal{B}}$.

Proof.

- From left to right: Suppose $(\varphi, \psi) \in derive_i^{OK}(N$, by definition $(\varphi, \psi) \in derive_i^{\mathbf{Fm}(X)}(N)$ and from Theorem 4.10 we have "$V(\psi) \in out_i^{\mathcal{B}}(N^V, \{V(\varphi)\})$ for every $\mathcal{B} \in \mathbf{BA}$, for every valuation V on $\mathcal{B}$". For the second part we need to notice that every maximal consistent subset defines a valuation and vice versa. So "$\forall M \in opt_{fA}(\varphi)(\psi \in M)$" is equivalent to that for any valuation $V_i \in opt_{\succeq_A}(\varphi)$ we have $V_i(\psi) = 1_{\mathcal{B}}$ and vice versa.
- From right to left the proof is similar.

By definition of $derive_i^{OK}(N)$ we can extend the theorem for the case $\Gamma > \bigcirc\psi \in derive_i^{OK}(N)$ where $\Gamma \subseteq Fm(X)$. □

We can rewrite the theorem as follows:

$$\varphi > \bigcirc\psi \in derive_i^{OK}(N)$$

if and only if

$$\psi \in out_i^{\mathbf{Fm}(X)}(N, \{\varphi\}) \text{ and for } M = \langle \mathbf{2}, \mathcal{V}, \succeq_A \rangle,$$
$$\text{for every valuation } V_i \in opt_{\succeq_A}(\varphi) \text{ we have } V_i(\psi) = 1_{\mathcal{B}}$$

or

$$\psi \in out_i^{\mathbf{Fm}(X)}(N, \{\varphi\}) \text{ and if } \varphi \text{ is consistent with } A$$
$$\text{then } A, \varphi \vdash \psi \text{ and if } \varphi \text{ is inconsistent with } A \text{ then } \varphi \vdash \psi[5]$$

The results for the constrained assumptions and preferences can be extended for the other introduced systems such as $derive_i^{AND}(N)$.

Example 4.6. *For the Chisholm conditional norm set* $N = \{(\top, g), (g, t), (\neg g, \neg t)\}$ *and premise set* $A = \{\neg g, \neg g \to \neg t\}$*. The best maximal consistent sets are type* s_3 *(see Example 4.3), those are type* s_1, s_2 *and* s_4 *are the second best.*

best $s_3\bullet$

– – – – – – – – – – – – – – – –

2nd best $s_2 \bullet g$ $s_1 \bullet g, t$ $s_4 \bullet t$

Since $\forall M \in opt_{fA}(\neg g)\,(\neg t \in M)$ *we have* $\neg g > \bigcirc \neg t \in derive_i^{OK}(N)$*.*

It is straightforward to rewrite the theorems for conditional permissions.

$\varphi > P\psi \in derive_i^{PK}(N)$
if and only if
$(\varphi, \psi) \in derive_i^{\mathbf{Fm}(X)}(N)$ and
For every factual-preference Boolean algebra $M = \langle \mathcal{B}, \mathcal{V}, \succeq_A \rangle$,
there is a valuation $V_i \in opt_{\succeq_A}(\varphi)$ such that $V_i(\psi) = 1_{\mathcal{B}}$

$\varphi > P\psi \in derive_i^{PH}(N)$
if and only if
$(\varphi, \psi) \in derive_i^{\mathbf{Fm}(X)}(N)$ and
For every preference Boolean algebra $M = \langle \mathcal{B}, \mathcal{V}, \succeq_f \rangle$,
there is a valuation $V_i \in opt_{\succeq_f}(\varphi)$ such that $V_i(\psi) = 1_{\mathcal{B}}$

4.4 Norm-based Sequent Calculus

Definition 4.20. Let X be a set of propositional variables. A sequent has the form

[5] By definition of $opt_{fA}(\varphi)$, ψ is true in all maximal consistent subsets include both A and φ, or, when φ is inconsistent with A, in all maximal consistent subsets include φ.

$$N \mapsto_i (\varphi, \psi)$$

where $N \subseteq Fm(X) \times Fm(X)$ and $(\varphi, \psi) \in Fm(X) \times Fm(X)$.

Definition 4.21. We define inference rules of norm-based sequent calculus as follows:

$$\text{ID}\,\frac{(\varphi, \psi) \in N}{N \mapsto (\varphi, \psi)}$$

$$\text{EQO}\,\frac{N \mapsto_i (\varphi, \psi) \qquad \psi \dashv\vdash_C \phi}{N \mapsto_i (\varphi, \phi)} \qquad \text{WO}\,\frac{N \mapsto_i (\varphi, \psi) \qquad \psi \vdash_C \phi}{N \mapsto_i (\varphi, \phi)}$$

$$\text{EQI}\,\frac{N \mapsto_i (\varphi, \psi) \qquad \varphi \dashv\vdash_C \phi}{N \mapsto_i (\phi, \psi)} \qquad \text{OR}\,\frac{N \mapsto_i (\varphi, \psi) \qquad N \mapsto_i (\phi, \psi)}{N \mapsto_i (\varphi \vee \phi, \psi)}$$

$$\text{SI}\,\frac{N \mapsto_i (\varphi, \psi) \qquad \phi \vdash_C \varphi}{N \mapsto_i (\phi, \psi)} \qquad \text{T}\,\frac{N \mapsto_i (\varphi, \psi) \qquad N \mapsto_i (\psi, \phi)}{N \mapsto_i (\varphi, \phi)}$$

We can define different derivation systems as before where $\mapsto_R = \{ID, EQO\}$, $\mapsto_L = \{ID, EQI\}$, $\mapsto_I = \{ID, SI, EQO\}$, $\mapsto_{II} = \{ID, WO, EQI\}$, $\mapsto_1 = \{ID, SI, WO\}$, $\mapsto_2 = \{ID, SI, WO, OR\}$, $\mapsto_3 = \{ID, SI, WO, T\}$ and $\mapsto_4 = \{ID, SI, WO, OR, T\}$.

Definition 4.22 (Interpretation of sequents)**.** For the interpretation of sequents $N \mapsto_i (\varphi, \psi)$ we extend the valuation function from $\mathbf{Fm}(X)$ into Boolean algebra $\mathcal{B}$ over subsets $N \subseteq Fm(X) \times Fm(X)$ such that $V(N) = N^V = \{(V(\varphi), V(\psi)|(\varphi, \psi) \in N\}$. So the sequent $N \mapsto_i (\varphi, \psi)$ can be interpreted as $V(\psi) \in out_i^{\mathcal{B}}(N^V, \{V(\varphi)\})$.

Theorem 4.23. *Let X be a set of propositional variables and $N \subseteq Fm(X) \times Fm(X)$. For a given Boolean algebra $\mathcal{B}$ and a valuation V on $\mathcal{B}$ we define $N^V = \{(V(\varphi), V(\psi))|(\varphi, \psi) \in N\}$. We can derive the sequent*

$$N \mapsto_i (\varphi, \psi)$$

if and only if

$$V(\psi) \in out_i^{\mathcal{B}}(N^V, \{V(\varphi)\}) \text{ for every } \mathcal{B} \in \mathbf{BA}, \text{ for every valuation } V \text{ on } \mathcal{B}.$$

Proof. The proof is similar to Theorem 4.10. Instead of induction on the length of proof $(\varphi, \psi) \in derive_i^{\mathbf{Fm}(X)}(N)$ we do induction on the length of derivation for $N \mapsto_i (\varphi, \psi)$. So we have the similar base case and inductive step in the induction. □

For an input set $\Gamma \subseteq Fm(X)$ we can extend the sequent calculus by another type of sequent $N \mapsto_i (\Gamma, q)$. We need to add a new rule to the system as follows:

$$\text{IDD}\ \frac{N \mapsto_i (\varphi, \psi), \quad \varphi \in \Gamma}{N \mapsto_i (\Gamma, \psi)}$$

This new sequent can be interpreted as $V(\psi) \in out_i^{\mathcal{B}}(N^V, \{V(\varphi)|V(\varphi) \in V(\Gamma)\})$. Moreover, we can extend the sequent calculus systems by adding other rules such as AND rule.

4.4.1 Nested norm-based conditional logic

Let X be a set of propositional variables and $N \subseteq (Fm(X) \times Fm(X)) \times (Fm(X) \times Fm(X))$. We can define input/output operations over each $\mapsto_i$.

Definition 4.24. $((\varphi, \psi), (\phi, \chi)) \in derive_i^{Nes}(N)$ if and only if $((\varphi, \psi), (\phi, \chi))$ is derivable from N using SI, WO, T.[6]

$$\text{SI}\ \frac{((\varphi, \psi), (\phi, \chi)) \qquad \{(\upsilon, \tau)\} \mapsto_i (\varphi, \psi)}{((\upsilon, \tau), (\phi, \chi))} \qquad \text{T}\ \frac{((\varphi, \psi), (\phi, \chi)) \qquad ((\phi, \chi), (\upsilon, \tau))}{((\varphi, \psi), (\upsilon, \tau))}$$

$$\text{WO}\ \frac{((\varphi, \psi), (\phi, \chi)) \qquad \{(\phi, \chi)\} \mapsto_i (\upsilon, \tau)}{((\varphi, \psi), (\upsilon, \tau))}$$

For $A \subseteq Fm(X) \times Fm(X)$, we define $(A, (\phi, \chi)) \in derive_i^{Nes}(N)$ if $((\varphi, \psi), (\phi, \chi)) \in derive_i^{Nes}(N)$ for some $(\varphi, \psi) \in A$. Put $derive_i^{Nes}(N, A) = \{((\varphi, \psi), (\phi, \chi)) : (A, (\phi, \chi)) \in derive_i^{Nes}(N)\}$.

Theorem 4.25. *$((\varphi, \psi), (\phi, \chi)) \in derive_i^{Nes}(N)$ iff $(\phi, \chi) \in out_i^{\mapsto_i}(N, \{(\varphi, \psi)\})$, where $i \in \{1, 3\}$.*

Proof. Since $\mapsto_i$ is a consequence relation over set of formulas $(Fm(X) \times Fm(X))$ by using Theorem 3.36, for $N \subseteq (Fm(X) \times Fm(X)) \times (Fm(X) \times Fm(X))$, we have: $((\varphi, \psi), (\phi, \chi)) \in derive_i^{Nes}(N)$ iff $(\phi, \chi) \in out_i^{\mapsto_i}(N, \{(\varphi, \psi)\})$. □

[6] We can use other rules such as OR, if we find an interpolation for the logical connectives.

4.5 I/O Mechanism over Heyting Algebras

Definition 4.26 (Heyting algebra)**.** A structure $\mathcal{H} = \langle H, \wedge, \vee, \rightarrow, 0, 1\rangle$ is Heyting algebra iff it satisfies following conditions for $x, y, z \in H$:

- $x \wedge y \leq x$, $x \wedge y \leq y$, $z \leq x$ and $z \leq y$ implies $z \leq x \wedge y$
- $x \leq x \vee y$, $y \leq x \vee y$, $x \leq z$ and $y \leq z$ implies $x \vee y \leq z$
- $x \leq 1$, $0 \leq x$, $z \leq (x \rightarrow y)$ iff $z \wedge x \leq y$

where $x \leq y$ iff $x \wedge y = x$. We can similarly define the set of Heyting-terms $Ter(H)$ and upward-closed set of A by $Up(A)$.

4.5.1 Semantics

Definition 4.27 (Zero Heyting I/O operation)**.** Given a Heyting algebra $\mathcal{H}$, a normative system $N \subseteq Ter(H) \times Ter(H)$ and an input set $A \subseteq Ter(H)$, we define the zero Heyting operation as follows:

- $out^{\mathcal{H}}_{R}(N, A) = Eq(N(A))$
- $out^{\mathcal{H}}_{L}(N, A) = N(Eq(A))$
- $out^{\mathcal{H}}_{0}(N, A) = Eq(N(Eq(A)))$

Definition 4.28 (Simple-minded Heyting I/O operation)**.** Given a Heyting algebra $\mathcal{H}$, a normative system $N \subseteq Ter(H) \times Ter(H)$ and an input set $A \subseteq Ter(H)$, we define the simple-minded Heyting operation as follows:

- $out^{\mathcal{H}}_{I}(N, A) = Eq(N(Up(A)))$
- $out^{\mathcal{H}}_{II}(N, A) = Up(N(Eq(A)))$
- $out^{\mathcal{H}}_{1}(N, A) = Up(N(Up(A)))$

Definition 4.29 (Basic Heyting I/O operation)**.** Given a Heyting algebra $\mathcal{H}$, a normative system $N \subseteq Ter(H) \times Ter(H)$ and an input set $A \subseteq Ter(H)$, we define the basic Heyting operation as follows:

$$out_2^{\mathcal{H}}(N,A)=\bigcap\{Up(N(V)),A\subseteq V,V\text{is saturated}\}$$

A set V is saturated in a Heyting algebra $\mathcal{H}$ iff

- if $a\in V$ and $b\geq a$, then $b\in V$;
- if $a\vee b\in V$, then $a\in V$ or $b\in V$.

Definition 4.30 (Reusable Heyting I/O operation)**.** Given a Heyting algebra $\mathcal{H}$, a normative system $N\subseteq Ter(H)\times Ter(H)$ and an input set $A\subseteq Ter(H)$, we define the reusable Heyting operation as follows:

$$out_3^{\mathcal{H}}(N,A)=\bigcap\{Up(N(V)),A\subseteq V=Up(V)\supseteq N(V)\}$$

4.5.2 Proof system

Given a Heyting algebra $\mathcal{H}$ and a normative system $N\subseteq Ter(H)\times Ter(H)$, we define $(a,x)\in derive_i^H(N)$ if and only if (a,x) is derivable from N using EQO, EQI, SI, WO, OR, T as follows:

$derive_i^{\mathcal{H}}$	Rules
$derive_R^{\mathcal{H}}$	{EQO}
$derive_L^{\mathcal{H}}$	{EQI}
$derive_L^{\mathcal{H}}$	{EQI, EQO}
$derive_1^{\mathcal{H}}$	{SI, EQO}
$derive_1^{\mathcal{H}}$	{WO, EQI}
$derive_1^{\mathcal{H}}$	{SI, WO}
$derive_2^{\mathcal{H}}$	{SI, WO, OR}
$derive_3^{\mathcal{H}}$	{SI, WO, T}

$$\text{EQO}\ \frac{(a,x)\qquad x=y}{(a,y)}\qquad\qquad \text{WO}\ \frac{(a,x)\qquad x\leq y}{(a,y)}$$

$$\text{EQI}\ \frac{(a,x)\qquad a=b}{(b,x)}\qquad\qquad \text{OR}\ \frac{(a,x)\qquad (b,x)}{(a\vee b,x)}$$

$$\text{SI}\ \frac{(a,x)\qquad b\leq a}{(b,x)}\qquad\qquad \text{T}\ \frac{(a,x)\qquad (x,y)}{(a,y)}$$

We define $(A,x)\in derive_i^{\mathcal{H}}(N)$ if $(a,x)\in derive_i^{\mathcal{H}}(N)$ for some $a\in A\subseteq Ter(H)$. Put $derive_i^{\mathcal{H}}(N,A)=\{x:(A,x)\in derive_i^{\mathcal{H}}(N)\}$.

Theorem 4.31. *Given a Heyting algebra $\mathcal{H}$ and a normative system $N\subseteq Ter(H)\times Ter(H)$, for an input set $A\subseteq Ter(H)$ we have*

$$(A,x) \in derive_R^{\mathcal{H}}(N) \text{ iff } x \in out_R^{\mathcal{H}}(N,A) \qquad (A,x) \in derive_{II}^{\mathcal{H}}(N) \text{ iff } x \in out_{II}^{\mathcal{H}}(N,A)$$

$$(A,x) \in derive_L^{\mathcal{H}}(N) \text{ iff } x \in out_L^{\mathcal{H}}(N,A) \qquad (A,x) \in derive_1^{\mathcal{H}}(N) \text{ iff } x \in out_1^{\mathcal{H}}(N,A)$$

$$(A,x) \in derive_0^{\mathcal{H}}(N) \text{ iff } x \in out_0^{\mathcal{H}}(N,A) \qquad (A,x) \in derive_2^{\mathcal{H}}(N) \text{ iff } x \in out_2^{\mathcal{H}}(N,A)$$

$$(A,x) \in derive_I^{\mathcal{H}}(N) \text{ iff } x \in out_I^{\mathcal{H}}(N,A) \qquad (A,x) \in derive_3^{\mathcal{H}}(N) \text{ iff } x \in out_3^{\mathcal{H}}(N,A)$$

Proof. Use the usual ordering in the given Heyting algebra $\mathcal{H}$ by $\leq$. The proofs are same as soundness and completeness theorems in Section 3.2. □

4.6 Intuitionistic Norm-based Conditional Logic

For the language of intuitionistic propositional logic $\mathcal{L}_{IPC} = \{\wedge, \vee, \rightarrow, \top, \bot\}$, the algebra of formulas over a set of variables X is the Heyting algebra as follows (for more details see Section 4.2):

$$\mathbf{Fm}(X) = \langle Fm(X), \wedge^{\mathbf{Fm}(X)}, \vee^{\mathbf{Fm}(X)}, \rightarrow^{\mathbf{Fm}(X)}, \top^{\mathbf{Fm}(X)}, \bot^{\mathbf{Fm}(X)} \rangle$$

We can represent an intuitionistic propositional logic as a pair $\langle \mathbf{Fm}(X), \vdash_{IPC} \rangle$. Let $\mathcal{H}$ be a Heyting algebra and X be a set of propositional variables. A valuation on $\mathcal{H}$ is a function from X into the universe of $\mathcal{H}$. Any valuation on $\mathcal{H}$ can be extended in a unique way to a homomorphism from the algebra $\mathbf{Fm}(X)$ into $\mathcal{H}$. A valuation V on $\mathcal{H}$ satisfies a formula if $V(\varphi) = 1_{\mathcal{H}}$ and it satisfies a set of formulas if it satisfies all its elements [115].

Definition 4.32. For any Heyting algebra $\mathcal{H}$, we can define the consequence relation $\vDash_{\mathcal{H}}$ as follows:

$$\Gamma \vDash_{\mathcal{H}} \varphi \text{ if and only if any valuation on } \mathcal{H} \text{ that } V(\Gamma) = 1_{\mathcal{H}} \text{ then } V(\varphi) = 1_{\mathcal{H}}$$

Definition 4.33. Let **HA** be the class of all Boolean algebras. We can define the consequence relation $\vDash_{\mathbf{HA}}$ as follows:

$$\Gamma \vDash_{\mathbf{HA}} \varphi \text{ if and only if for any Heyting algebra } \mathcal{H},\ \Gamma \vDash_{\mathcal{H}} \varphi$$

Theorem 4.34. *For every set of formulas Γ and every formula φ,*

$$\Gamma \vDash_{\mathbf{HA}} \varphi \textit{ if and only if } \Gamma \vdash_{IPC} \varphi.$$

Proof. See [46, 115]. □

Definition 4.35. Let $N \subseteq Fm(X) \times Fm(X)$ where X is a set of propositional variables. $(a, x) \in derive_i^{\mathbf{Fm}(X)}(N)$ if and only if (a, x) is derivable from N using EQO, EQI, SI, WO, OR, T as follows:

$derive_i^{\mathbf{Fm}(X)}$	Rules
$derive_R^{\mathbf{Fm}(X)}$	{EQO}
$derive_L^{\mathbf{Fm}(X)}$	{EQI}
$derive_0^{\mathbf{Fm}(X)}$	{EQI, EQO}
$derive_I^{\mathbf{Fm}(X)}$	{SI, EQO}
$derive_{II}^{\mathbf{Fm}(X)}$	{WO, EQI}
$derive_1^{\mathbf{Fm}(X)}$	{SI, WO}
$derive_2^{\mathbf{Fm}(X)}$	{SI, WO, OR}
$derive_3^{\mathbf{Fm}(X)}$	{SI, WO, T}

$$\text{EQO}\ \frac{(\varphi, \psi) \qquad \psi \dashv\vdash_{IPC} \phi}{(\varphi, \phi)} \qquad \text{WO}\ \frac{(\varphi, \psi) \qquad \psi \vdash_{IPC} \phi}{(\varphi, \phi)}$$

$$\text{EQI}\ \frac{(\varphi, \psi) \qquad \varphi \dashv\vdash_{IPC} \phi}{(\phi, \psi)} \qquad \text{OR}\ \frac{(\varphi, \psi) \qquad (\phi, \psi)}{(\varphi \vee \phi, \psi)}$$

$$\text{SI}\ \frac{(\varphi, \psi) \qquad \phi \vdash_{IPC} \varphi}{(\phi, \psi)} \qquad \text{T}\ \frac{(\varphi, \psi) \qquad (\psi, \phi)}{(\varphi, \phi)}$$

We define $(\Gamma, \psi) \in derive_i^{\mathbf{Fm}(X)}(N)$ if $(\varphi, \psi) \in derive_i^{\mathbf{Fm}(X)}(N)$ for some $\varphi \in \Gamma \subseteq Fm(X)$. Put $derive_i^{\mathbf{Fm}(X)}(N, \Gamma) = \{\psi : (\Gamma, \psi) \in derive_i^{\mathbf{Fm}(X)}(N)\}$.

Theorem 4.36. *Let X be a set of propositional variables and $N \subseteq Fm(X) \times Fm(X)$. For a given Heyting algebra $\mathcal{H}$ and valuation V on $\mathcal{H}$ we define $N^V = \{(V(\varphi), V(\psi) | (\varphi, \psi) \in N\}$. We have*

$$(\varphi, \psi) \in derive_i^{\mathbf{Fm}(X)}(N)$$

if and only if

$$V(\psi) \in out_i^{\mathcal{H}}(N^V, \{V(\varphi)\}) \textit{ for every } \mathcal{H} \in \mathbf{HA} \textit{ for every valuation } V \textit{ on } \mathcal{H}.$$

Proof. The proof is similar to Theorem 4.10 by induction on the length of proof $(\varphi, \psi) \in derive_i^{\mathbf{Fm}(X)}(N)$ and Theorem 4.31. By definition of $derive_i^{\mathbf{Fm}(X)}(N)$ we can extend the theorem for the case $(\Gamma, \psi) \in derive_i^{\mathbf{Fm}(X)}(N)$ where $\Gamma \subseteq Fm(X)$. □

Like classical propositional logic, we can add other rules or introduce constraints, preference, and premise sets to the intuitionistic one as well as sequent calculus and nested conditionals.

4.7 Conclusion

We have introduced a compositional (norm-based) conditional logic, for obligation and permission, by adding a set of explicitly giving conditional norms to classical and intuitionistic propositional logic. The evaluation of conditionals is by referring to a set of norms and a preference order. The presented framework is capable of resolving contrary-to-duties paradoxes. Moreover, sequent calculus and counterpart neighborhood semantics were given. The system can represent nested conditionals that would be a good linguistic property. The proof system of the new compositional system is based on the included I/O proof system and the dyadic operator. For example, since the identity, $(\varphi, \varphi) \in derive_i^{\mathbf{Fm}(X)}(N)$, does not generally hold in the I/O proof systems, the proposed compositional system does not satisfy this property. More specifically, suppose there are no specific properties over the preference relation; the related axiomatization is the logical system **E** studied in Chapter 6. If we combine the simple-minded output operation with the preference relation, the resulting compositional operator satisfies weakening of the output (WO). Furthermore, since the dyadic operator validates reasoning by cases (OR), the compositional system, included the basic output operation, verifies this rule as a theorem. It is open to characterize and find the limit of the axiomatization of this framework.

Chapter 5

Norms, Semantics, and Computation

> *"deontic logic has fallen into ruts. The older rut is the axiomatic approach, with its succession of propositional and occasionally quantified calculi. The newer one is possible worlds semantics, with endless minor variations in the details."*
>
> (Makinson [138])

In this chapter we have devised an indirect and a direct approach for semantical embedding of input/output logic into HOL.

Indirect approach We propose a deontic reasoner based on I/O logic. *Basic Output* and *Basic Reusable Output* are two important I/O semantics that can be formulated with possible worlds semantics. We encode these two I/O semantics in classical higher-order logic (HOL). For the embeddings of *Basic Output* and *Basic Reusable Output* in HOL, we use the shallow semantical embedding of Kripke semantics (**K** and **KT**) in classical higher-order logic. Both of the semantical embeddings are faithful.

Theorem 5.1. *$x \in out_2(G, A)$ if and only if $x \in Cn(G(\mathcal{L}))$ and $G^\Box \cup A \vdash_{\mathbf{S}} \Box x$ for any modal logic $\boldsymbol{S}$ with $\boldsymbol{K_0} \subseteq \boldsymbol{S} \subseteq \boldsymbol{K45}$.*[1]

[1] See Subsection 2.2.2 for definitions. $\mathbf{K_0}$ is a subsystem of system $\mathbf{K}$ with axiom K, *modus ponens* and the inference rule "from ψ, infer $\Box\psi$, for all tautologies in propositional logic". $G^\Box$ denotes the set containing all modal formulas of the form $b \to \Box y$, such that $(b, y) \in G$. We have that $G^\Box \cup A \vdash_{\mathbf{S}} \Box x$ if for a finite subset Y of $G^\Box \cup A$, it holds that $(\bigwedge Y \to \Box x) \in \mathbf{S}$. The notation $\bigwedge Y$ stands for the conjunction of all the elements $y_1, y_2, \ldots, y_n$ in Y, i.e., $y_1 \wedge y_2 \wedge \cdots \wedge y_n$

Theorem 5.2. *$x \in out_4(G, A)$ if and only if $x \in Cn(G(\mathcal{L}))$ and $G^{\Box} \cup A \vdash_{\mathbf{S}} \Box x$ for any modal logic **S** with $\boldsymbol{K_0T} \subseteq \boldsymbol{S} \subseteq \boldsymbol{KT45}$.*

The embeddings have been encoded in Isabelle/HOL to enable experiments for deontic reasoning frameworks. We have examined an application of General Data Protection Regulation (GDPR) and Moral luck as a practical examples [154] in our implementation. The experiments with this environment provide evidence that the logic's implementation fruitfully enables interactive and automated reasoning at the meta-level and the object level.

Direct approach There are precise semantical embeddings of Boolean algebras into HOL. By building the I/O mechanism over Boolean algebras, we can make a direct embedding of the I/O mechanism in HOL in a straightway.

The chapter is structured as follows: The semantical embeddings of *Basic Output* and *Basic Reusable Output* in HOL are then devised and studied in Section 5.1. This section also shows the faithfulness (viz. soundness and completeness) of the embeddings. In Section 5.2, we use GDPR as a use-case example for our deontic reasoning framework. In Section 5.3, we apply the framework to the *Drink and Drive* example. In Section 5.4, we introduce our direct semantical embedding of I/O mechanism over Boolean algebras into HOL, that includes soundness and completeness (faithfulness). In Section 5.5, the direct semantical embedding is implemented in Isabelle/HOl. Section 5.6 concludes the chapter.

5.1 Semantical Embedding of I/O Logic in HOL

This section presents shallow semantical embeddings of the I/O operators out_2 and out_4 in HOL and provides proofs for the soundness and completeness of both operators. To realize these embeddings, we use the provided modal formulations of the operators.

Given a finite set of conditional norms G and an input set A of propositional formulas. For the embeddings of out_2 and out_4, we first use the corresponding translation into modal logic and afterwards we apply above theorem, respectively, in order to prove faithfulness. First we need to mention the theorem about faithful embedding of System **KT** in HOL.

Theorem 5.3 (Faithfulness of the embedding of system **KT** in HOL)**.**

$$\models_{\mathcal{C}_{KT}} \varphi \textit{ if and only if } \{REF\} \models^{HOL} vld\,\lfloor\varphi\rfloor$$

Proof. See [39, 40] for a proof of the faithfulness of the embedding of modal logic **K** in HOL. In system **KT** the class of Kripke models satisfies the property of *reflexivity*, which corresponds to axiom **T**. The higher-order counter part of this property is represented as $REF : \forall X_i (r_{i \to i \to o} X_i X_i)$ and has to be satisfied by the constant $r_{i \to i \to o}$. The result for logic **KT** follows as a simple corollary; see also Section 3.3 in [35]. □

Theorem 5.4 (Faithfulness of the embedding of out_2 in HOL)**.**

$$\varphi \in out_2(G, A)$$

if and only if

$$\models^{HOL} vld \lfloor \bigwedge (G^{\Box} \cup A) \to \Box\varphi \rfloor \; \textit{and} \; \models^{HOL} vld \lfloor \bigwedge G(\mathcal{L}) \to \varphi \rfloor$$

Proof. We choose $\mathbf{S} = \mathbf{K}$ in Theorem 5.1 and then apply Theorem 2.4.

$$\varphi \in out_2(G, A)$$

if and only if

$$G^{\Box} \cup A \vdash_{\mathbf{K}} \Box\varphi \text{ and } \varphi \in Cn(G(\mathcal{L}))$$

if and only if

$$\models_{\mathcal{C}_K} \bigwedge (G^{\Box} \cup A) \to \Box\varphi \text{ and } \bigwedge G(\mathcal{L}) \vdash \varphi$$

if and only if

$$\models_{\mathcal{C}_K} \bigwedge (G^{\Box} \cup A) \to \Box\varphi \text{ and } \models_{\mathcal{C}_K} \bigwedge G(\mathcal{L}) \to \varphi$$

if and only if

$$\models^{HOL} vld \lfloor \bigwedge (G^{\Box} \cup A) \to \Box\varphi \rfloor \text{ and } \models^{HOL} vld \lfloor \bigwedge G(\mathcal{L}) \to \varphi \rfloor$$

□

Theorem 5.5 (Faithfulness of the embedding of out_4 in HOL)**.**

$$\varphi \in out_4(G, A)$$

if and only if

$$\{REF\} \models^{HOL} vld \lfloor \bigwedge (G^{\Box} \cup A) \to \Box\varphi \rfloor \; \textit{and} \; \{REF\} \models^{HOL} vld \lfloor \bigwedge G(\mathcal{L}) \to \varphi \rfloor$$

Proof. We choose $\mathbf{S} = \mathbf{KT}$ in Theorem 5.2 and then apply Theorem 5.3.

$$\varphi \in out_4(G, A)$$

if and only if

$$G^{\Box} \cup A \vdash_{\mathbf{KT}} \Box\varphi \text{ and } \varphi \in Cn(G(\mathcal{L}))$$

if and only if

$$\models_{\mathcal{C}_{KT}} \bigwedge(G^{\Box} \cup A) \to \Box\varphi \text{ and } \bigwedge G(\mathcal{L}) \vdash \varphi$$

if and only if

$$\models_{\mathcal{C}_{KT}} \bigwedge(G^{\Box} \cup A) \to \Box\varphi \text{ and } \models_{\mathcal{C}_{KT}} \bigwedge G(\mathcal{L}) \to \varphi$$

if and only if

$$\{REF\} \models^{HOL} vld \lfloor \bigwedge(G^{\Box} \cup A) \to \Box\varphi \rfloor \text{ and } \{REF\} \models^{HOL} vld \lfloor \bigwedge G(\mathcal{L}) \to \varphi \rfloor$$

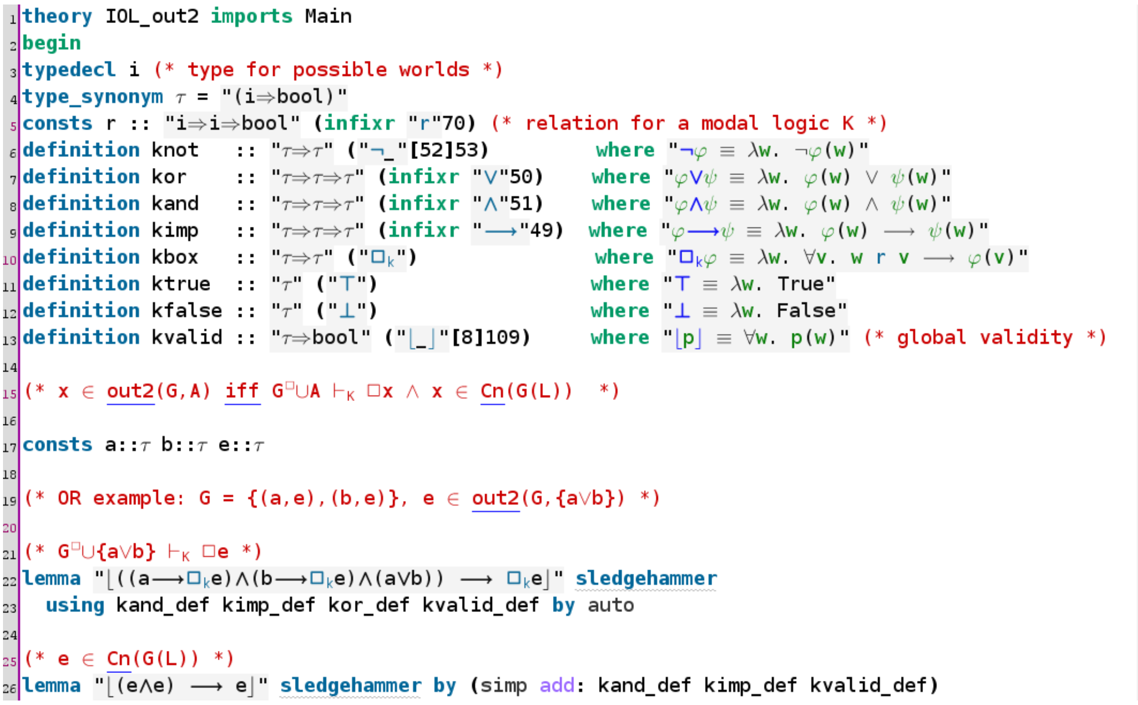
□

5.1.1 Implementation of I/O logic in Isabelle/HOL

The semantical embeddings of the operations out_2 and out_4 in HOL have been implemented in the higher-order proof assistant tool Isabelle/HOL [156], see Fig. 5.1. We declare the type i to denote possible worlds and introduce the relevant connectives in lines 6–12.

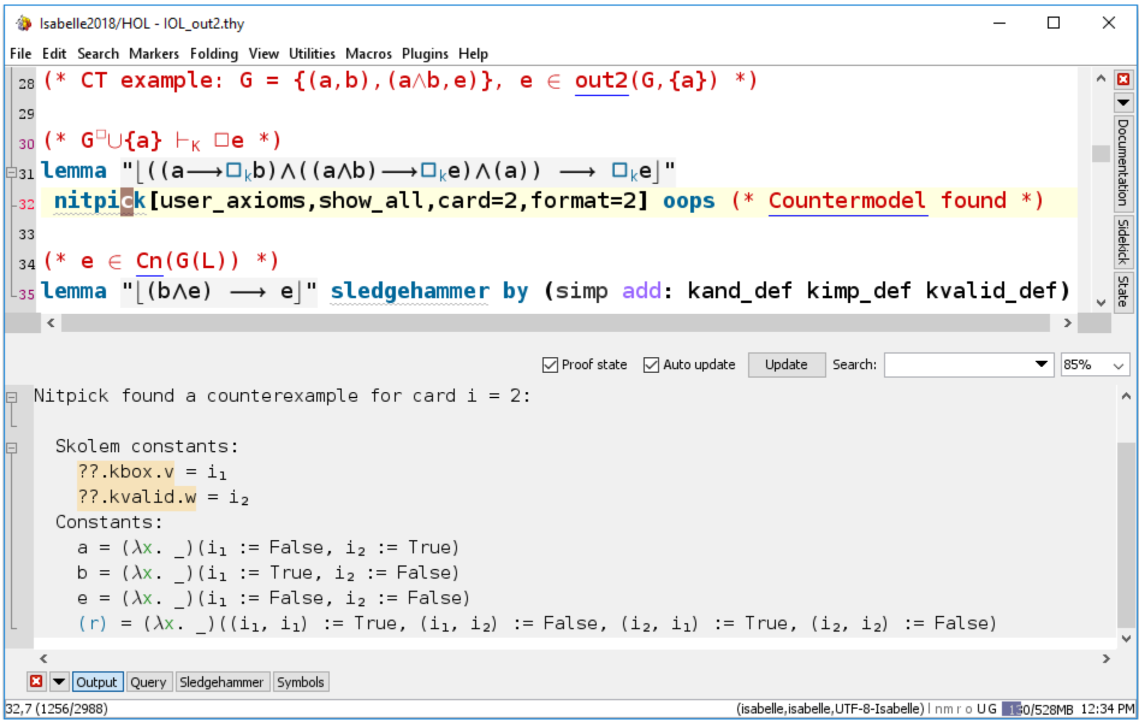
Let the set of conditional norms G be composed of the elements (a, e) and (b, e), where a, b and e are propositional symbols, and let the input set A correspond to the singleton set containing $a \vee b$. By the rule of disjunction (OR), we should have that $e \in out_2(G, A)$. According to the provided translation, $e \in out_2(G, A)$ if and only if $G^{\Box} \cup A \vdash_{\mathbf{K}} \Box e$ and $e \in Cn(G(\mathcal{L}))$. Theorem 5.4 provides us now with higher-order formulations for both of these statements, i.e., $\models^{HOL} vld \lfloor \bigwedge(G^{\Box} \cup A) \to \Box e \rfloor$ and $\models^{HOL} vld \lfloor \bigwedge G(\mathcal{L}) \to e \rfloor$, respectively. Regarding the implementation, the propositional symbols a, b and e have to be declared as constants of type τ. The framework's integrated automatic theorem provers (ATPs), called via the Sledgehammer tool [45], are able to prove both statements. This is shown in Fig. 5.1, lines 22–23 and 26.

Consider the set of conditional norms $G = \{(a, b), (a \wedge b, e)\}$ with the input set $A = \{a\}$. The rule of cumulative transitivity (CT) is not satisfied by the operation out_2. This can also be verified with our implementation. The model finder Nitpick [44] is able to generate



```
theory IOL_out2 imports Main
begin
typedecl i (* type for possible worlds *)
type_synonym τ = "(i⇒bool)"
consts r :: "i⇒i⇒bool" (infixr "r"70) (* relation for a modal logic K *)
definition knot   :: "τ⇒τ" ("¬_"[52]53)          where "¬φ ≡ λw. ¬φ(w)"
definition kor    :: "τ⇒τ⇒τ" (infixr "∨"50)     where "φ∨ψ ≡ λw. φ(w) ∨ ψ(w)"
definition kand   :: "τ⇒τ⇒τ" (infixr "∧"51)     where "φ∧ψ ≡ λw. φ(w) ∧ ψ(w)"
definition kimp   :: "τ⇒τ⇒τ" (infixr "⟶"49)    where "φ⟶ψ ≡ λw. φ(w) ⟶ ψ(w)"
definition kbox   :: "τ⇒τ" ("□ₖ")                 where "□ₖφ ≡ λw. ∀v. w r v ⟶ φ(v)"
definition ktrue  :: "τ" ("⊤")                    where "⊤ ≡ λw. True"
definition kfalse :: "τ" ("⊥")                    where "⊥ ≡ λw. False"
definition kvalid :: "τ⇒bool" ("⌊_⌋"[8]109)       where "⌊p⌋ ≡ ∀w. p(w)" (* global validity *)

(* x ∈ out2(G,A) iff G□∪A ⊢K □x ∧ x ∈ Cn(G(L))  *)

consts a::τ b::τ e::τ

(* OR example: G = {(a,e),(b,e)}, e ∈ out2(G,{a∨b}) *)

(* G□∪{a∨b} ⊢K □e *)
lemma "⌊((a⟶□ₖe)∧(b⟶□ₖe)∧(a∨b)) ⟶ □ₖe⌋" sledgehammer
  using kand_def kimp_def kor_def kvalid_def by auto

(* e ∈ Cn(G(L)) *)
lemma "⌊(e∧e) ⟶ e⌋" sledgehammer by (simp add: kand_def kimp_def kvalid_def)
```

Figure 5.1: Semantical embedding of out_2 in Isabelle/HOL



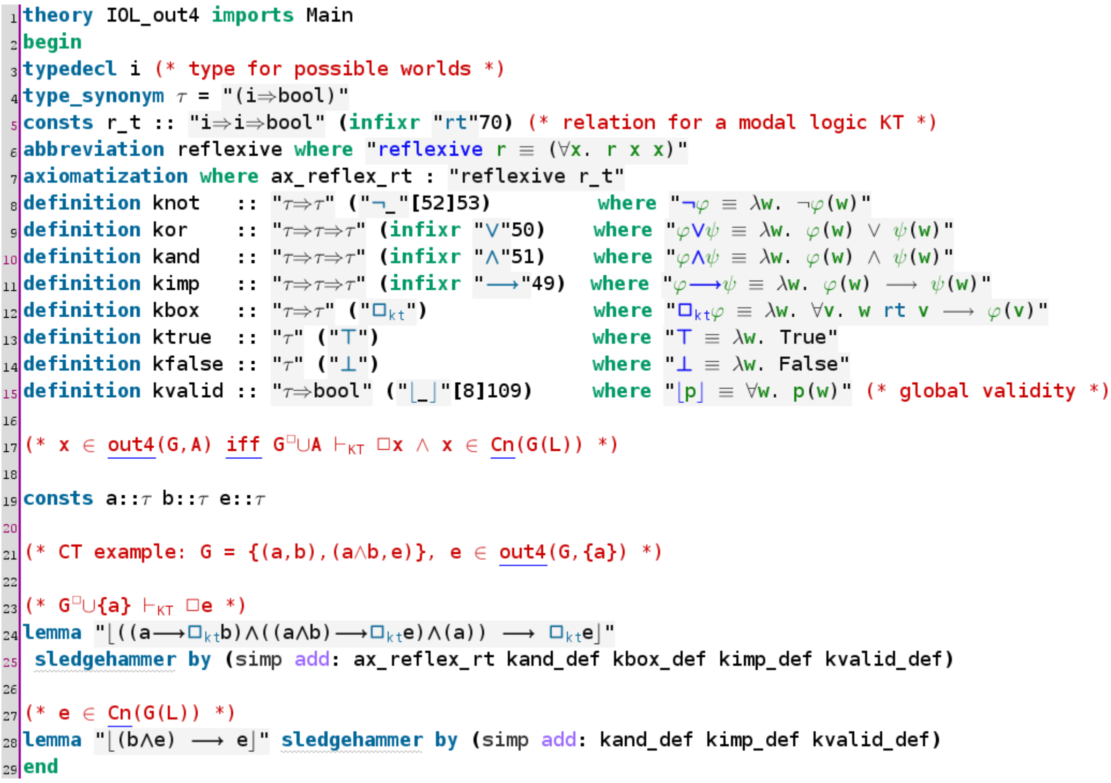
Isabelle2018/HOL - IOL_out2.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
Documentation
Sidekick
State
☑ Proof state ☑ Auto update Update Search: 85%
Output Query Sledgehammer Symbols
32,7 (1256/2988)
(isabelle,isabelle,UTF-8-Isabelle) | nmro UG 130/528MB 12:34 PM

```
(* CT example: G = {(a,b),(a∧b,e)}, e ∈ out2(G,{a}) *)

(* G□∪{a} ⊢K □e *)
lemma "⌊((a⟶□ₖb)∧((a∧b)⟶□ₖe)∧(a)) ⟶ □ₖe⌋"
 nitpick[user_axioms,show_all,card=2,format=2] oops (* Countermodel found *)

(* e ∈ Cn(G(L)) *)
lemma "⌊(b∧e) ⟶ e⌋" sledgehammer by (simp add: kand_def kimp_def kvalid_def)
```

```
Nitpick found a counterexample for card i = 2:

  Skolem constants:
    ??.kbox.v = i₁
    ??.kvalid.w = i₂
  Constants:
    a = (λx. _)(i₁ := False, i₂ := True)
    b = (λx. _)(i₁ := True, i₂ := False)
    e = (λx. _)(i₁ := False, i₂ := False)
    (r) = (λx. _)((i₁, i₁) := True, (i₁, i₂) := False, (i₂, i₁) := True, (i₂, i₂) := False)
```

Figure 5.2: Failure of CT for out_2

a countermodel for the statement $G^{\Box} \cup A \vdash_{\mathbf{K}} \Box e$ and therefore we were able to show that $e \notin out_2(G, A)$. In particular, Nitpick came up with a model M consisting of two possible worlds i_1 and i_2. We have that $V(a) = \{i_2\}$, $V(b) = \{i_1\}$ and $V(e) = \emptyset$. And $R = \{(i_1, i_1), (i_2, i_1)\}$. The formula $((a \to \Box b) \wedge ((a \wedge b) \to \Box e) \wedge a) \to \Box e$ is not valid in this model. The formulation of the example and the generation of the countermodel is illustrated in Fig. 5.2.



```
theory IOL_out4 imports Main
begin
typedecl i (* type for possible worlds *)
type_synonym τ = "(i⇒bool)"
consts r_t :: "i⇒i⇒bool" (infixr "rt"70) (* relation for a modal logic KT *)
abbreviation reflexive where "reflexive r ≡ (∀x. r x x)"
axiomatization where ax_reflex_rt : "reflexive r_t"
definition knot   :: "τ⇒τ" ("¬_"[52]53)        where "¬φ ≡ λw. ¬φ(w)"
definition kor    :: "τ⇒τ⇒τ" (infixr "∨"50)    where "φ∨ψ ≡ λw. φ(w) ∨ ψ(w)"
definition kand   :: "τ⇒τ⇒τ" (infixr "∧"51)    where "φ∧ψ ≡ λw. φ(w) ∧ ψ(w)"
definition kimp   :: "τ⇒τ⇒τ" (infixr "⟶"49)   where "φ⟶ψ ≡ λw. φ(w) ⟶ ψ(w)"
definition kbox   :: "τ⇒τ" ("□kt")              where "□ktφ ≡ λw. ∀v. w rt v ⟶ φ(v)"
definition ktrue  :: "τ" ("⊤")                  where "⊤ ≡ λw. True"
definition kfalse :: "τ" ("⊥")                  where "⊥ ≡ λw. False"
definition kvalid :: "τ⇒bool" ("⌊_⌋"[8]109)     where "⌊p⌋ ≡ ∀w. p(w)" (* global validity *)

(* x ∈ out4(G,A) iff G□∪A ⊢KT □x ∧ x ∈ Cn(G(L)) *)

consts a::τ b::τ e::τ

(* CT example: G = {(a,b),(a∧b,e)}, e ∈ out4(G,{a}) *)

(* G□∪{a} ⊢KT □e *)
lemma "⌊((a⟶□ktb)∧((a∧b)⟶□kte)∧(a)) ⟶ □kte⌋"
 sledgehammer by (simp add: ax_reflex_rt kand_def kbox_def kimp_def kvalid_def)

(* e ∈ Cn(G(L)) *)
lemma "⌊(b∧e) ⟶ e⌋" sledgehammer by (simp add: kand_def kimp_def kvalid_def)
end
```

FIGURE 5.3: Semantical embedding of out_4 in Isabelle/HOL

The embedding of the operation out_4 refers to system **KT** which means that the corresponding class of Kripke models satisfies the property of reflexivity. In our implementation, the accessibility relation for this system is denoted by the constant r_t which we declare as reflexive. Due to this property, the Sledgehammer tool is able to prove the statement $G^{\Box} \cup A \vdash_{\mathbf{KT}} \Box e$ and thus we can verify that $e \in out_4(G, A)$. Fig. 5.3 shows the encoding of the operation out_4 in Isabelle/HOL and the verification of the CT example.

5.2 Application in Legal Reasoning

The paper [36] already documents some practical experiments of automated I/O logic in the domain of legal reasoning. In particular, General Data Protection Regulation (GDPR, Regulation EU 2016/679) is used as an application scenario. By this regulation, the European Parliament, the Council of the European Union and the European Commission aim to strengthen and unify data protection for all the individuals within the European Union. The following two norms are part of GDPR:

1. Personal data shall be processed lawfully (Art. 5). For example, the data subject must have given consent to the processing of his or her personal data for one or more specific purposes (Art. 6/1.a).
2. If the personal data have been processed unlawfully (none of the requirements for a lawful processing applies), the controller has the obligation to erase the personal data in question without delay (Art. 17.d, right to be forgotten).

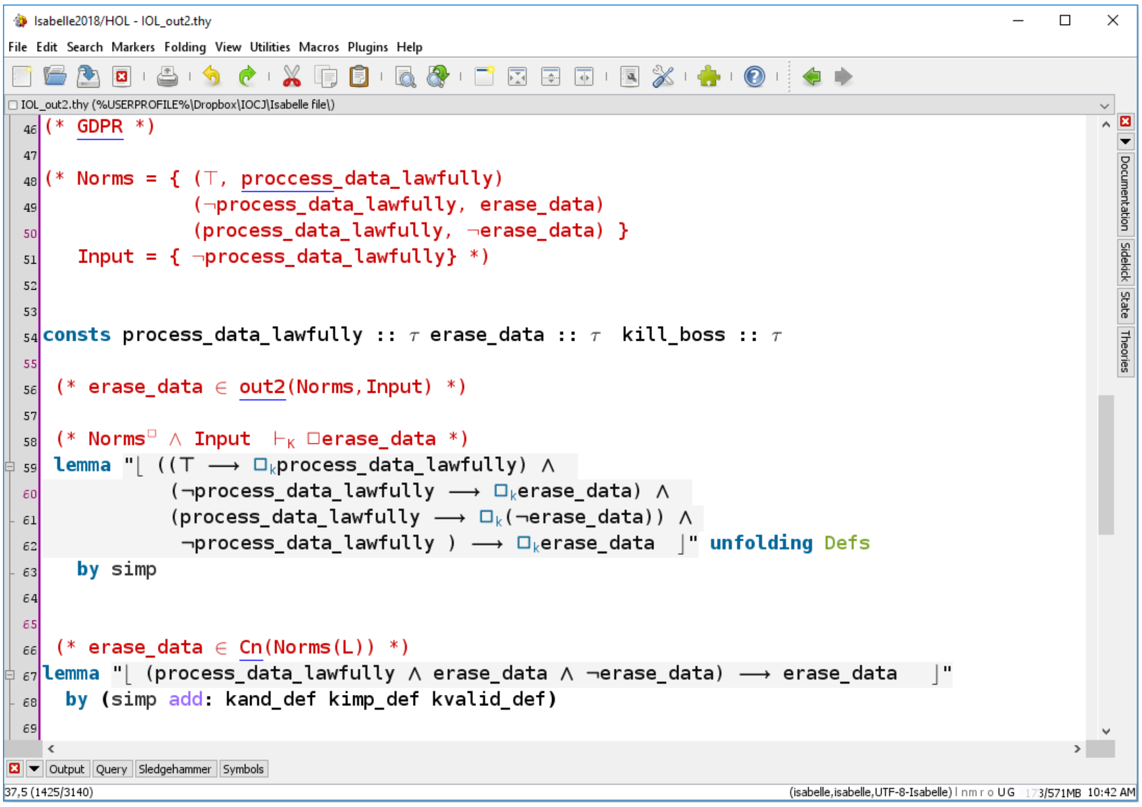
The authors [36] added the following two knowledge units. This establishes a typical contrary-to-duty (CTD) scenario which was then analyzed in the context of I/O logic.

3. It is obligatory (e.g. as part of a respective agreement between a customer and a company) to keep the personal data (as relevant to the agreement) provided that it is processed lawfully.
4. Some data in the context of such an agreement has been processed unlawfully.

We formulated this GDPR scenario in Isabelle/HOL as an application scenario for the embedded out_2 operator; cf. Figure 5.4. The lines 48-50 show the set of *Norms* which is composed of:

- $(\top, process\_data\_lawfully)$
  This norm states that it is obligatory to process data lawfully.
- $(\neg process\_data\_lawfully, erase\_data)$
  This norms states that if the data was not processed lawfully then it is obligatory to erase the data.
- $(process\_data\_lawfully, \neg erase\_data)$
  This norms states that if the data has been processed lawfully then it is obligatory to keep the data.

Line 51 shows the *Input* set. We assume a situation where the data has not been processed lawfully, formally $Input = \{\neg process\_data\_lawfully\}$. At line 54, we introduce


Isabelle2018/HOL - IOL_out2.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
IOL_out2.thy (%USERPROFILE%\Dropbox\IOCJ\Isabelle file\)
Documentation
Sidekick
State
Theories
Output
Query
Sledgehammer
Symbols
37,5 (1425/3140)
(isabelle,isabelle,UTF-8-Isabelle) | nm r o UG 173/571MB 10:42 AM

```
46 (* GDPR *)
47
48 (* Norms = { (⊤, proccess_data_lawfully)
49              (¬process_data_lawfully, erase_data)
50              (process_data_lawfully, ¬erase_data) }
51    Input = { ¬process_data_lawfully} *)
52
53
54 consts process_data_lawfully :: τ erase_data :: τ  kill_boss :: τ
55
56  (* erase_data ∈ out2(Norms,Input) *)
57
58  (* Norms□ ∧ Input  ⊢K □erase_data *)
59  lemma "⌊ ((⊤ ⟶ □_kprocess_data_lawfully) ∧
60             (¬process_data_lawfully ⟶ □_kerase_data) ∧
61             (process_data_lawfully ⟶ □_k(¬erase_data)) ∧
62              ¬process_data_lawfully ) ⟶ □_kerase_data  ⌋" unfolding Defs
63    by simp
64
65
66  (* erase_data ∈ Cn(Norms(L)) *)
67 lemma "⌊ (process_data_lawfully ∧ erase_data ∧ ¬erase_data) ⟶ erase_data   ⌋"
68   by (simp add: kand_def kimp_def kvalid_def)
69
```

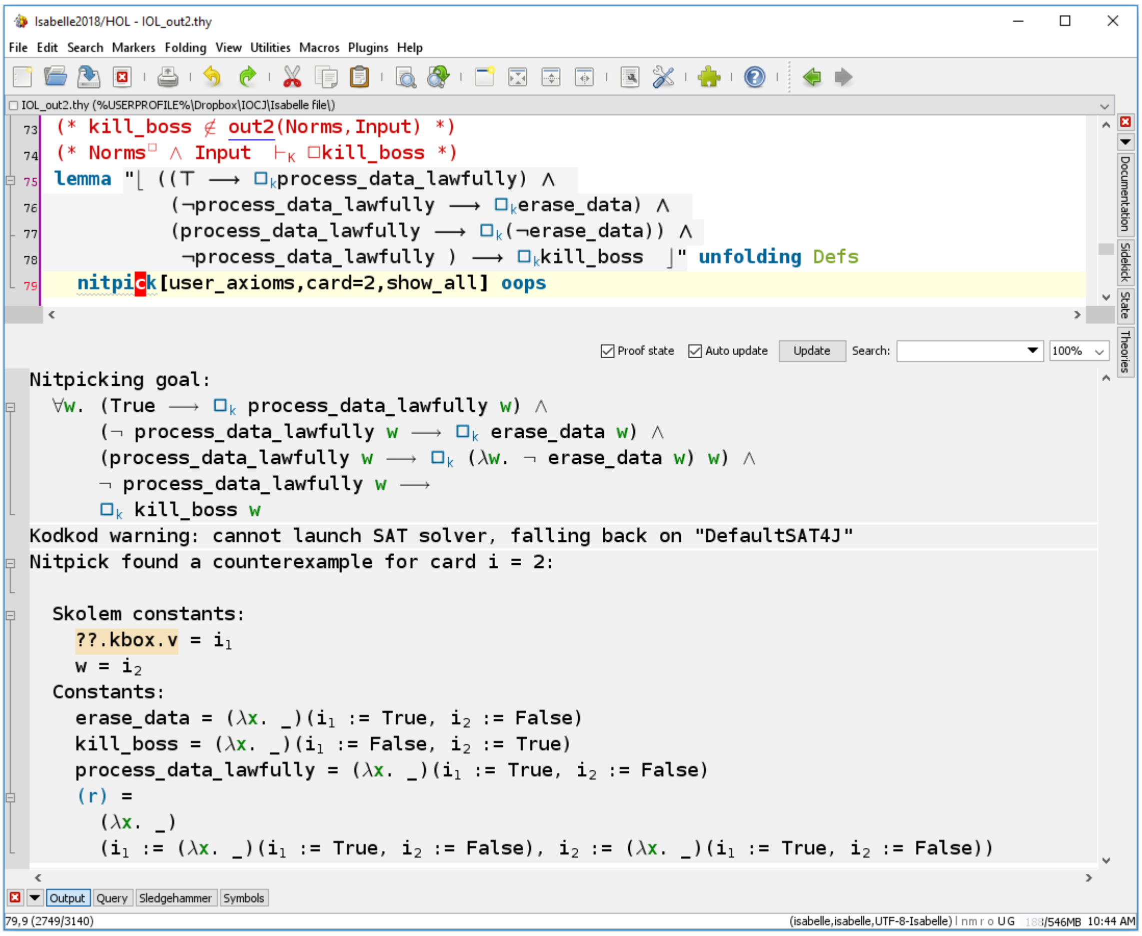
FIGURE 5.4: GDPR scenario in Isabelle/HOL

the constants symbols which are representing the propositions. Just like for the previous examples, each proposition is declared as a constant of type τ. Next, we want to check that $erase\_data$ is outputted in the context of $\neg process\_data\_lawfully$, meaning that the data should be erased in the situation when the data has not been processed lawfully. So we have to verify that $erase\_data \in out_2(Norms, Input)$. By the modal translation of this operator, we have to check that $Norms^{\Box} \cup Input \vdash_{\mathbf{K}} \Box erase\_data$ and $earse\_data \in Cn(Norms(\mathcal{L}))$. The lines 59-62 and 67 show the formulations for those statements, respectively. Both of them could be by proven by the integrated ATPs of Isabelle/HOL.

Like in a related paper [36], we also showed that we are not able to derive any weird or unethical conclusions such as killing the boss, cf. Figure 5.5. The model finder Nitpick is able to generate a countermodel for the following statement:

$$Norms^{\Box} \cup Input \vdash_{\mathbf{K}} \Box kill\_boss$$

Therefore it showed that $kill\_boss \notin out_2(Norms, Input)$. Nitpick found a model M consisting of two possible worlds i_1 and i_2. For the valuation function V, we have that


Isabelle2018/HOL - IOL_out2.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
IOL_out2.thy (%USERPROFILE%\Dropbox\IOCJ\Isabelle file\)
Documentation
Sidekick
State
Theories
☑ Proof state
☑ Auto update
Update
Search:
100%
Output
Query
Sledgehammer
Symbols
79,9 (2749/3140)
(isabelle,isabelle,UTF-8-Isabelle) | nm r o UG 188/546MB 10:44 AM

```
(* kill_boss ∉ out2(Norms,Input) *)
(* Norms□ ∧ Input  ⊢K □kill_boss *)
lemma "⌊ ((⊤ ⟶ □ₖprocess_data_lawfully) ∧
           (¬process_data_lawfully ⟶ □ₖerase_data) ∧
           (process_data_lawfully ⟶ □ₖ(¬erase_data)) ∧
            ¬process_data_lawfully ) ⟶ □ₖkill_boss  ⌋" unfolding Defs
  nitpick[user_axioms,card=2,show_all] oops
```

```
Nitpicking goal:
  ∀w. (True ⟶ □ₖ process_data_lawfully w) ∧
      (¬ process_data_lawfully w ⟶ □ₖ erase_data w) ∧
      (process_data_lawfully w ⟶ □ₖ (λw. ¬ erase_data w) w) ∧
      ¬ process_data_lawfully w ⟶
      □ₖ kill_boss w
Kodkod warning: cannot launch SAT solver, falling back on "DefaultSAT4J"
Nitpick found a counterexample for card i = 2:

  Skolem constants:
    ??.kbox.v = i₁
    w = i₂
  Constants:
    erase_data = (λx. _)(i₁ := True, i₂ := False)
    kill_boss = (λx. _)(i₁ := False, i₂ := True)
    process_data_lawfully = (λx. _)(i₁ := True, i₂ := False)
    (r) =
      (λx. _)
      (i₁ := (λx. _)(i₁ := True, i₂ := False), i₂ := (λx. _)(i₁ := True, i₂ := False))
```

FIGURE 5.5: Further experiments with the GDPR scenario in Isabelle/HOL

$V(erase\_data) = \{i_1\}$, $V(proccess\_data\_lawfully) = \{i_1\}$ and $V(kill\_boss) = \{i_2\}$ and for the relation R, we have that $R = \{(i_1, i_1), (i_2, i_1)\}$. The world i_2 satisfies the following formulas:

- $\top \rightarrow \Box process\_data\_lawfully$
- $\neg process\_data\_lawfully \rightarrow \Box erase\_data$
- $process\_data\_lawfully \rightarrow \Box\neg erase\_data$
- $\neg process\_data\_lawfully$

However, the formula $\Box kill\_boss$ is not satisfied in the possible world i_2.

5.3 Application in Moral Luck

The literature on moral luck [154] is addressing the question whether luck can ever make a moral difference or not. Examples involving moral luck are typical scenarios in which an agent is held accountable for his actions and its consequences even though it is clear that the agent was neither in full control of his actions nor its consequences. These examples are thus in conflict with the ethical principle that agents are not morally responsible for actions that they are unable to control.

The *Drink and Drive* [150] example highlights a classical scenario of moral luck. There exist many different variations of this example and a possible variant can be formulated as follows:

> Assume a situation where two persons, Ali and Paul, go out for a drink in the evening. Both of them go to the same bar, consume the same amount of alcoholic drinks and end up pretty drunk. At one point during the night, they both decide to leave the place. So they go to their own individual vehicles and hit the road in order to drive home. The roads are pretty deserted at that time and Ali manages to drive home safely even with the high percentage of alcohol in his blood. Paul, in contrast, is facing something unexpected. Out of nowhere, a child appears in front of his car. Since he had a few drinks too much, his reaction time is impaired by the alcohol and it makes it impossible for him to stop and swerve to avoid hitting and killing the child.

Both *Ali* and *Paul*, made the blameworthy decision of driving while being drunk. But neither one of them had the intention to hit and kill anyone. Nevertheless, most people would tend to judge *Paul* more guilty than *Ali* simply because in his case a child got killed. However, both of them violated the same obligation, namely that one should not drive while being intoxicated and it was only a matter of luck that nobody got harmed or killed in the case of *Ali*. Therefore we say that *Ali* got morally lucky.

To formulate the *Drink and Drive* example in Isabelle/HOL, we first import the Isabelle/HOL file containing the implementation of the operation out_2. This can be done using the Isabelle/HOL command *imports* (cf. Fig. 5.6, line 1). Next, we have to declare three individuals, namely *Ali* and *Paul*, representing the two drivers, and *Child*, representing the *child* in the scenario (cf. Fig. 5.6, line 3). In lines 4–5, we define the constant symbols for the relevant propositions (state of affairs or action).

One associates with each driver a set of *Norms* and an *Input*. For the individual *Paul*, the set of *Norms G* is defined as follows: (cf. Fig. 5.6, lines 9–11)

- $(\top, \neg Kill\,Child \wedge \neg Hurt\,Child)$
  This norm states that it is forbidden to kill or even hurt the *child.*
- $(\top, Drive\_carefully\,Paul)$
  This norm states that *Paul* is obligated to drive carefully in any situation.
- $(\neg Drive\_carefully\,Paul, Stay\,Paul)$
  This norm states that if *Paul* does not drive carefully, he should stay (at his current location).

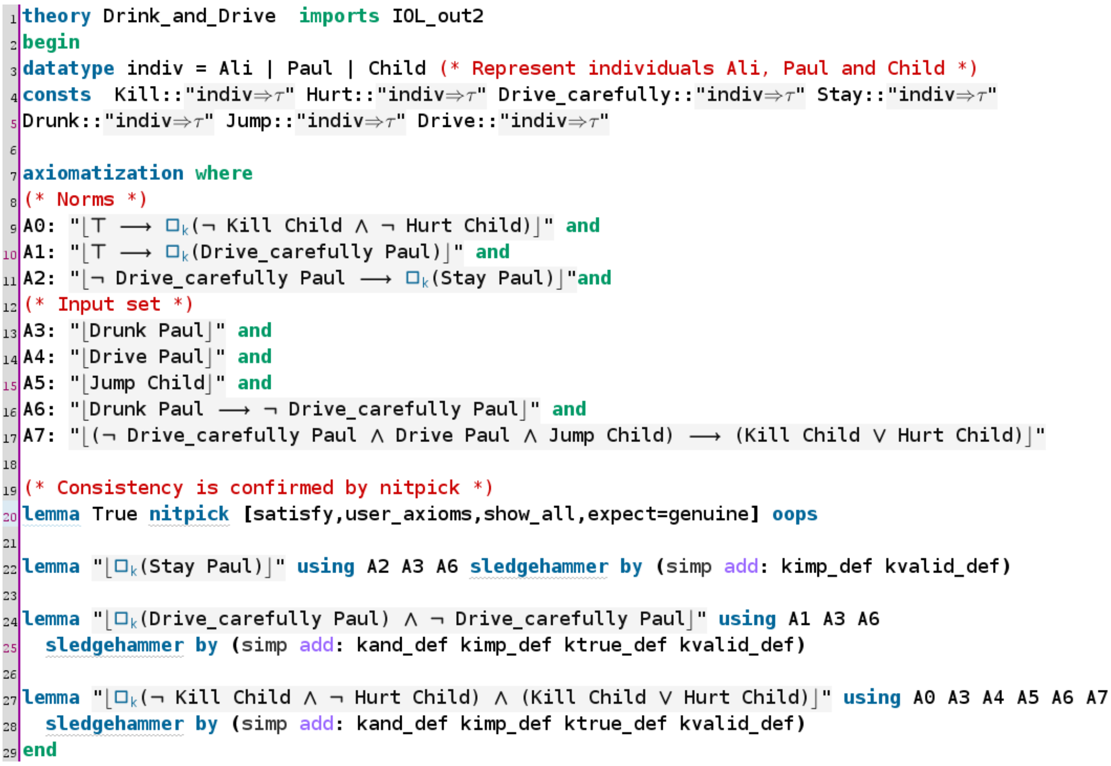
To complete the formalization for the individual *Paul*, we need to add the following facts to the *Input* set A: (cf. Fig. 5.6, lines 13–17)

- $Drunk\,Paul$; $Drive\,Paul$; $Jump\,Child$
  Paul is actually drunk; *Paul* drives home; The *child* jumps.
- $Drunk\,Paul \rightarrow \neg Drive\_carefully\,Paul$
  If *Paul* is drunk then he drives not carefully.
- $(\neg Drive\_carefullyPaul \wedge DrivePaul \wedge JumpChild)$
  $\rightarrow (KillChild \vee HurtChild)$
  If *Paul* drives, but does not do it carefully, and the *child* jumps in front of his car then *Paul* will kill or hurt the *child.*

Since Nitpick finds a model satisfying our statements, the formalization of the *Drink and Drive* is consistent; cf. Fig. 5.6, line 20.

Actually, we are able to derive the obligation that *Paul* should stay (at his current position) by using the norm A2 and the facts A3 and A6, meaning that we can derive $G^{\Box} \cup A \vdash_{\mathbf{K}} \Box Stay\_Paul$ and $Stay\_Paul \in Cn(G(\mathcal{L}))$. The first statement is proven by Sledgehammer tool; cf. Fig. 5.6, line 22. In this example, we skip checking the following (trivial) statements $X \in Cn(G(\mathcal{L}))$ for $X \in \{Stay\_Paul, Drive\_carefully\_Paul, \neg KillChild \wedge \neg HurtChild\}$ in Isabelle/HOL.

Furthermore, our implementation is capable of recognizing violations to norms, formally written as $\alpha \in out_2(G, A)$ and $\neg\alpha \in Cn(A)$. In particular, *Paul* violated the norms A0 and A1. For instance, the violation to A1 is proven by Sledgehammer; cf. Fig. 5.6, lines 24–25. *Paul* did not drive carefully even though there is an obligation to do so, meaning that we have $G^{\Box} \cup A \vdash_{\mathbf{K}} \Box Drive\_carefully\_Paul$ (using A1), $Drive\_carefully\_Paul \in Cn(G(\mathcal{L}))$ and $\neg Drive\_carefully\_Paul \in Cn(A)$ (using A3 and A6).

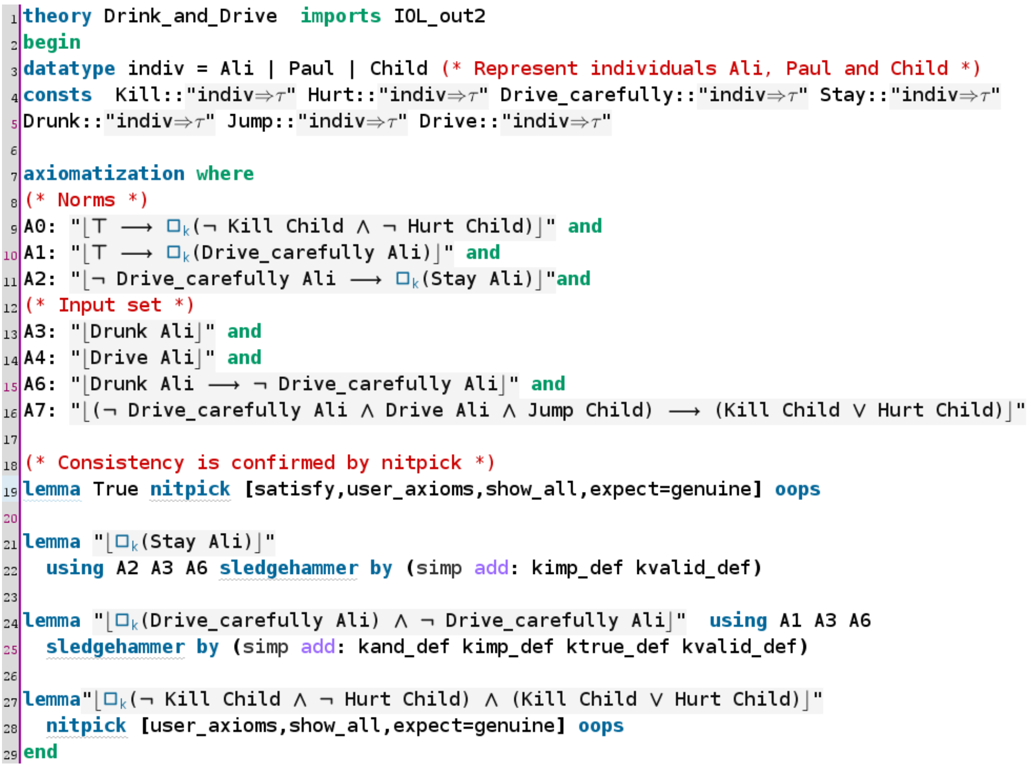


```
theory Drink_and_Drive  imports IOL_out2
begin
datatype indiv = Ali | Paul | Child (* Represent individuals Ali, Paul and Child *)
consts  Kill::"indiv⇒τ" Hurt::"indiv⇒τ" Drive_carefully::"indiv⇒τ" Stay::"indiv⇒τ"
Drunk::"indiv⇒τ" Jump::"indiv⇒τ" Drive::"indiv⇒τ"

axiomatization where
(* Norms *)
A0: "⌊⊤ ⟶ □k(¬ Kill Child ∧ ¬ Hurt Child)⌋" and
A1: "⌊⊤ ⟶ □k(Drive_carefully Paul)⌋" and
A2: "⌊¬ Drive_carefully Paul ⟶ □k(Stay Paul)⌋"and
(* Input set *)
A3: "⌊Drunk Paul⌋" and
A4: "⌊Drive Paul⌋" and
A5: "⌊Jump Child⌋" and
A6: "⌊Drunk Paul ⟶ ¬ Drive_carefully Paul⌋" and
A7: "⌊(¬ Drive_carefully Paul ∧ Drive Paul ∧ Jump Child) ⟶ (Kill Child ∨ Hurt Child)⌋"

(* Consistency is confirmed by nitpick *)
lemma True nitpick [satisfy,user_axioms,show_all,expect=genuine] oops

lemma "⌊□k(Stay Paul)⌋" using A2 A3 A6 sledgehammer by (simp add: kimp_def kvalid_def)

lemma "⌊□k(Drive_carefully Paul) ∧ ¬ Drive_carefully Paul⌋" using A1 A3 A6
  sledgehammer by (simp add: kand_def kimp_def ktrue_def kvalid_def)

lemma "⌊□k(¬ Kill Child ∧ ¬ Hurt Child) ∧ (Kill Child ∨ Hurt Child)⌋" using A0 A3 A4 A5 A6 A7
  sledgehammer by (simp add: kand_def kimp_def ktrue_def kvalid_def)
end
```

FIGURE 5.6: *Drink and Drive* scenario for *Paul* in Isabelle/HOL

For the individual *Ali*, the set of *Norms* remains the same except that we adapted the name of the individual accordingly. However, in *Ali's* case, the *child* was not involved. Therefore, the *Input* set only consists of our facts: A3, A4, A6 and A7 (cf. Fig. 5.7, lines 13–16).

The formalization of *Ali's* scenario is consistent, again proven by Nitpick (cf. Figure 5.7, line 19). In contrast to *Paul*, *Ali* did not violated the norm A0 as Nitpick find a counter model for the corresponding statement (cf. Fig. 5.7, lines 27–28).



```
theory Drink_and_Drive  imports IOL_out2
begin
datatype indiv = Ali | Paul | Child (* Represent individuals Ali, Paul and Child *)
consts  Kill::"indiv⇒τ" Hurt::"indiv⇒τ" Drive_carefully::"indiv⇒τ" Stay::"indiv⇒τ"
Drunk::"indiv⇒τ" Jump::"indiv⇒τ" Drive::"indiv⇒τ"

axiomatization where
(* Norms *)
A0: "⌊⊤ ⟶ □ₖ(¬ Kill Child ∧ ¬ Hurt Child)⌋" and
A1: "⌊⊤ ⟶ □ₖ(Drive_carefully Ali)⌋" and
A2: "⌊¬ Drive_carefully Ali  ⟶ □ₖ(Stay Ali)⌋"and
(* Input set *)
A3: "⌊Drunk Ali⌋" and
A4: "⌊Drive Ali⌋" and
A6: "⌊Drunk Ali  ⟶ ¬ Drive_carefully Ali⌋" and
A7: "⌊(¬ Drive_carefully Ali ∧ Drive Ali ∧ Jump Child) ⟶ (Kill Child ∨ Hurt Child)⌋"

(* Consistency is confirmed by nitpick *)
lemma True nitpick [satisfy,user_axioms,show_all,expect=genuine] oops

lemma "⌊□ₖ(Stay Ali)⌋"
  using A2 A3 A6 sledgehammer by (simp add: kimp_def kvalid_def)

lemma "⌊□ₖ(Drive_carefully Ali) ∧ ¬ Drive_carefully Ali⌋"  using A1 A3 A6
  sledgehammer by (simp add: kand_def kimp_def ktrue_def kvalid_def)

lemma"⌊□ₖ(¬ Kill Child ∧ ¬ Hurt Child) ∧ (Kill Child ∨ Hurt Child)⌋"
  nitpick [user_axioms,show_all,expect=genuine] oops
end
```

Figure 5.7: *Drink and Drive* scenario for *Ali* in Isabelle/HOL

5.4 Semantical Embedding of I/O Mechanism over Boolean Algebras in HOL

The direct semantical embedding of I/O operations are based on Theorem 4.10.

Let X be a set of propositional variables and $N \subseteq Fm(X) \times Fm(X)$. It has been shown that (Theorem 4.10)

$$(\varphi, \psi) \in derive_i^{\mathbf{Fm}(X)}(N)$$

if and only if

$$V(\psi) \in out_i^{\mathcal{B}}(N^V, \{V(\varphi)\}) \text{ for every } \mathcal{B} \in \mathbf{BA}, \text{ for every valuation } V \text{ on } \mathcal{B}.$$

If we call the structure $\mathcal{N} = \langle \mathcal{B}, V, N^V \rangle$ a Boolean normative model, $(\varphi, \psi) \in derive_i^{\mathbf{Fm}(X)}(N)$ holds if and only if $V(\psi) \in out_i^{\mathcal{B}}(N^V, \{V(\varphi)\})$ holds in all Boolean normative models. In the reminder of this chapter we show how the direct embedding works for $\mathbf{Fm}(X)$. Type $i \rightarrow o$ is abbreviated as τ in the remainder. The HOL signature is assumed to contain the constant symbols $N_{i\to\tau}$, $\neg_{i\to i}$, $\vee_{i\to i\to i}$, $\wedge_{i\to i\to i}$, $\top_i$ and $\bot_i$. Moreover, for each atomic propositional symbol $p^j \in X$ of $\mathbf{Fm}(X)$, the HOL signature must contain a respective constant symbol p^j_i. Without loss of generality, we assume that besides those symbols and the primitive logical connectives of HOL, no other constant symbols are given in the signature of HOL.

The mapping $\lfloor\cdot\rfloor$ translates element $\varphi \in \mathbf{Fm}(X)$ into HOL terms $\lfloor\varphi\rfloor$ of type i. The mapping is recursively defined:

$$\begin{array}{lcll}
\lfloor p^j\rfloor & = & p^j_i & p^j \in X \\
\lfloor\top\rfloor & = & \top_i & \\
\lfloor\bot\rfloor & = & \bot_i & \\
\lfloor\neg\varphi\rfloor & = & \neg_{i\to i}(\lfloor\varphi\rfloor) & \\
\lfloor\varphi\vee\psi\rfloor & = & \vee_{i\to i\to i}\lfloor\varphi\rfloor\lfloor\psi\rfloor & \\
\lfloor\varphi\wedge\psi\rfloor & = & \wedge_{i\to i\to i}\lfloor\varphi\rfloor\lfloor\psi\rfloor & \\
\lfloor d_i(N)(\varphi,\psi)\rfloor & = & (\bigcirc_i(N)_{\tau\to\tau}\{\lfloor\varphi\rfloor\})\lfloor\psi\rfloor &
\end{array}$$

$\bigcirc_I(N)_{\tau\to\tau}$, $\bigcirc_{II}(N)_{\tau\to\tau}$, $\bigcirc_1(N)_{\tau\to\tau}$, $\bigcirc_2(N)_{\tau\to\tau}$ and $\bigcirc_3(N)_{\tau\to\tau}$ thereby abbreviate the following HOL terms:

$$\begin{array}{ll}
\bigcirc_I(N)_{\tau\to\tau} & = \lambda A_\tau \lambda X_i(\exists U\,(\exists Y\,(\exists Z\,(A\,Z \wedge Z = Y \wedge N\,Y\,U \wedge U \leq X)))) \\
\bigcirc_{II}(N)_{\tau\to\tau} & = \lambda A_\tau \lambda X_i(\exists U\,(\exists Y\,(\exists Z\,(A\,Z \wedge Z \leq Y \wedge N\,Y\,U \wedge U = X)))) \\
\bigcirc_1(N)_{\tau\to\tau} & = \lambda A_\tau \lambda X_i(\exists U\,(\exists Y\,(\exists Z\,(A\,Z \wedge Z \leq Y \wedge N\,Y\,U \wedge U \leq X))))
\end{array}$$

$$\begin{array}{ll}
\bigcirc_2(N)_{\tau\to\tau} & = \lambda A_\tau \lambda X_i(\forall V\,(\mathit{Saturated}\,V \wedge \forall U(A\,U \to V\,U) \\
& \to \exists Y\,(\exists Z\,(Z \leq X \wedge N\,Y\,Z \wedge V\,Y))))
\end{array}$$

$$\begin{array}{ll}
\bigcirc_3(N)_{\tau\to\tau} & = \lambda A_\tau \lambda X_i(\forall V(\forall U(A\,U \to V\,U) \wedge V = Up\,V \\
& \wedge\forall W(\exists Y(V\,Y \wedge NYW) \to VW) \\
& \quad\to \exists Y\,(\exists Z\,(Z \leq X \wedge N\,Y\,Z \wedge V\,Y))))
\end{array}$$

where

$$\begin{array}{ll}
\leq & = \lambda X_i \lambda Y_i(X_i \wedge_{i\to i\to i} Y_i = X_i) \\
\mathit{Saturated} & = \lambda A_\tau(\forall X\,\forall Y((A\,(X \vee Y)\, \to A\,X \vee A\,Y) \\
& \wedge(A\,X \wedge X \leq Y \to A\,Y))) \\
Up & = \lambda A_\tau \lambda X_i(\exists Z(A\,Z \wedge Z \leq X))
\end{array}$$

No more specification is needed for $N_{i\to\tau}$, $\neg_{i\to i}$, $\vee_{i\to i\to i}$, $\wedge_{i\to i\to i}$, $\top_i$ and $\bot_i$.

5.4.1 Soundness and completeness

To prove the soundness and completeness, that is, faithfulness, of the above embedding, a mapping from Boolean normative systems into Henkin models is employed.

Definition 5.6 (Henkin model $H^{\mathcal{N}}$ for Boolean normative model $\mathcal{N}$)**.** For any Boolean normative model $\mathcal{N} = \langle \mathcal{B}, V, N^V \rangle$, we define corresponding Henkin model $H^{\mathcal{N}}$. Thus, let a Boolean normative model $\mathcal{N} = \langle \mathcal{B}, V, N^V \rangle$ be given. Moreover, assume the finite set $X = \{p^1, , ..., p^m\}$, for $m \geq 1$, are the only atomic symbols of $\mathbf{Fm}(X)$. The embedding requires the corresponding signature of HOL to provide constant symbols p_i^j such that $\lfloor p^j \rfloor = p_i^j$

A Henkin model $H^{\mathcal{N}} = \langle \{D_\alpha\}_{\alpha \in T}, I \rangle$ for $\mathcal{N}$ is now defined as follows: D_i is chosen as the set of B ; all other sets $D_{\alpha \to \beta}$ are chosen as (not necessarily full) sets of functions from D_α to D_β. For all $D_{\alpha \to \beta}$ the rule that every term $t_{\alpha \to \beta}$ must have a denotation in $D_{\alpha \to \beta}$ must be obeyed (Denotatpflicht). In particular, it is required that D_τ, $D_{i \to \tau}$ and $D_{\tau \to \tau \to o}$ contain the elements Ip_i^j, $I\top_i$, $I\bot_i$, $I\neg_{i \to i}$, $I\vee_{i \to i \to i}$, $I\wedge_{i \to i \to i}$ and $IN_{i \to \tau}$. The interpretation function I of $H^{\mathcal{N}}$ is defined as follows:

1. For $j = 1, ..., m$, $Ip_i^j \in D_i$ is chosen such that $Ip_i^j = V(p^j)$ in $\mathcal{N}$.
2. $I\top_i \in D_i$ is chosen such that $I\top_i = V(\top)$ in $\mathcal{N}$.
3. $I\bot_i \in D_i$ is chosen such that $I\bot_i = V(\bot)$ in $\mathcal{N}$.
4. $I\neg_{i \to i} \in D_{i \to i}$ is chosen such that $I(\neg_{i \to i} \varphi) = \psi$ iff $\neg V(\varphi) = V(\psi)$ in $\mathcal{N}$.
5. $I\vee_{i \to i \to i} \in D_{i \to i \to i}$ is chosen such that $I \vee_{i \to i \to i} \varphi\psi = \phi$ iff $V(\varphi) \vee V(\psi) = V(\phi)$ in $\mathcal{N}$.
6. $I\wedge_{i \to i \to i} \in D_{i \to i \to i}$ is chosen such that $I \wedge_{i \to i \to i} \varphi\psi = \phi$ iff $V(\varphi) \wedge V(\psi) = V(\phi)$ in $\mathcal{N}$.
7. $IN_{i \to \tau} \in D_{i \to \tau}$ is chosen such that $IN_{i \to \tau}(\varphi, \psi) = T$ iff $(V(\varphi), V(\psi)) \in N^V$ in $\mathcal{N}$.
8. For the logical connectives $\neg$,$\wedge$, $\vee$, Π and $=$ of HOL the interpretation function I is defined as usual (see the previous section).

Existence of the valuation V, which is a Boolean homomorphism, from the Boolean algebra $\mathbf{Fm}(X)$ into the Boolean algebra $\mathcal{B}$ guarantee the existence of I and its mentioned requirements. Since we assume that there are no other symbols (besides the $\top_i$, $\bot_i$, $\neg_{i \to i}$, $\vee_{i \to i \to i}$, $\wedge_{i \to i \to i}$, $N_{i \to \tau}$ and; $\neg$, $\vee$, $\prod$ and $=$) in the signature of HOL, I is a total function. Moreover, the above construction guarantees that $H^{\mathcal{N}}$ is a Henkin model: $\langle D, I \rangle$ is a frame, and the choice of I in combination with the Denotatpflicht ensures that for arbitrary assignments g, $\|.\|^{H^M, g}$ is an total evaluation function.

Lemma 5.7. *Let $H^M = \langle \{D_\alpha\}_{\alpha\in T}, I\rangle$ be a Henkin model for a Boolean normative model $\mathcal{N}$. We have $H^{\mathcal{N}} \models^{HOL} \Sigma$ for all $\Sigma \in \{COM\vee, COM\wedge, ASS\vee, ASS\wedge, IDE\vee, IDE\wedge, COMP\vee, COMP\wedge, Dis\vee\wedge, Dis\wedge\vee\}$, where*

COM$\vee$	*is*	$\forall X_i Y_i\ (X \vee Y = Y \vee X)$
COM$\wedge$	*is*	$\forall X_i Y_i\ (X \wedge Y = Y \wedge X)$
ASS$\vee$	*is*	$\forall X_i Y_i Z_i\ (X \vee (Y \vee Z) = (X \vee Y) \vee Z)$
ASS$\wedge$	*is*	$\forall X_i Y_i Z_i\ (X \wedge (Y \wedge Z) = (X \wedge Y) \wedge Z)$
IDE$\vee$	*is*	$\forall X_i\ (X \vee \bot = X)$
IDE$\wedge$	*is*	$\forall X_i\ (X \wedge \top = X)$
COMP$\vee$	*is*	$\forall X_i\ (X \vee \neg X = \top)$
COMP$\wedge$	*is*	$\forall X_i\ (X \wedge \neg X = \bot)$
Dis$\vee\wedge$	*is*	$\forall X_i Y_i Z_i\ (X \vee (Y \wedge Z) = (X \vee Y) \wedge (X \vee Z))$
Dis$\wedge\vee$	*is*	$\forall X_i Y_i Z_i\ (X \wedge (Y \vee Z) = (X \wedge Y) \vee (X \wedge Z))$

Proof. The proof is straightforward, for example for COM $\vee$ we have

COM $\vee$:

$$\begin{array}{ll} & \text{For all } a, b \in D_i\text{: } I\vee_{i\to i\to i}\ a\,b = I\vee_{i\to i\to i}\ b\,a \qquad \text{(by definition of } I\vee_{i\to i\to i} \text{ and } \vee\text{)} \\ \Leftrightarrow & \text{For all assignments } g\text{, for all } a, b \in D_i \\ & \|X \vee Y = Y \vee X\|^{H^M, g[a/X_i][b/Y_i]} = T \\ \Leftrightarrow & \text{For all } g\text{, we have } \|\forall X \forall Y (X \vee Y = Y \vee X)\|^{H^{\mathcal{N}}, g} = T \\ \Leftrightarrow & H^{\mathcal{N}} \models^{\text{HOL}} \text{COM}\vee \end{array}$$

□

Lemma 5.8. *Let $H^{\mathcal{N}}$ be a Henkin model for a Boolean normative model $\mathcal{N} = \langle \mathcal{B}, V, N^V\rangle$. For all conditional norms (φ, ψ), arbitrary variable assignments g it holds: $V(\psi) \in out_i^{\mathcal{B}}(N, \{V(\varphi)\})$ if and only if $\|\lfloor d_i(N)(\varphi,\psi)\rfloor\|^{H^{\mathcal{N}}, g} = T$.*

Proof. Fact: We should notice that for all $\varphi \in \mathbf{Fm}(X)$ and for all assignments g by induction on the structure of φ we have $\|\lfloor\varphi\rfloor\|^{H^{\mathcal{N}}, g} = V(\varphi)$. For simplification we use the term abbreviations for saturated set, the ordering $\leq$ and upward set. It is easy to see that these terms abbreviations have the same corresponding sets in the corresponding Henkin model as the Boolean algebra.

$(d_1(N))$

$$
\begin{array}{ll}
 & \|\lfloor d_1(N)(\varphi,\psi)\rfloor\|^{H^{\mathcal{N}},g} = T \\
\Leftrightarrow & \|(\bigcirc_1(N)_{\tau\to\tau}\{\lfloor\varphi\rfloor\})\lfloor\psi\rfloor\|^{H^{\mathcal{N}},g} = T \\
\Leftrightarrow & \|(\lambda A_\tau \lambda X_i(\exists U\,(\exists Y\,(\exists Z\,(A\,Z \wedge Z \leq Y \wedge N\,Y\,U \wedge U \leq X))))\{\lfloor\varphi\rfloor\})\lfloor\psi\rfloor\|^{H^{\mathcal{N}},g} = T \\
\Leftrightarrow & \|(\lambda X_i(\exists U\,(\exists Y\,(\exists Z\,(\{\lfloor\varphi\rfloor\}\,Z \wedge Z \leq Y \wedge N\,Y\,U \wedge U \leq X)))))\lfloor\psi\rfloor\|^{H^{\mathcal{N}},g} = T \\
\Leftrightarrow & \|\exists U\,(\exists Y\,(\exists Z\,(\{\lfloor\varphi\rfloor\}\,Z \wedge Z \leq Y \wedge N\,Y\,U \wedge U \leq \lfloor\psi\rfloor)))\|^{H^{\mathcal{N}},g} = T \\
\Leftrightarrow & \|\exists U\,(\exists Y\,(\,\lfloor\varphi\rfloor \leq Y \wedge N\,Y\,U \wedge U \leq \lfloor\psi\rfloor))\|^{H^{\mathcal{N}},g} = T \\
\Leftrightarrow & \text{There are elements } b \text{ and } c \text{ such that } b, c \in D_i \text{ and} \\
 & \|\lfloor\varphi\rfloor \leq Y \wedge N\,Y\,U \wedge U \leq \lfloor\psi\rfloor\|^{H^M,g[b/U_i][c/Y_i]} = T \\
\Leftrightarrow & \text{There are elements } b, c \in B \text{ such that} \\
 & V(\varphi) \leq c \wedge N^V\,c\,b \wedge b \leq V(\psi) \\
\Leftrightarrow & V(\psi) \in Up(N^V(Up(\{V(\varphi)\}))) \\
\Leftrightarrow & V(\psi) \in out_1^{\mathcal{B}}(N^V, \{V(\varphi)\})
\end{array}
$$

$(d_2(N))$

$$
\begin{array}{ll}
 & \|\lfloor d_2(N)(\varphi,\psi)\rfloor\|^{H^{\mathcal{N}},g} = T \\
\Leftrightarrow & \|(\bigcirc_2(N)_{\tau\to\tau}\{\lfloor\varphi\rfloor\})\lfloor\psi\rfloor\|^{H^{\mathcal{N}},g} = T \\
\Leftrightarrow & \|(\lambda A_\tau \lambda X_i(\forall V\,(\mathit{Saturated}\,V \wedge \forall U(A\,U \to V\,U) \\
 & \to \exists Y\,(\exists Z\,(Z \leq X \wedge N\,Y\,Z \wedge V\,Y))))\{\lfloor\varphi\rfloor\})\lfloor\psi\rfloor\|^{H^{\mathcal{N}},g} = T \\
\Leftrightarrow & \|(\lambda X_i(\forall V\,(\mathit{Saturated}\,V \wedge \forall U(\{\lfloor\varphi\rfloor\}\,U \to V\,U) \\
 & \to \exists Y\,(\exists Z\,(Z \leq X \wedge N\,Y\,Z \wedge V\,Y)))))\lfloor\psi\rfloor\|^{H^{\mathcal{N}},g} = T \\
\Leftrightarrow & \|\forall V\,(\mathit{Saturated}\,V \wedge \forall U(\{\lfloor\varphi\rfloor\}\,U \to V\,U) \\
 & \to \exists Y\,(\exists Z\,(Z \leq \lfloor\psi\rfloor \wedge N\,Y\,Z \wedge V\,Y)))\|^{H^{\mathcal{N}},g} = T \\
\Leftrightarrow & \text{There are elements } b \text{ and } c \text{ such that } b, c \in D_i \text{ and} \\
 & \|\forall V\,(\mathit{Saturated}\,V \wedge \forall U(\{\lfloor\varphi\rfloor\}\,U \to V\,U) \\
 & \to (Z \leq \lfloor\psi\rfloor \wedge N\,Y\,Z \wedge V\,Y))\|^{H^{\mathcal{N}},g[b/Y_i][c/Z_i]} = T \\
\Leftrightarrow & \text{For every saturated set } V \text{ that } \{V(\varphi)\} \subseteq V \\
 & \text{there are elements } b, c \in B \text{ such that} \\
 & c \leq V(\psi) \wedge N^V\,b\,c \wedge V\,b \\
\Leftrightarrow & \text{For every saturated set } V \text{ such that } \{V(\varphi)\} \subseteq V \\
 & \text{we have } V(\psi) \in Up(N^V(V))) \\
\Leftrightarrow & V(\psi) \in out_2^{\mathcal{B}}(N^V, \{V(\varphi)\})
\end{array}
$$

$(d_3(N))$

$$
\begin{aligned}
& \|\lfloor d_3(N)(\varphi,\psi)\rfloor\|^{H^{\mathcal{N}},g} = T \\
\Leftrightarrow\quad & \|(\bigcirc_3(N)_{\tau\to\tau}\{\lfloor\varphi\rfloor\})\lfloor\psi\rfloor\|^{H^{\mathcal{N}},g} = T \\
\Leftrightarrow\quad & \|(\lambda A_\tau \lambda X_i(\forall V(\forall U(A\,U \to V\,U) \wedge V = Up\,V \\
& \wedge\forall W(\exists Y(V\,Y \wedge NYW) \to VW) \\
& \to \exists Y\,(\exists Z\,(Z \leq X \wedge N\,Y\,Z \wedge V\,Y)))))\{\lfloor\varphi\rfloor\})\lfloor\psi\rfloor\|^{H^{\mathcal{N}},g} = T \\
\Leftrightarrow\quad & \|(\lambda X_i(\forall V(\forall U(\{\lfloor\varphi\rfloor\}\,U \to V\,U) \wedge V = Up\,V \\
& \wedge\forall W(\exists Y(V\,Y \wedge NYW) \to VW) \\
& \to \exists Y\,(\exists Z\,(Z \leq X \wedge N\,Y\,Z \wedge V\,Y)))))\lfloor\psi\rfloor\|^{H^{\mathcal{N}},g} = T \\
\Leftrightarrow\quad & \|\forall V(\forall U(\{\lfloor\varphi\rfloor\}\,U \to V\,U) \wedge V = Up\,V \\
& \wedge\forall W(\exists Y(V\,Y \wedge NYW) \to VW) \\
& \to \exists Y\,(\exists Z\,(Z \leq \lfloor\psi\rfloor \wedge N\,Y\,Z \wedge V\,Y)))\|^{H^{\mathcal{N}},g} = T \\
\Leftrightarrow\quad & \text{There are elements } b \text{ and } c \text{ such that } b, c \in D_i \text{ and} \\
& \|\forall V(\forall U(\{\lfloor\varphi\rfloor\}\,U \to V\,U) \wedge V = Up\,V \\
& \wedge\forall W(\exists Y(V\,Y \wedge NYW) \to VW) \\
& \to (Z \leq \lfloor\psi\rfloor \wedge N\,Y\,Z \wedge V\,Y))\|^{H^{\mathcal{N}},g[b/Y_i][c/Z_i]} = T \\
\Leftrightarrow\quad & \text{For every set } V \text{ that } Up(V) = V,\ \{V(\varphi)\} \subseteq V \text{ and } N^V(V) \subseteq V \\
& \text{there are elements } b, c \in B \text{ such that} \\
& c \leq V(\psi) \wedge N^V\,b\,c \wedge V\,b \\
\Leftrightarrow\quad & \text{For every set } V \text{ that } Up(V) = V,\ \{V(\varphi)\} \subseteq V \text{ and } N^V(V) \subseteq V \\
& \text{we have } V(\psi) \in Up(N^V(V))) \\
\Leftrightarrow\quad & V(\psi) \in out_3^{\mathcal{B}}(N, \{V(\varphi)\})
\end{aligned}
$$

□

Lemma 5.9. *For every Henkin model* $H = \langle\{D_\alpha\}_{\alpha\in T}, I\rangle$ *such that* $H \models^{HOL} \Sigma$ *for all* $\Sigma \in \{COM\vee, COM\wedge, ASS\vee, ASS\wedge, IDE\vee, IDE\wedge, COMP\vee, COMP\wedge, Dis\vee\wedge,\ Dis\wedge\vee\}$*, there exists a corresponding Boolean normative model* $\mathcal{N}$*. Corresponding means that for all conditional norms* (φ,ψ) *and for all assignment* g*,* $\|\lfloor d_i(N)(\varphi,\psi)\rfloor\|^{H,g} = T$ *if and only if* $V(\psi) \in out_i^{\mathcal{B}}(N^V, \{V(\varphi)\})$*.*

Proof. Suppose that $H = \langle\{D_\alpha\}_{\alpha\in T}, I\rangle$ is a Henkin model such that $H \models^{\mathrm{HOL}} \Sigma$ for all $\Sigma \in \{COM\vee, ..., Dis\wedge\vee\}$. Without loss of generality, we can assume that the domains of H are denumerable [110]. We construct the corresponding Boolean normative model $\mathcal{N}$ as follows:

- $B = D_i$.
- $1 = I\top_i$.
- $0 = I\bot_i$.
- $a \vee b = c$ for $a, b, c \in B$ iff $I \vee_{i \to i \to i} ab = c$.
- $a \wedge b = c$ for $a, b, c \in B$ iff $I \wedge_{i \to i \to i} ab = c$.
- $a = \neg b$ for $a, b \in B$ iff $I \neg_{i \to i} a = b$.
- The valuation on $\mathcal{B}$ is defined such that for all $p^j \in X$, $V(p^j) = I(p^j_i)$.
- $(a, b) \in N^V$ for $a, b \in B$ iff $IN_{i \to \tau}(a, b) = T$.

Since $H \models^{\text{HOL}} \Sigma$ for all $\Sigma \in \{COM\vee, ..., Dis \wedge \vee\}$, it is straightforward (but tedious) to verify that $\wedge$, $\vee$, $\neg$, 0 and 1 satisfy the conditions as required for a Boolean algebra .

Moreover, the above construction ensures that H is a Henkin model $H^{\mathcal{N}}$ for Boolean normative system $\mathcal{N}$. Hence, Lemma 5.8 applies. This ensures that for all conditional norms (φ, ψ), for all assignment g we have $\|\lfloor d_i(N)(\varphi, \psi)\rfloor\|^{H,g} = T$ if and only if $V(\psi) \in out_i^{\mathcal{B}}(N^V, \{V(\varphi)\})$. □

Theorem 5.10 (Soundness and Completeness of the Embedding)**.**

$$\textit{For every Boolean normative model } \mathcal{N} \quad V(\psi) \in out_i^{\mathcal{B}}(N^V, \{V(\varphi)\})$$

if and only if

$$\{COM\vee, ..., Dis \wedge \vee\} \models^{HOL} \lfloor d_i(N)(\varphi, \psi)\rfloor$$

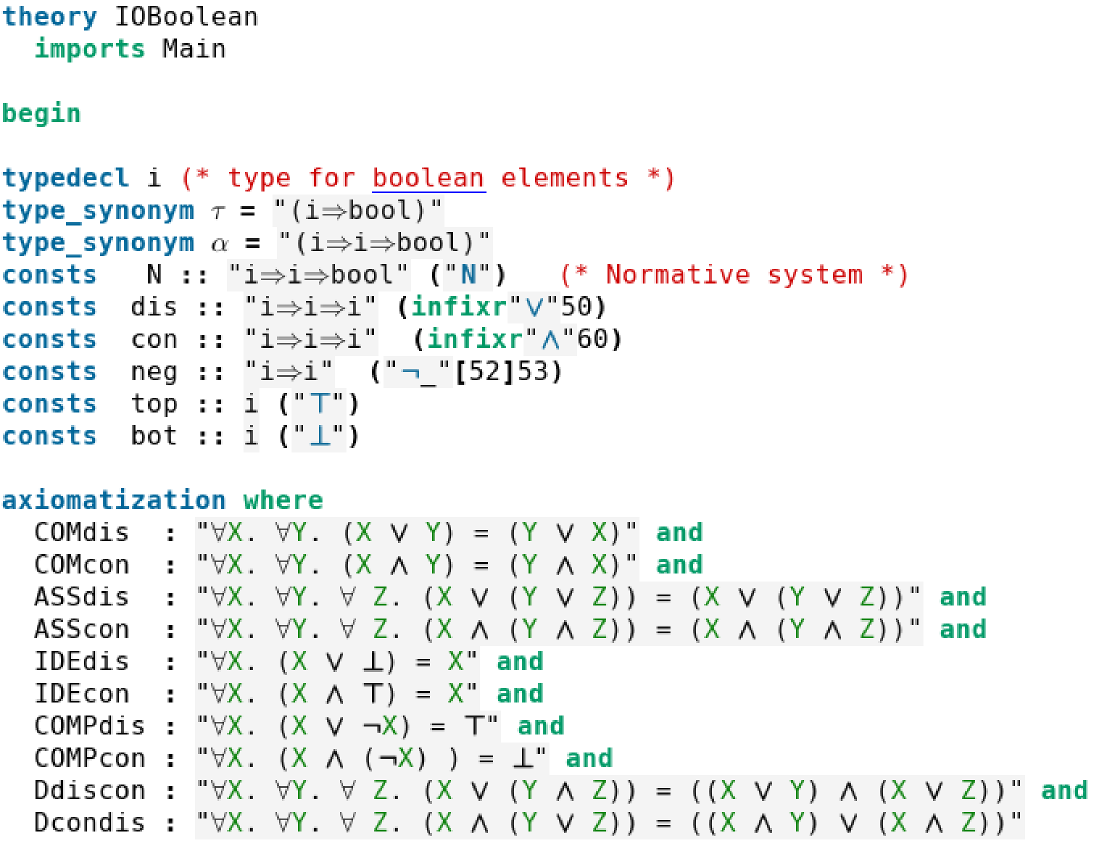
Proof. (Soundness, $\leftarrow$) The proof is by contraposition. Suppose for a Boolean normative model $\langle \mathcal{B}, N^V \rangle$ we have $V(\psi) \notin out_i^{\mathcal{B}}(N^V, \{V(\psi)\})$ Now let $H^{\mathcal{N}}$ be a Henkin model for Boolean normative model $\mathcal{N}$. Then by Lemma 5.8 for an arbitrary assignment g, it holds that $\|\lfloor d_i(N)(\varphi, \psi)\rfloor\|^{H^{\mathcal{N}},g} = F$, but $\|COM \vee \|^{H^{\mathcal{N}},g} = T$ ... $\|Dis \wedge \vee\|^{H^{\mathcal{N}},g} = T$ that is contradiction.

(Completeness, $\rightarrow$) The proof is again by contraposition. Assume $\{COM\vee, ..., Dis \wedge \vee\} \not\models^{\text{HOL}} \lfloor d_i(N)(\varphi, \psi)\rfloor$ then there is a Henkin model $H = \langle \{D_\alpha\}_{\alpha \in T}, I\rangle$ such that $H \models^{\text{HOL}} \Sigma$ for all $\Sigma \in \{COM\vee, ..., Dis \wedge \vee\}$, but $\|\lfloor d_i(N)(\varphi, \psi)\rfloor\|^{H,g} = F$ for some assignment g. By Lemma 5.9, there is a Boolean normative model $\mathcal{N}$ such that $V(\psi) \notin out_i^{\mathcal{B}}(N^V, \{V(\varphi)\})$, that is contradiction.

□

5.5 Implementation in Isabelle/HOL

The semantical embedding as devised in last section has been implemented in the higher-order proof assistant Isabelle/HOL [156]. Figures 5.8 and 5.9 display the respective encoding. In Figure 5.8, after introducing type i for representing the elements of Boolean algebra, we introduced the algebraic operators as constants in higher-order logic. Also the algebraic operators are characterized according to the definition of Boolean algebra.

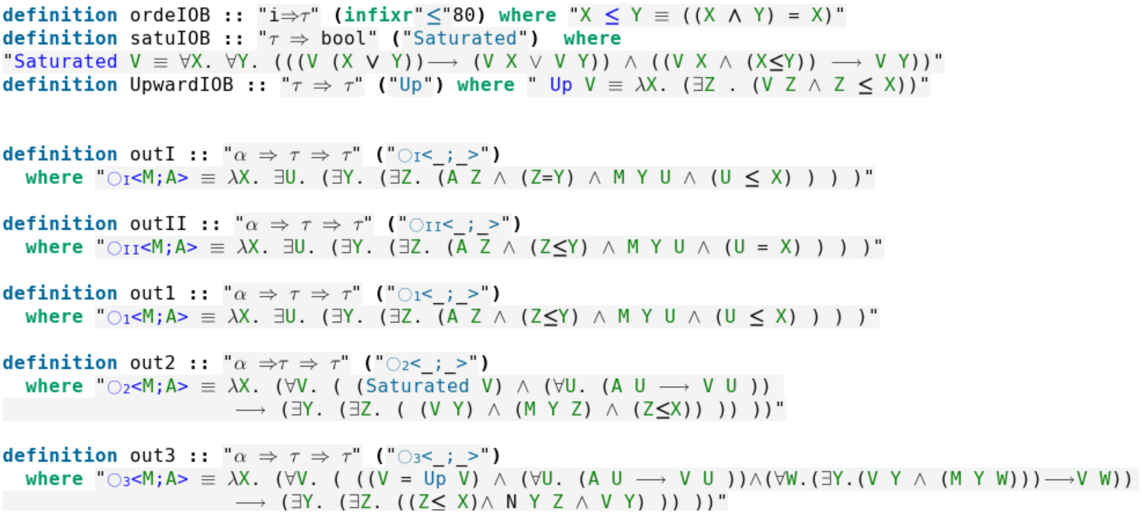


```
theory IOBoolean
  imports Main

begin

typedecl i (* type for boolean elements *)
type_synonym τ = "(i⇒bool)"
type_synonym α = "(i⇒i⇒bool)"
consts   N :: "i⇒i⇒bool" ("N")   (* Normative system *)
consts  dis :: "i⇒i⇒i" (infixr"∨"50)
consts  con :: "i⇒i⇒i"  (infixr"∧"60)
consts  neg :: "i⇒i"  ("¬_"[52]53)
consts  top :: i ("⊤")
consts  bot :: i ("⊥")

axiomatization where
  COMdis   : "∀X. ∀Y. (X ∨ Y) = (Y ∨ X)" and
  COMcon   : "∀X. ∀Y. (X ∧ Y) = (Y ∧ X)" and
  ASSdis   : "∀X. ∀Y. ∀ Z. (X ∨ (Y ∨ Z)) = (X ∨ (Y ∨ Z))" and
  ASScon   : "∀X. ∀Y. ∀ Z. (X ∧ (Y ∧ Z)) = (X ∧ (Y ∧ Z))" and
  IDEdis   : "∀X. (X ∨ ⊥) = X" and
  IDEcon   : "∀X. (X ∧ ⊤) = X" and
  COMPdis : "∀X. (X ∨ ¬X) = ⊤" and
  COMPcon : "∀X. (X ∧ (¬X) ) = ⊥" and
  Ddiscon : "∀X. ∀Y. ∀ Z. (X ∨ (Y ∧ Z)) = ((X ∨ Y) ∧ (X ∨ Z))" and
  Dcondis : "∀X. ∀Y. ∀ Z. (X ∧ (Y ∨ Z)) = ((X ∧ Y) ∨ (X ∧ Z))"
```

FIGURE 5.8: Semantical embedding of Boolean algebra in Isabelle/HOL

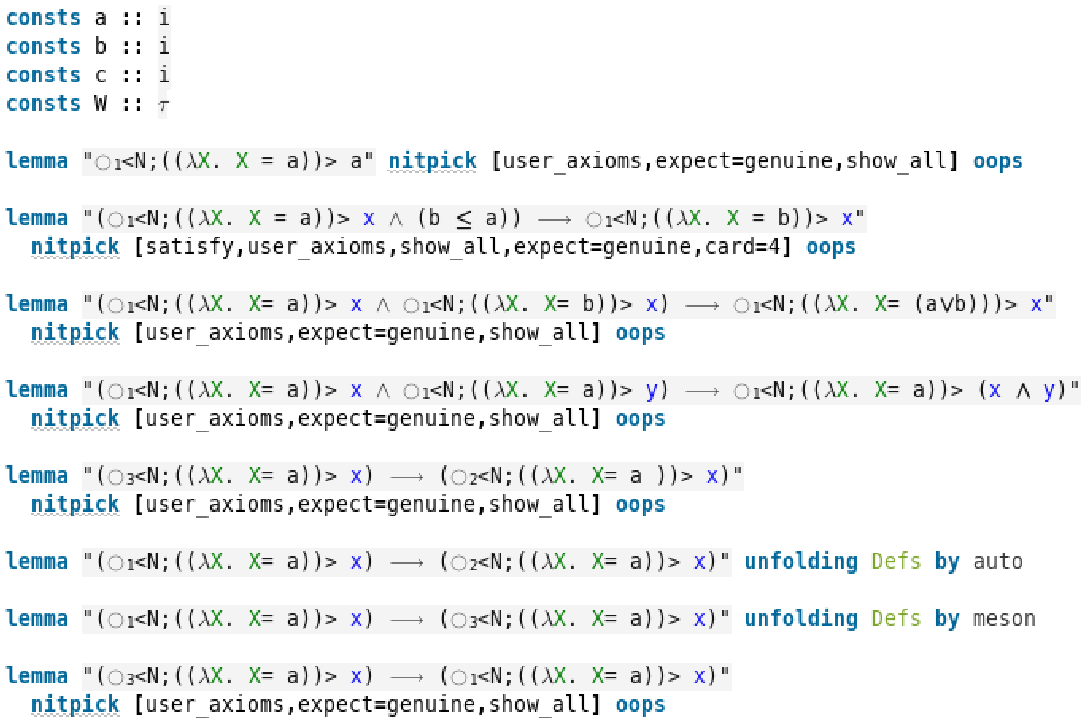
Figure 5.9 displays the semantical embedding of I/O operations in HOL, upward operator, and the property of saturated set.

Figure 5.10 shows some experiments, models and counter models finding, and some theorems about I/O operations.



```
definition ordeIOB :: "i⇒τ" (infixr"≤"80) where "X ≤ Y ≡ ((X ∧ Y) = X)"
definition satuIOB :: "τ ⇒ bool" ("Saturated")  where
"Saturated V ≡ ∀X. ∀Y. (((V (X ∨ Y))⟶ (V X ∨ V Y)) ∧ ((V X ∧ (X≤Y)) ⟶ V Y))"
definition UpwardIOB :: "τ ⇒ τ" ("Up") where " Up V ≡ λX. (∃Z . (V Z ∧ Z ≤ X))"


definition outI :: "α ⇒ τ ⇒ τ" ("○I<_;_>")
  where "○I<M;A> ≡ λX. ∃U. (∃Y. (∃Z. (A Z ∧ (Z=Y) ∧ M Y U ∧ (U ≤ X) ) ) )"

definition outII :: "α ⇒ τ ⇒ τ" ("○II<_;_>")
  where "○II<M;A> ≡ λX. ∃U. (∃Y. (∃Z. (A Z ∧ (Z≤Y) ∧ M Y U ∧ (U = X) ) ) )"

definition out1 :: "α ⇒ τ ⇒ τ" ("○1<_;_>")
  where "○1<M;A> ≡ λX. ∃U. (∃Y. (∃Z. (A Z ∧ (Z≤Y) ∧ M Y U ∧ (U ≤ X) ) ) )"

definition out2 :: "α ⇒τ ⇒ τ" ("○2<_;_>")
  where "○2<M;A> ≡ λX. (∀V. ( (Saturated V) ∧ (∀U. (A U ⟶ V U ))
                         ⟶ (∃Y. (∃Z. ( (V Y) ∧ (M Y Z) ∧ (Z≤X)) )) ))"

definition out3 :: "α ⇒ τ ⇒ τ" ("○3<_;_>")
  where "○3<M;A> ≡ λX. (∀V. ( ((V = Up V) ∧ (∀U. (A U ⟶ V U ))∧(∀W.(∃Y.(V Y ∧ (M Y W)))⟶V W))
                         ⟶ (∃Y. (∃Z. ((Z≤ X)∧ N Y Z ∧ V Y) )) ))"
```

FIGURE 5.9: Semantical embedding of out_i in Isabelle/HOL

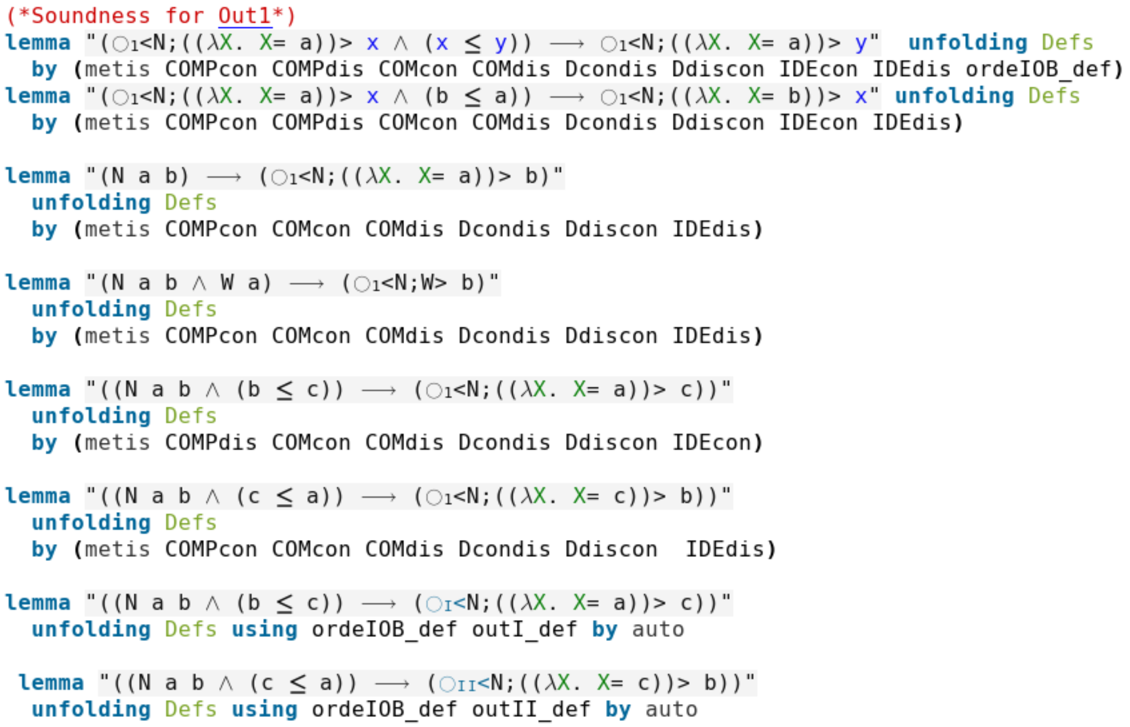


```
consts a :: i
consts b :: i
consts c :: i
consts W :: τ

lemma "○1<N;((λX. X = a))> a" nitpick [user_axioms,expect=genuine,show_all] oops

lemma "(○1<N;((λX. X = a))> x ∧ (b ≤ a)) ⟶ ○1<N;((λX. X = b))> x"
  nitpick [satisfy,user_axioms,show_all,expect=genuine,card=4] oops

lemma "(○1<N;((λX. X= a))> x ∧ ○1<N;((λX. X= b))> x) ⟶ ○1<N;((λX. X= (a∨b)))> x"
  nitpick [user_axioms,expect=genuine,show_all] oops

lemma "(○1<N;((λX. X= a))> x ∧ ○1<N;((λX. X= a))> y) ⟶ ○1<N;((λX. X= a))> (x ∧ y)"
  nitpick [user_axioms,expect=genuine,show_all] oops

lemma "(○3<N;((λX. X= a))> x) ⟶ (○2<N;((λX. X= a ))> x)"
  nitpick [user_axioms,expect=genuine,show_all] oops

lemma "(○1<N;((λX. X= a))> x) ⟶ (○2<N;((λX. X= a))> x)" unfolding Defs by auto

lemma "(○1<N;((λX. X= a))> x) ⟶ (○3<N;((λX. X= a))> x)" unfolding Defs by meson

lemma "(○3<N;((λX. X= a))> x) ⟶ (○1<N;((λX. X= a))> x)"
  nitpick [user_axioms,expect=genuine,show_all] oops
```

FIGURE 5.10: Some experiments about out_i in Isabelle/HOL

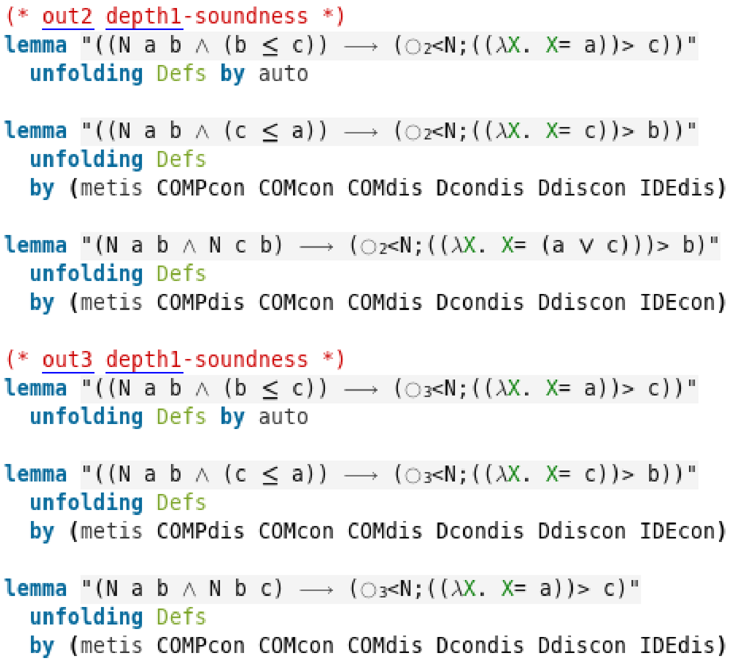
The first two lemmas prove the soundness of out_1, cf. Fig. 5.11. The next two lemmas show the factual detachment of this output operation. The last two lemmas illustrate the soundness for the out_I and out_{II} where the depth of inference is one.

Moreover, Figure 5.12 shows the soundness of out_2 and out_3 for the depth one.



```
(*Soundness for Out1*)
lemma "(○₁<N;((λX. X= a))> x ∧ (x ≤ y)) ⟶ ○₁<N;((λX. X= a))> y"  unfolding Defs
  by (metis COMPcon COMPdis COMcon COMdis Dcondis Ddiscon IDEcon IDEdis ordeIOB_def)
lemma "(○₁<N;((λX. X= a))> x ∧ (b ≤ a)) ⟶ ○₁<N;((λX. X= b))> x" unfolding Defs
  by (metis COMPcon COMPdis COMcon COMdis Dcondis Ddiscon IDEcon IDEdis)

lemma "(N a b) ⟶ (○₁<N;((λX. X= a))> b)"
  unfolding Defs
  by (metis COMPcon COMcon COMdis Dcondis Ddiscon IDEdis)

lemma "(N a b ∧ W a) ⟶ (○₁<N;W> b)"
  unfolding Defs
  by (metis COMPcon COMcon COMdis Dcondis Ddiscon IDEdis)

lemma "((N a b ∧ (b ≤ c)) ⟶ (○₁<N;((λX. X= a))> c))"
  unfolding Defs
  by (metis COMPdis COMcon COMdis Dcondis Ddiscon IDEcon)

lemma "((N a b ∧ (c ≤ a)) ⟶ (○₁<N;((λX. X= c))> b))"
  unfolding Defs
  by (metis COMPcon COMcon COMdis Dcondis Ddiscon  IDEdis)

lemma "((N a b ∧ (b ≤ c)) ⟶ (○I<N;((λX. X= a))> c))"
  unfolding Defs using ordeIOB_def outI_def by auto

 lemma "((N a b ∧ (c ≤ a)) ⟶ (○II<N;((λX. X= c))> b))"
  unfolding Defs using ordeIOB_def outII_def by auto
```

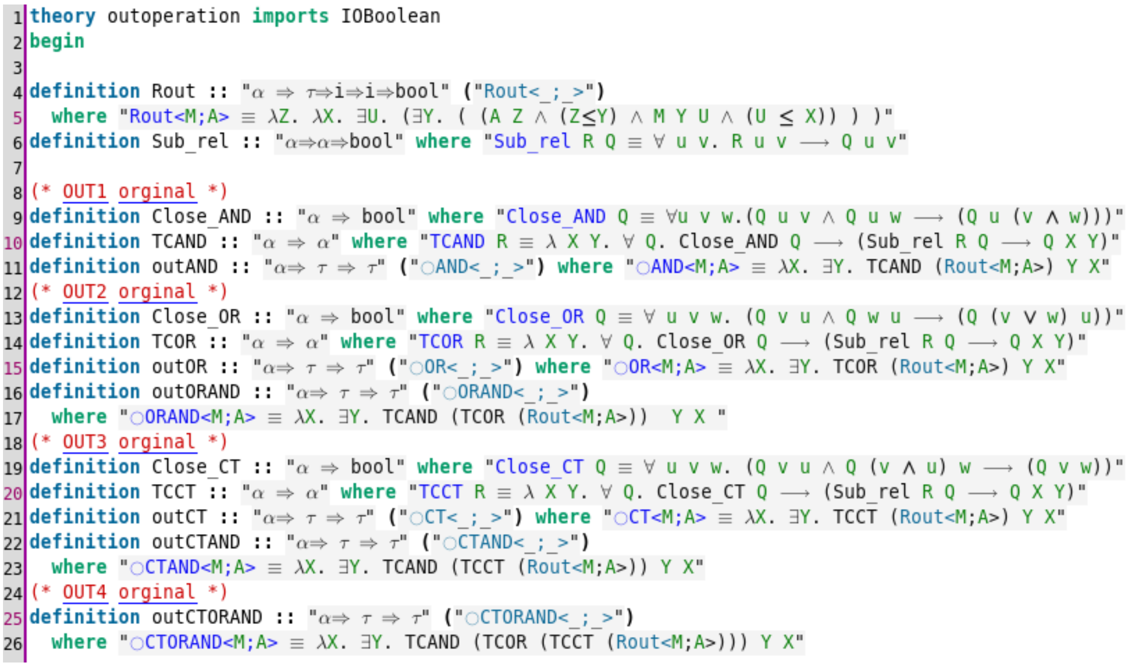
FIGURE 5.11: Soundness of out_1 in Isabelle/HOL



```
(* out2 depth1-soundness *)
lemma "((N a b ∧ (b ≤ c)) ⟶ (○₂<N;((λX. X= a))> c))"
  unfolding Defs by auto

lemma "((N a b ∧ (c ≤ a)) ⟶ (○₂<N;((λX. X= c))> b))"
  unfolding Defs
  by (metis COMPcon COMcon COMdis Dcondis Ddiscon IDEdis)

lemma "(N a b ∧ N c b) ⟶ (○₂<N;((λX. X= (a ∨ c)))> b)"
  unfolding Defs
  by (metis COMPdis COMcon COMdis Dcondis Ddiscon IDEcon)

(* out3 depth1-soundness *)
lemma "((N a b ∧ (b ≤ c)) ⟶ (○₃<N;((λX. X= a))> c))"
  unfolding Defs by auto

lemma "((N a b ∧ (c ≤ a)) ⟶ (○₃<N;((λX. X= c))> b))"
  unfolding Defs
  by (metis COMPdis COMcon COMdis Dcondis Ddiscon IDEcon)

lemma "(N a b ∧ N b c) ⟶ (○₃<N;((λX. X= a))> c)"
  unfolding Defs
  by (metis COMPcon COMcon COMdis Dcondis Ddiscon IDEdis)
```

FIGURE 5.12: Soundness of out_2 and out_3 in Isabelle/HOL

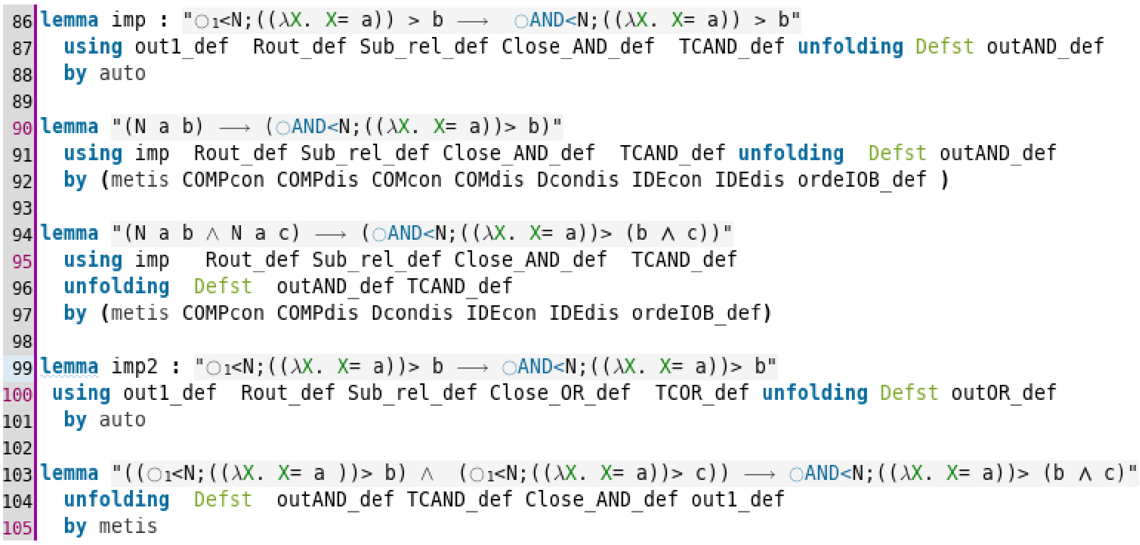
The output operations are implemented in Figure 5.13. The implementations are based on *reversibility of rules in the derivation systems.* We built the four output operations, introduced by Makinson and van der Torre [140], over the *simple-minded output operation*, see Section 3.3.



```
theory outoperation imports IOBoolean
begin

definition Rout :: "α ⇒ τ⇒i⇒i⇒bool" ("Rout<_;_>")
  where "Rout<M;A> ≡ λZ. λX. ∃U. (∃Y. ( (A Z ∧ (Z≤Y) ∧ M Y U ∧ (U ≤ X)) ) )"
definition Sub_rel :: "α⇒α⇒bool" where "Sub_rel R Q ≡ ∀ u v. R u v ⟶ Q u v"

(* OUT1 orginal *)
definition Close_AND :: "α ⇒ bool" where "Close_AND Q ≡ ∀u v w.(Q u v ∧ Q u w ⟶ (Q u (v ∧ w)))"
definition TCAND :: "α ⇒ α" where "TCAND R ≡ λ X Y. ∀ Q. Close_AND Q ⟶ (Sub_rel R Q ⟶ Q X Y)"
definition outAND :: "α⇒ τ ⇒ τ" ("○AND<_;_>") where "○AND<M;A> ≡ λX. ∃Y. TCAND (Rout<M;A>) Y X"
(* OUT2 orginal *)
definition Close_OR :: "α ⇒ bool" where "Close_OR Q ≡ ∀ u v w. (Q v u ∧ Q w u ⟶ (Q (v ∨ w) u))"
definition TCOR :: "α ⇒ α" where "TCOR R ≡ λ X Y. ∀ Q. Close_OR Q ⟶ (Sub_rel R Q ⟶ Q X Y)"
definition outOR :: "α⇒ τ ⇒ τ" ("○OR<_;_>") where "○OR<M;A> ≡ λX. ∃Y. TCOR (Rout<M;A>) Y X"
definition outORAND :: "α⇒ τ ⇒ τ" ("○ORAND<_;_>")
  where "○ORAND<M;A> ≡ λX. ∃Y. TCAND (TCOR (Rout<M;A>))  Y X "
(* OUT3 orginal *)
definition Close_CT :: "α ⇒ bool" where "Close_CT Q ≡ ∀ u v w. (Q v u ∧ Q (v ∧ u) w ⟶ (Q v w))"
definition TCCT :: "α ⇒ α" where "TCCT R ≡ λ X Y. ∀ Q. Close_CT Q ⟶ (Sub_rel R Q ⟶ Q X Y)"
definition outCT :: "α⇒ τ ⇒ τ" ("○CT<_;_>") where "○CT<M;A> ≡ λX. ∃Y. TCCT (Rout<M;A>) Y X"
definition outCTAND :: "α⇒ τ ⇒ τ" ("○CTAND<_;_>")
  where "○CTAND<M;A> ≡ λX. ∃Y. TCAND (TCCT (Rout<M;A>)) Y X"
(* OUT4 orginal *)
definition outCTORAND :: "α⇒ τ ⇒ τ" ("○CTORAND<_;_>")
  where "○CTORAND<M;A> ≡ λX. ∃Y. TCAND (TCOR (TCCT (Rout<M;A>))) Y X"
```

Figure 5.13: Semantical embedding of initial output operations in Isabelle/HOL

Following lemmas (cf. Fig. 5.14) shows the automation capability of implemented output operations, case of out_1.

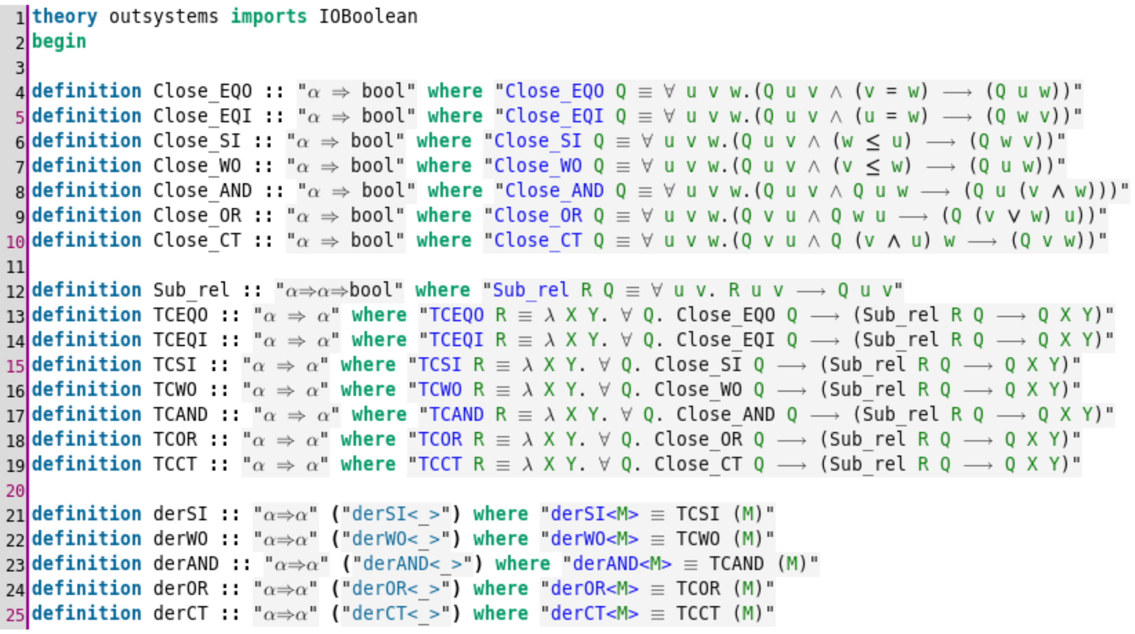


```
lemma imp : "○₁<N;((λX. X= a)) > b ⟶  ○AND<N;((λX. X= a)) > b"
  using out1_def  Rout_def Sub_rel_def Close_AND_def  TCAND_def unfolding Defst outAND_def
  by auto

lemma "(N a b) ⟶ (○AND<N;((λX. X= a))> b)"
  using imp  Rout_def Sub_rel_def Close_AND_def  TCAND_def unfolding  Defst outAND_def
  by (metis COMPcon COMPdis COMcon COMdis Dcondis IDEcon IDEdis ordeIOB_def )

lemma "(N a b ∧ N a c) ⟶ (○AND<N;((λX. X= a))> (b ∧ c))"
  using imp   Rout_def Sub_rel_def Close_AND_def  TCAND_def
  unfolding  Defst  outAND_def TCAND_def
  by (metis COMPcon COMPdis Dcondis IDEcon IDEdis ordeIOB_def)

lemma imp2 : "○₁<N;((λX. X= a))> b ⟶ ○AND<N;((λX. X= a))> b"
 using out1_def  Rout_def Sub_rel_def Close_OR_def  TCOR_def unfolding Defst outOR_def
  by auto

lemma "((○₁<N;((λX. X= a ))> b) ∧  (○₁<N;((λX. X= a))> c)) ⟶ ○AND<N;((λX. X= a))> (b ∧ c)"
  unfolding  Defst  outAND_def TCAND_def Close_AND_def out1_def
  by metis
```

Figure 5.14: Semantical embedding of initial output operations in Isabelle/HOL

Finally, we can implement the proof system of input/output logic directly in Isabelle/HOL, cf. Fig. 5.15 and 5.16. The idea is based on (universal) order of rules in a derivation.

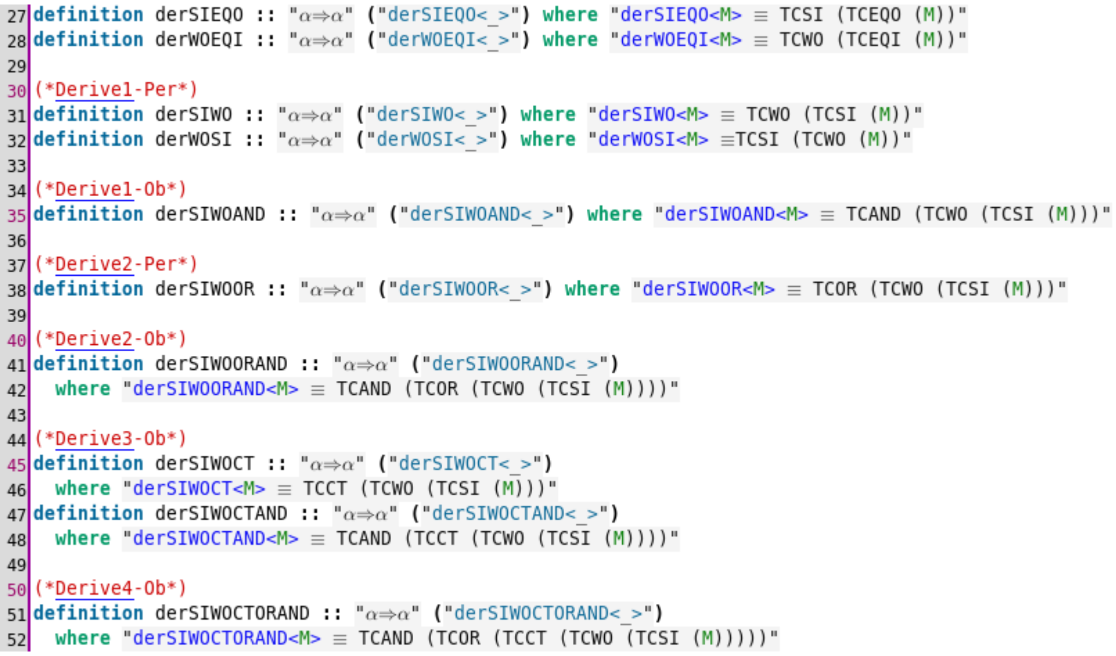
Ordering of rules and closure operation of syntactical properties of proof systems are the main trick for defining the derivation systems. For more details see Section 3.3.



```
theory outsystems imports IOBoolean
begin

definition Close_EQO :: "α ⇒ bool" where "Close_EQO Q ≡ ∀ u v w.(Q u v ∧ (v = w) ⟶ (Q u w))"
definition Close_EQI :: "α ⇒ bool" where "Close_EQI Q ≡ ∀ u v w.(Q u v ∧ (u = w) ⟶ (Q w v))"
definition Close_SI :: "α ⇒ bool" where "Close_SI Q ≡ ∀ u v w.(Q u v ∧ (w ≤ u) ⟶ (Q w v))"
definition Close_WO :: "α ⇒ bool" where "Close_WO Q ≡ ∀ u v w.(Q u v ∧ (v ≤ w) ⟶ (Q u w))"
definition Close_AND :: "α ⇒ bool" where "Close_AND Q ≡ ∀ u v w.(Q u v ∧ Q u w ⟶ (Q u (v ∧ w)))"
definition Close_OR :: "α ⇒ bool" where "Close_OR Q ≡ ∀ u v w.(Q v u ∧ Q w u ⟶ (Q (v ∨ w) u))"
definition Close_CT :: "α ⇒ bool" where "Close_CT Q ≡ ∀ u v w.(Q v u ∧ Q (v ∧ u) w ⟶ (Q v w))"

definition Sub_rel :: "α⇒α⇒bool" where "Sub_rel R Q ≡ ∀ u v. R u v ⟶ Q u v"
definition TCEQO :: "α ⇒ α" where "TCEQO R ≡ λ X Y. ∀ Q. Close_EQO Q ⟶ (Sub_rel R Q ⟶ Q X Y)"
definition TCEQI :: "α ⇒ α" where "TCEQI R ≡ λ X Y. ∀ Q. Close_EQI Q ⟶ (Sub_rel R Q ⟶ Q X Y)"
definition TCSI :: "α ⇒ α" where "TCSI R ≡ λ X Y. ∀ Q. Close_SI Q ⟶ (Sub_rel R Q ⟶ Q X Y)"
definition TCWO :: "α ⇒ α" where "TCWO R ≡ λ X Y. ∀ Q. Close_WO Q ⟶ (Sub_rel R Q ⟶ Q X Y)"
definition TCAND :: "α ⇒ α" where "TCAND R ≡ λ X Y. ∀ Q. Close_AND Q ⟶ (Sub_rel R Q ⟶ Q X Y)"
definition TCOR :: "α ⇒ α" where "TCOR R ≡ λ X Y. ∀ Q. Close_OR Q ⟶ (Sub_rel R Q ⟶ Q X Y)"
definition TCCT :: "α ⇒ α" where "TCCT R ≡ λ X Y. ∀ Q. Close_CT Q ⟶ (Sub_rel R Q ⟶ Q X Y)"

definition derSI :: "α⇒α" ("derSI< _>") where "derSI<M> ≡ TCSI (M)"
definition derWO :: "α⇒α" ("derWO< _>") where "derWO<M> ≡ TCWO (M)"
definition derAND :: "α⇒α" ("derAND< _>") where "derAND<M> ≡ TCAND (M)"
definition derOR :: "α⇒α" ("derOR< _>") where "derOR<M> ≡ TCOR (M)"
definition derCT :: "α⇒α" ("derCT< _>") where "derCT<M> ≡ TCCT (M)"
```

FIGURE 5.15: Semantical embedding of I/O proof systems in Isabelle/HOL

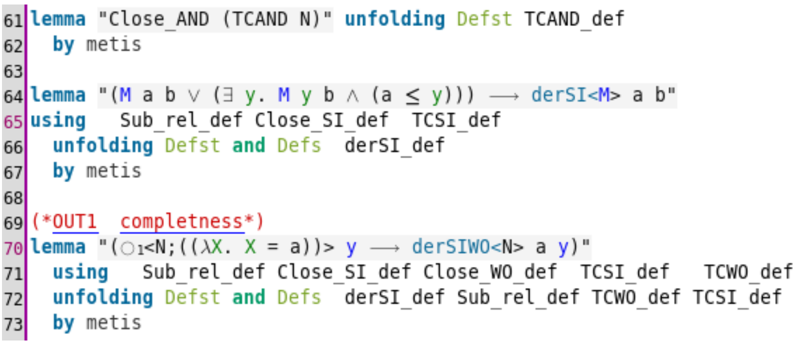
For example in the line 27 in Fig. 5.16, `derSIEQO` introduce the derivation system $derive_I$ with rules of $\{SI, EQO\}$ and lines 51–52 `derSIWOCTORAND` the derivation system $derive_4$, with rules of $\{SI, WO, CT, OR, AND\}$



```
27 definition derSIEQO :: "α⇒α" ("derSIEQO< _>") where "derSIEQO<M> ≡ TCSI (TCEQO (M))"
28 definition derWOEQI :: "α⇒α" ("derWOEQI< _>") where "derWOEQI<M> ≡ TCWO (TCEQI (M))"
29
30 (*Derive1-Per*)
31 definition derSIWO :: "α⇒α" ("derSIWO< _>") where "derSIWO<M> ≡ TCWO (TCSI (M))"
32 definition derWOSI :: "α⇒α" ("derWOSI< _>") where "derWOSI<M> ≡TCSI (TCWO (M))"
33
34 (*Derive1-Ob*)
35 definition derSIWOAND :: "α⇒α" ("derSIWOAND< _>") where "derSIWOAND<M> ≡ TCAND (TCWO (TCSI (M)))"
36
37 (*Derive2-Per*)
38 definition derSIWOOR :: "α⇒α" ("derSIWOOR< _>") where "derSIWOOR<M> ≡ TCOR (TCWO (TCSI (M)))"
39
40 (*Derive2-Ob*)
41 definition derSIWOORAND :: "α⇒α" ("derSIWOORAND< _>")
42   where "derSIWOORAND<M> ≡ TCAND (TCOR (TCWO (TCSI (M))))"
43
44 (*Derive3-Ob*)
45 definition derSIWOCT :: "α⇒α" ("derSIWOCT< _>")
46   where "derSIWOCT<M> ≡ TCCT (TCWO (TCSI (M)))"
47 definition derSIWOCTAND :: "α⇒α" ("derSIWOCTAND< _>")
48   where "derSIWOCTAND<M> ≡ TCAND (TCCT (TCWO (TCSI (M))))"
49
50 (*Derive4-Ob*)
51 definition derSIWOCTORAND :: "α⇒α" ("derSIWOCTORAND< _>")
52   where "derSIWOCTORAND<M> ≡ TCAND (TCOR (TCCT (TCWO (TCSI (M)))))"
```

FIGURE 5.16: Semantical embedding of I/O proof systems in Isabelle/HOL

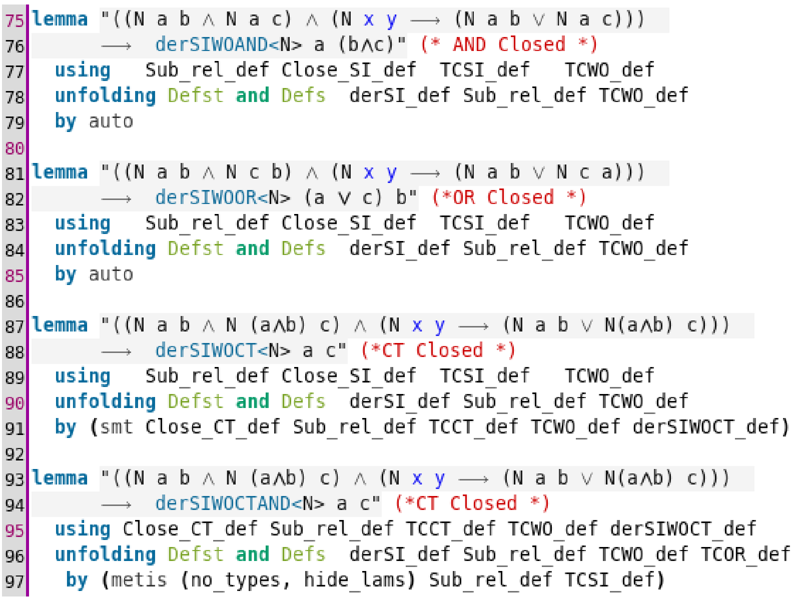
The advantage of implementing the proof system of I/O logic besides the output operations is possibility to check completeness theorems. For example, it is checked the completeness of *out*1 as figured in Fig. 5.17, lines 70–73. Lines 61 and 62 shows the AND closure is closed under AND. Lines 64–67 prove the capability of the implementation for a normative system M.



```
61 lemma "Close_AND (TCAND N)" unfolding Defst TCAND_def
62   by metis
63
64 lemma "(M a b ∨ (∃ y. M y b ∧ (a ≤ y))) ⟶ derSI<M> a b"
65 using   Sub_rel_def Close_SI_def  TCSI_def
66   unfolding Defst and Defs  derSI_def
67   by metis
68
69 (*OUT1  completness*)
70 lemma "(○₁<N;((λX. X = a))> y ⟶ derSIWO<N> a y)"
71   using   Sub_rel_def Close_SI_def Close_WO_def  TCSI_def   TCWO_def
72   unfolding Defst and Defs  derSI_def Sub_rel_def TCWO_def TCSI_def
73   by metis
```

Figure 5.17: Completeness checking of *out*1 in Isabelle/HOL

Moreover, we can examine the proof theoretical difference of I/O systems (cf. Fig. 5.18). For example lines 81–85 show that the implemented derivation system `derSIWOOR` (*derive*$_2$) is sound, for the rule of OR, for the depth one.



```
75 lemma "((N a b ∧ N a c) ∧ (N x y ⟶ (N a b ∨ N a c)))
76         ⟶  derSIWOAND<N> a (b∧c)" (* AND Closed *)
77   using   Sub_rel_def Close_SI_def  TCSI_def   TCWO_def
78   unfolding Defst and Defs  derSI_def Sub_rel_def TCWO_def
79   by auto
80
81 lemma "((N a b ∧ N c b) ∧ (N x y ⟶ (N a b ∨ N c a)))
82         ⟶  derSIWOOR<N> (a ∨ c) b" (*OR Closed *)
83   using   Sub_rel_def Close_SI_def  TCSI_def   TCWO_def
84   unfolding Defst and Defs  derSI_def Sub_rel_def TCWO_def
85   by auto
86
87 lemma "((N a b ∧ N (a∧b) c) ∧ (N x y ⟶ (N a b ∨ N(a∧b) c)))
88         ⟶  derSIWOCT<N> a c" (*CT Closed *)
89   using   Sub_rel_def Close_SI_def  TCSI_def   TCWO_def
90   unfolding Defst and Defs  derSI_def Sub_rel_def TCWO_def
91   by (smt Close_CT_def Sub_rel_def TCCT_def TCWO_def derSIWOCT_def)
92
93 lemma "((N a b ∧ N (a∧b) c) ∧ (N x y ⟶ (N a b ∨ N(a∧b) c)))
94         ⟶  derSIWOCTAND<N> a c" (*CT Closed *)
95   using Close_CT_def Sub_rel_def TCCT_def TCWO_def derSIWOCT_def
96   unfolding Defst and Defs  derSI_def Sub_rel_def TCWO_def TCOR_def
97    by (metis (no_types, hide_lams) Sub_rel_def TCSI_def)
```

Figure 5.18: Some experiments about I/O proof systems in Isabelle/HOL

5.6 Conclusion

We have presented an (indirect) embedding of two I/O operations in HOL, and we have shown that each embedding is faithful. The implementation already supports non-trivial applications in legal reasoning (GDPR) and moral theory (Moral luck). Moreover, we could similarly implement the intuitionistic I/O logic [162].

Also, a direct embedding of I/O operators in HOL is devised and shown that those embeddings are sound and complete, i.e., faithful. Moreover, based on the (universal) order of derivation, we have implemented the proof system of input/output logic in Isabelle/HOL. We have employed our implementation to systematically study some meta-logical properties of I/O logic, such as soundness and completeness for the simple-minded output operation within Isabelle/HOL.

.

Chapter 6

Deontic Modals, Preferences, and Computation

Preferences figure in our everyday language and actions. For example, economists invoke preferences to explain, predict, and assess economic outcomes. Preferences are subjective attitudes that determine how agents rank alternatives and can serve as reasons for actions.

A landmark and historically important family of dyadic deontic logics has been proposed by B. Hansson [109]. These logics have been recast in the framework of possible world semantics by Åqvist [12]. They come with a preference semantics, in which a binary preference relation ranks the possible words in terms of betterness. The framework was motivated by the well-known paradoxes of *contrary-to-duty* (CTD) reasoning like Chisholm [72]'s paradox. In this thesis, we focus on the class of all preference models, in which no specific properties (like reflexivity or transitivity) are required of the betterness relation. This class of models has a known axiomatic characterization, given by Åqvist's system **E** [161].

The chapter is structured as follows: Section 6.1 describes system **E**. The semantical embedding of **E** in HOL is then devised and studied in Section 6.2. This section also shows the faithfulness (viz. soundness and completeness) of the embedding. Section 6.3 discusses the implementation in Isabelle/HOL [156] and Section 6.4 compares system **E** and the Kratzerian framework. Section 6.5 concludes the chapter.

6.1 Dyadic Deontic Logic E

6.1.1 Syntax

The language of **E** is obtained by adding the following operators to the language of propositional logic: $\Box$ (for necessity); $\Diamond$ (for possibility); and $\bigcirc(\_/\_)$ (for conditional obligation); $P(\_/\_)$ (for conditional permission). $\bigcirc(\psi/\varphi)$ is read "If φ, then ψ is obligatory", and $P(\psi/\varphi)$ is read "If φ, then ψ is permitted'. The set of well-formed formulas (wffs) is defined in the straightforward way. Iteration of the modal and deontic operators is permitted, and so are "mixed" formulas, e.g., $\bigcirc(\psi/\varphi) \wedge \varphi$. We put $\top := \neg q \vee q$, for some atomic wff q, and $\bot := \neg\top$. $\Diamond$ is the dual of $\Box$, viz. $\Diamond\varphi := \neg\Box\neg\varphi$. P is also the dual of $\bigcirc$, viz. $P(\psi/\varphi) := \neg \bigcirc (\neg\psi/\varphi)$. System **E** is defined by the following axioms and rules:

Axiom schemata for propositional logic (PL)

S5-schemata for $\Box$ and $\Diamond$ (S5)

$$\bigcirc(\psi_1 \to \psi_2/\varphi) \to (\bigcirc(\psi_1/\varphi) \to \bigcirc(\psi_2/\varphi)) \tag{COK}$$

$$\bigcirc(\psi/\varphi) \to \Box\bigcirc(\psi/\varphi) \tag{Abs}$$

$$\Box\psi \to \bigcirc(\psi/\varphi) \tag{Nec}$$

$$\Box(\varphi_1 \leftrightarrow \varphi_2) \to (\bigcirc(\psi/\varphi_1) \leftrightarrow \bigcirc(\psi/\varphi_2)) \tag{Ext}$$

$$\bigcirc(\varphi/\varphi) \tag{Id}$$

$$\bigcirc(\psi/\varphi_1 \wedge \varphi_2) \to \bigcirc(\varphi_2 \to \psi/\varphi_1) \tag{Sh}$$

If $\vdash \varphi$ and $\vdash \varphi \to \psi$ then $\vdash \psi$ (MP)

If $\vdash \varphi$ then $\vdash \Box\varphi$ (N)

The notions of theoremhood, deducibility and consistency are defined as usual.

6.1.2 Semantics

A preference model is a structure $M = \langle W, \succeq, V\rangle$ where:

- W is a non-empty set of possible worlds (W is called "universe");
- $\succeq\ \subseteq W \times W$ (intuitively, $\succeq$ is a betterness or comparative goodness relation; "$s \succeq t$" can be read as "world s is at least as good as world t");

- V is a function assigning to each atomic wff a set of worlds, i.e., $V(p) \subseteq W$ (intuitively, $V(p)$ is the set of worlds at which p is true).

No specific properties (like reflexivity or transitivity) are required of the betterness relation.

Given a preference model $M = \langle W, \succeq, V \rangle$ and a world $s \in W$, we define the satisfaction relation $M, s \vDash \varphi$ (read as "world s satisfies φ in model M") by induction on the structure of φ as described below. Intuitively, the evaluation rule for the dyadic obligation operator puts $\bigcirc(\psi/\varphi)$ true whenever all the best φ-worlds are ψ-worlds. Here, best is defined in terms of optimality rather than maximality [161]. A φ-world is optimal if it is at least as good as any other φ-world. We define $V^M(\varphi) = \{s \in W \mid M, s \models \varphi\}$ and $\text{opt}_{\succeq}(V^M(\varphi)) = \{s \in V^M(\varphi) \mid \forall t(t \vDash \varphi \to s \succeq t)\}$. Whenever the model M is obvious from context, we write $V(\varphi)$ instead of $V^M(\varphi)$.

$$M, s \models p \text{ if and only if } s \in V(p)$$

$$M, s \models \neg\varphi \text{ if and only if } M, s \not\models \varphi \text{ (that is, not } M, s \models \varphi)$$

$$M, s \models \varphi \vee \psi \text{ if and only if } M, s \models \varphi \text{ or } M, s \models \psi$$

$$M, s \models \Box\varphi \text{ if and only if } V(\varphi) = W$$

$$M, s \models \bigcirc(\psi/\varphi) \text{ if and only if } \text{opt}_{\succeq}(V(\varphi)) \subseteq V(\psi)$$

As usual, a formula φ is valid in a preference model $M = \langle W, \succeq, V \rangle$ (notation: $M \models \varphi$) if and only if, for all worlds $s \in W$, $M, s \models \varphi$. A formula φ is valid (notation: $\models \varphi$) if and only if it is valid in every preference model. The notions of semantic consequence and satisfiability in a model are defined as usual.

Theorem 6.1. *System* **E** *is sound and complete with respect to the class of all preference models. System* **E** *is also sound and complete with respect to the class of those in which* $\succeq$ *is reflexive, and with respect to the class of those in which* $\succeq$ *is total (for all* $s, t \in W$, $s \succeq t$ *or* $t \succeq s$*).*

Proof. See Parent [161]. □

We can put more properties on the preference relation $\succeq$ such as

- limitedness: if $V(\phi) \neq \emptyset$, then $\text{opt}_{\succeq}(V(\phi)) \neq \emptyset$.
- transitivity: if $s \succeq t$ and $t \succeq z$, then $s \succeq z$.

F is the proof system obtained by supplementing **E** with (D^*) and **G** is the proof system obtained by supplementing **F** with (Sp).

$$\Diamond\varphi \rightarrow (\bigcirc(\psi/\varphi) \rightarrow P(\psi/\varphi)) \tag{D*}$$

$$(P(\psi/\varphi) \wedge \bigcirc((\psi \rightarrow \chi)/\varphi) \rightarrow \bigcirc(\chi/(\varphi \wedge \psi)) \tag{Sp}$$

Under the $\text{opt}_{\succeq}$ evaluation rule, **F** (**G**) is strongly sound and complete with respect to the class of preference models in which $\succeq$ is limited (limited and transitive).

6.2 Embedding E into HOL

6.2.1 Semantical embedding

The formulas of **E** are identified in our semantical embedding with certain HOL terms (predicates) of type $i \rightarrow o$. They can be applied to terms of type i, which are assumed to denote possible worlds. That is, the HOL type i is now identified with a (non-empty) set of worlds. Type $i \rightarrow o$ is abbreviated as τ in the remainder. The HOL signature is assumed to contain the constant symbol $r_{i\rightarrow\tau}$. Moreover, for each atomic propositional symbol p^j of **E**, the HOL signature must contain the corresponding constant symbol p^j_τ. Without loss of generality, we assume that besides those symbols and the primitive logical connectives of HOL, no other constant symbols are given in the signature of HOL.

The mapping $\lfloor\cdot\rfloor$ translates a formula φ of **E** into a term $\lfloor\varphi\rfloor$ of HOL of type τ. The mapping is defined recursively:

$$\begin{array}{lcl}
\lfloor p^j \rfloor & = & p^j_\tau \\
\lfloor \neg\varphi \rfloor & = & \neg_{\tau\rightarrow\tau} \lfloor\varphi\rfloor \\
\lfloor \varphi \vee \psi \rfloor & = & \vee_{\tau\rightarrow\tau\rightarrow\tau} \lfloor\varphi\rfloor \lfloor\psi\rfloor \\
\lfloor \Box\varphi \rfloor & = & \Box_{\tau\rightarrow\tau} \lfloor\varphi\rfloor \\
\lfloor \bigcirc(\psi/\varphi) \rfloor & = & \bigcirc_{\tau\rightarrow\tau\rightarrow\tau} \lfloor\varphi\rfloor \lfloor\psi\rfloor
\end{array}$$

$\neg_{\tau\rightarrow\tau}$, $\vee_{\tau\rightarrow\tau\rightarrow\tau}$, $\Box_{\tau\rightarrow\tau}$ and $\bigcirc_{\tau\rightarrow\tau\rightarrow\tau}$ abbreviate the following terms of HOL:

$$\begin{array}{ll}
\neg_{\tau\rightarrow\tau} & = \lambda A_\tau \lambda X_i \neg(A\,X) \\
\vee_{\tau\rightarrow\tau\rightarrow\tau} & = \lambda A_\tau \lambda B_\tau \lambda X_i (A\,X \vee B\,X) \\
\Box_{\tau\rightarrow\tau} & = \lambda A_\tau \lambda X_i \forall Y_i (A\,Y) \\
\bigcirc_{\tau\rightarrow\tau\rightarrow\tau} & = \lambda A_\tau \lambda B_\tau \lambda X_i \forall W_i (\,(\lambda V_i (A\,V \wedge (\forall Y_i (A\,Y \rightarrow r_{i\rightarrow\tau} V\,Y))))\,W \rightarrow B\,W)^{1}
\end{array}$$

Analyzing the truth of formula φ, represented by the HOL term $\lfloor\varphi\rfloor$, in a particular world w, represented by the term w_i, corresponds to evaluating the application $(\lfloor\varphi\rfloor\, w_i)$. In line with previous work [40], we define $vld_{\tau\to o} = \lambda A_\tau \forall S_i (A\, S)$. With this definition, validity of a formula s in $\mathbf{E}$ corresponds to the validity of the formula $(vld\,\lfloor\varphi\rfloor)$ in HOL, and vice versa.

6.2.2 Soundness and completeness

To prove the soundness and completeness, that is, faithfulness, of the above embedding, a mapping from preference models into Henkin models is employed.

Definition 6.2 (Preference model ⇒ Henkin model)**.** Let $M = \langle W, \succeq, V\rangle$ be a preference model. Let $p^1, ..., p^m$ for $m \geq 1$ be atomic propositional symbols and $\lfloor p^j\rfloor = p^j_\tau$ for $j = 1, ..., m$. A Henkin model $H^M = \langle\{D_\alpha\}_{\alpha\in T}, I\rangle$ for M is defined as follows: D_i is chosen as the set of possible worlds W and all other sets $D_{\alpha\to\beta}$ are chosen as (not necessarily full) sets of functions from D_α to D_β. For all $D_{\alpha\to\beta}$ the rule that every term $t_{\alpha\to\beta}$ must have a denotation in $D_{\alpha\to\beta}$ must be obeyed, in particular, it is required that D_τ and $D_{i\to\tau}$ contain the elements Ip^j_τ and $Ir_{i\to\tau}$. Interpretation I is constructed as follows:

1. For $1 \leq i \leq m$, $Ip^j_\tau \in D_\tau$ is chosen such that $Ip^j_\tau(s) = T$ iff $s \in V(p^j)$ in M.
2. $Ir_{i\to\tau} \in D_{i\to\tau}$ is chosen such that $Ir_{i\to\tau}(s, u) = T$ iff $s \succeq u$ in M.

Since we assume that there are no other symbols (besides the r, the p^j and the primitive logical connectives) in the signature of HOL, I is a total function. Moreover, the above construction guarantees that H^M is a Henkin model: $\langle D, I\rangle$ is a frame, and the choice of I in combination with the Denotatpflicht ensures that for arbitrary assignments g, $\|.\|^{H^M,g}$ is a total evaluation function.

Lemma 6.3. *Let H^M be a Henkin model for a preference model M. For all formulas δ of* $\mathbf{E}$*, all assignments g and worlds s it holds:*

$$M, s \models \delta \textit{ if and only if } \|\lfloor\delta\rfloor\, S_i\|^{H^M, g[s/S_i]} = T$$

Proof. See Appendix A.1. □

[1] If $\text{opt}_{\succeq}(A)$ is taken as a abbreviation for $\lambda V_i(AV \wedge (\forall Y_i(AY \to r_{i\to\tau} V\, Y)))$, then this can be simplified to $\bigcirc_{\tau\to\tau\to\tau} = \lambda A_\tau \lambda B_\tau \lambda X_i(\text{opt}_{\succeq}(A) \subseteq B)$.

Lemma 6.4 (Henkin model ⇒ Preference model)**.** *For every Henkin model $H = \langle\{D_\alpha\}_{\alpha\in T}, I\rangle$ there exists a corresponding preference model M. Corresponding here means that for all formulas δ of* **E** *and for all assignments g and worlds s,*

$$\|\lfloor\delta\rfloor S_i\|^{H,g[s/S_i]} = T \textit{ if and only if } M, s \vDash \delta$$

Proof. Suppose that $H = \langle\{D_\alpha\}_{\alpha\in T}, I\rangle$ is a Henkin model. Without loss of generality, we can assume that the domains of H are denumerable [110]. We construct the corresponding preference model M as follows:

- $W = D_i$.
- $s \succeq u$ for $s, u \in W$ iff $Ir_{i\to\tau}(s,u) = T$.
- $s \in V(p^j_\tau)$ iff $Ip^j_\tau(s) = T$ for all p^j.

Moreover, the above construction ensures that H is a Henkin model for M. Hence, Lemma 6.3 applies. This ensures that for all formulas δ of **E**, for all assignments g and all worlds s we have $\|\lfloor\delta\rfloor S_i\|^{H,g[s/S_i]} = T$ if and only if $M, s \vDash \delta$. □

Theorem 6.5 (Soundness and completeness of the embedding)**.**

$$\models \varphi \textit{ if and only if } \models^{HOL} \mathit{vld}\,\lfloor\varphi\rfloor$$

Proof. (Soundness, ←) The proof is by contraposition. Assume $\not\models \varphi$, i.e, there is a preference model $M = \langle W, \succeq, V\rangle$, and a world $s \in W$, such that $M, s \not\models \varphi$. By Lemma 6.3 for an arbitrary assignment g it holds that $\|\lfloor\varphi\rfloor S_i\|^{H^M,g[s/S_i]} = F$ in Henkin model $H^M = \langle\{D_\alpha\}_{\alpha\in T}, I\rangle$. Thus, by definition of $\|.\|$, it holds that $\|\forall S_i(\lfloor\varphi\rfloor S_i)\|^{H^M,g} = \|\mathit{vld}\,\lfloor\varphi\rfloor\|^{H^M,g} = F$. Hence, $H^M \not\models^{HOL} \mathit{vld}\,\lfloor\varphi\rfloor$. By definition $\not\models^{HOL} \mathit{vld}\,\lfloor\varphi\rfloor$.

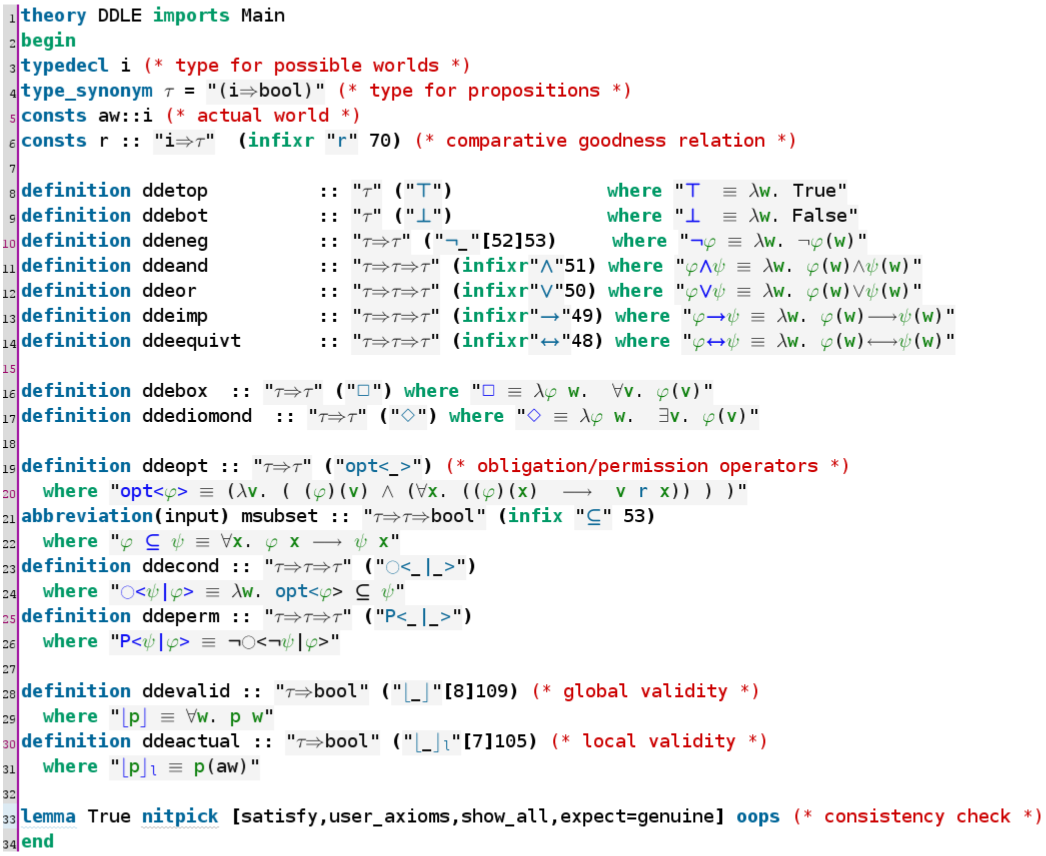
(Completeness, →) The proof is again by contraposition. Assume $\not\models^{HOL} \mathit{vld}\,\lfloor\varphi\rfloor$, i.e., there is a Henkin model $H = \langle\{D_\alpha\}_{\alpha\in T}, I\rangle$ and an assignment g such that $\|\mathit{vld}\,\lfloor\varphi\rfloor\|^{H,g} = F$. By Lemma 6.4, there is a preference model M such that $M \nvDash \varphi$. Hence, $\not\models \varphi$. □

Remark: In contrast to a deep logical embedding, in which the syntactical structure and the semantics of logic L would be formalized in full detail (using e.g., structural induction and recursion), only the core differences in the semantics of both system **E** and meta-logic HOL have been explicitly encoded in our shallow semantical embedding. In a certain sense we have thus shown, that system **E** can, in fact, be identified and handled as a natural fragment of HOL.

6.3 Implementation in Isabelle/HOL

6.3.1 Implementation

The semantical embedding as devised in Section 6.2 has been implemented in the higher-order proof assistant Isabelle/HOL [156]. Figure 6.1 displays the respective encoding.



```
1  theory DDLE imports Main
2  begin
3  typedecl i (* type for possible worlds *)
4  type_synonym τ = "(i⇒bool)" (* type for propositions *)
5  consts aw::i (* actual world *)
6  consts r :: "i⇒τ"  (infixr "r" 70) (* comparative goodness relation *)
7
8  definition ddetop          :: "τ" ("⊤")              where "⊤  ≡ λw. True"
9  definition ddebot          :: "τ" ("⊥")              where "⊥  ≡ λw. False"
10 definition ddeneg          :: "τ⇒τ" ("¬_"[52]53)     where "¬φ ≡ λw. ¬φ(w)"
11 definition ddeand          :: "τ⇒τ⇒τ" (infixr"∧"51) where "φ∧ψ ≡ λw. φ(w)∧ψ(w)"
12 definition ddeor           :: "τ⇒τ⇒τ" (infixr"∨"50) where "φ∨ψ ≡ λw. φ(w)∨ψ(w)"
13 definition ddeimp          :: "τ⇒τ⇒τ" (infixr"→"49) where "φ→ψ ≡ λw. φ(w)⟶ψ(w)"
14 definition ddeequivt       :: "τ⇒τ⇒τ" (infixr"↔"48) where "φ↔ψ ≡ λw. φ(w)⟷ψ(w)"
15
16 definition ddebox  :: "τ⇒τ" ("□") where "□ ≡ λφ w.  ∀v. φ(v)"
17 definition ddediomond  :: "τ⇒τ" ("◇") where "◇ ≡ λφ w.  ∃v. φ(v)"
18
19 definition ddeopt :: "τ⇒τ" ("opt<_>") (* obligation/permission operators *)
20   where "opt<φ> ≡ (λv. ( (φ)(v) ∧ (∀x. ((φ)(x)  ⟶  v r x)) ) )"
21 abbreviation(input) msubset :: "τ⇒τ⇒bool" (infix "⊆" 53)
22   where "φ ⊆ ψ ≡ ∀x. φ x ⟶ ψ x"
23 definition ddecond :: "τ⇒τ⇒τ" ("○<_|_>")
24   where "○<ψ|φ> ≡ λw. opt<φ> ⊆ ψ"
25 definition ddeperm :: "τ⇒τ⇒τ" ("P<_|_>")
26   where "P<ψ|φ> ≡ ¬○<¬ψ|φ>"
27
28 definition ddevalid :: "τ⇒bool" ("⌊_⌋"[8]109) (* global validity *)
29   where "⌊p⌋ ≡ ∀w. p w"
30 definition ddeactual :: "τ⇒bool" ("⌊_⌋ι"[7]105) (* local validity *)
31   where "⌊p⌋ι ≡ p(aw)"
32
33 lemma True nitpick [satisfy,user_axioms,show_all,expect=genuine] oops (* consistency check *)
34 end
```

FIGURE 6.1: Shallow semantical embedding of system **E** in Isabelle/HOL

- On line 3, the primitive type i for possible words is introduced.
- On line 4, the type τ for formulas is introduced.
- On line 5, a designated constant for the actual world (aw) is introduced.
- On line 6, the constant r is introduced. r is used to define the preference relation $\succeq$.
- Lines 8–14 define the Boolean logical connectives in the usual way.
- Lines 16 and 17 introduce the alethic operators $\Box$ and $\Diamond$.

- The dyadic deontic operators are defined in lines 19–26. Lines 19–20 introduce the notion of optimal φ-world, and lines 23–26 define the dyadic operators using this notion.
- Lines 28–31 introduce the notion of global validity (i.e, truth in all worlds) and local validity (truth at the actual world).

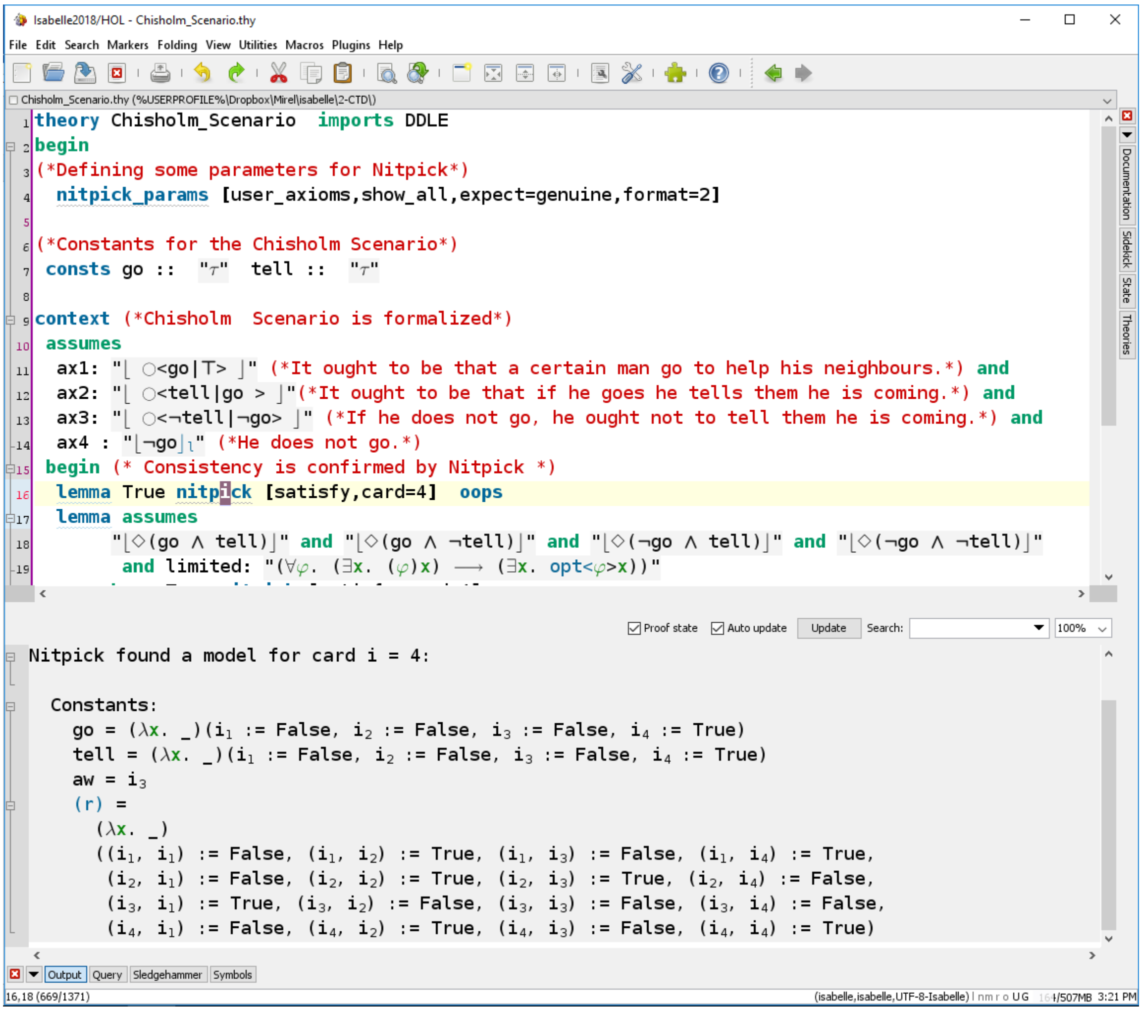
In the remainder of this chapter, we illustrate how the implementation in Isabelle/HOL can be used.

6.3.2 Contrary-to-duty scenarios

Chisholm's scenario We have discussed this paradox in Subsection 2.2.3. Figure 6.2 applies our implementation to Chisholm's scenario (cf. [72]).

1. It ought to be that a certain man goes (to the assistance of his neighbours).
2. It ought to be that if he goes he tells them he is coming.
3. If he does not go, he ought not to tell them he is coming.
4. He does not go.

These statements can be given a consistent formalisation in system **E**; cf. Figure 6.2. This is confirmed by the model finder Nitpick [44] integrated with Isabelle/HOL. Nitpick computes (cf. Figure 6.2, line 16) an intuitive, small model M_1 for the scenario consisting a set of possible worlds $\{i_1, i_2, i_3, i_4\}$ and the actual world i_1. The preference relation in this model is $\succeq = \{(i_1, i_2), (i_1, i_4), (i_2, i_2), (i_2, i_3), (i_3, i_1), (i_4, i_2), (i_4, i_4)\}$. In addition, the valuation function is $V(go) = V(tell) = \{i_4\}$. In the actual world the man doesn't go to help his neighbors and doesn't tell them that he is coming. Also, we have $\text{opt}_{\succeq}(V(\top)) = \emptyset$. So, $M_1, i_1 \models \bigcirc(go/\top)$ by the evaluation rule for $\bigcirc$. Similarly, $\text{opt}_{\succeq}(V(\neg go)) = \emptyset$ implies $M_1, i_1 \models \bigcirc(\neg tell/\neg go)$. Moreover, since $\text{opt}_{\succeq}(V(go)) = \{i_4\}$ and $V(tell) = \{i_4\}$ we have $M_1, i_1 \models \bigcirc(tell/go)$.


Isabelle2018/HOL - Chisholm_Scenario.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
Documentation
Sidekick
State
Theories
☑ Proof state
☑ Auto update
Update
Search:
100%
Output
Query
Sledgehammer
Symbols

□ Chisholm_Scenario.thy (%USERPROFILE%\Dropbox\Mirel\isabelle\2-CTD\)

```
theory Chisholm_Scenario  imports DDLE
begin
(*Defining some parameters for Nitpick*)
  nitpick_params [user_axioms,show_all,expect=genuine,format=2]

(*Constants for the Chisholm Scenario*)
 consts go ::  "τ"  tell ::  "τ"

context (*Chisholm  Scenario is formalized*)
 assumes
  ax1: "⌊ ○<go|⊤> ⌋" (*It ought to be that a certain man go to help his neighbours.*) and
  ax2: "⌊ ○<tell|go > ⌋"(*It ought to be that if he goes he tells them he is coming.*) and
  ax3: "⌊ ○<¬tell|¬go> ⌋" (*If he does not go, he ought not to tell them he is coming.*) and
  ax4 : "⌊¬go⌋l" (*He does not go.*)
 begin (* Consistency is confirmed by Nitpick *)
  lemma True nitpick [satisfy,card=4]  oops
  lemma assumes
       "⌊◇(go ∧ tell)⌋" and "⌊◇(go ∧ ¬tell)⌋" and "⌊◇(¬go ∧ tell)⌋" and "⌊◇(¬go ∧ ¬tell)⌋"
        and limited: "(∀φ. (∃x. (φ)x) ⟶ (∃x. opt<φ>x))"
```

```
Nitpick found a model for card i = 4:

  Constants:
    go = (λx. _)(i1 := False, i2 := False, i3 := False, i4 := True)
    tell = (λx. _)(i1 := False, i2 := False, i3 := False, i4 := True)
    aw = i3
    (r) =
      (λx. _)
      ((i1, i1) := False, (i1, i2) := True, (i1, i3) := False, (i1, i4) := True,
       (i2, i1) := False, (i2, i2) := True, (i2, i3) := True, (i2, i4) := False,
       (i3, i1) := True, (i3, i2) := False, (i3, i3) := False, (i3, i4) := False,
       (i4, i1) := False, (i4, i2) := True, (i4, i3) := False, (i4, i4) := True)
```

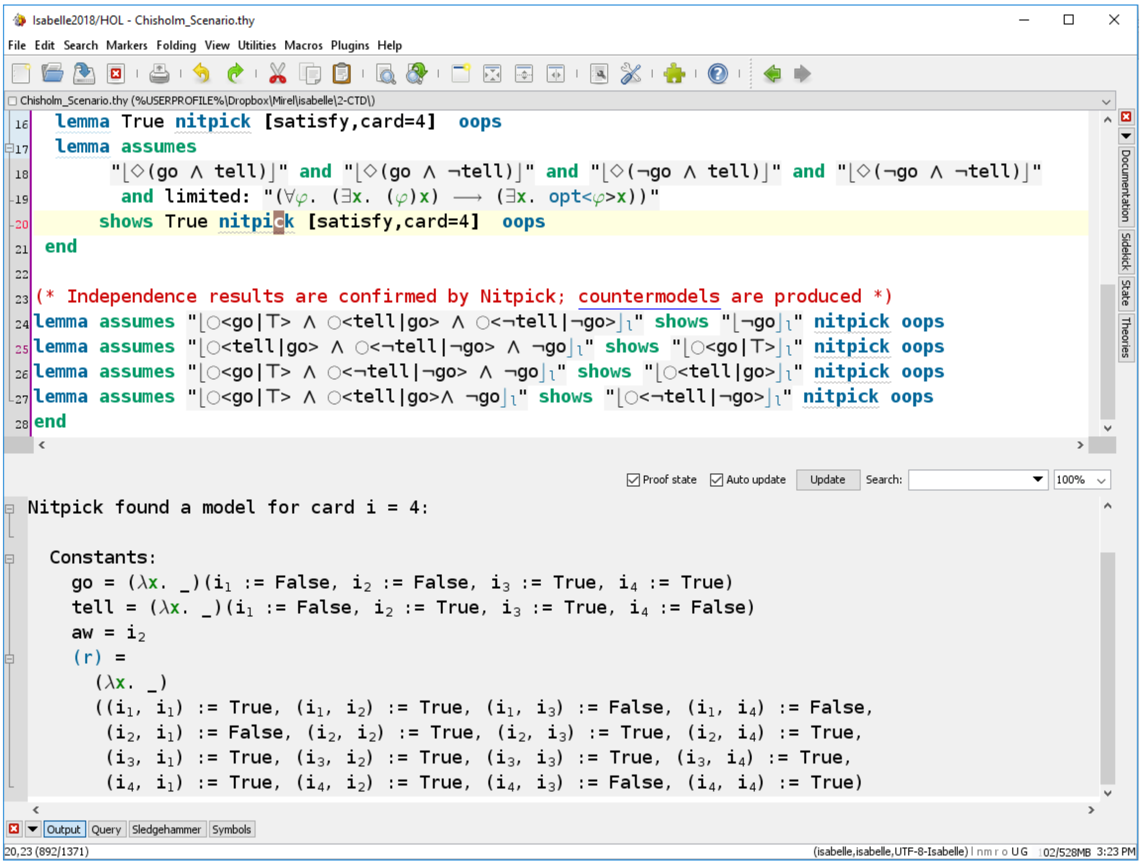
16,18 (669/1371)   (isabelle,isabelle,UTF-8-Isabelle) | nm r o UG 164/507MB 3:21 PM

FIGURE 6.2: The Chisholm's scenario encoded in system **E**

We can add more assumptions to the Chisholm's scenario ($\Diamond(go \wedge tell)$, $\Diamond(go \wedge \neg tell)$, $\Diamond(\neg go \wedge tell)$, $\Diamond(\neg go \wedge \neg tell)$ and the limitedness) as displayed in Figure 6.3, lines 18 and 19. In this case, Nitpick finds a more intuitive model M_2 with four possible worlds $\{i_1, i_2, i_3, i_4\}$ such that $\succeq = \{(i_1, i_1), (i_1, i_2), (i_2, i_2), (i_2, i_3), (i_2, i_4), (i_3, i_1), (i_3, i_2), (i_3, i_3), (i_3, i_4), (i_4, i_1), (i_4, i_2), (i_4, i_4)\}$. The valuation function is $V(go) = \{i_3, i_4\}$ and $V(tell) = \{i_2, i_3\}$. Since $\text{opt}_{\succeq}(V(\top)) = \{i_3\} \subseteq V(go) = \{i_3, i_4\}$, we have $M_2, i_2 \models \bigcirc(go/\top)$. Similarly, $\text{opt}_{\succeq}(V(go)) = \{i_3\} \subseteq V(tell) = \{i_2, i_3\}$ and $\text{opt}_{\succeq}(V(\neg go)) = \{i_1\} \subseteq V(\neg tell) = \{i_1\}$ implies $M_2, i_2 \models \bigcirc(tell/go)$ and $M_2, i_2 \models \bigcirc(\neg tell/\neg go)$ by the evaluation rule for dyadic obligation $\bigcirc$.

Moreover, the lines 24 to 27 in Figure 6.3 confirm that Chisholm's sentences are logically independent in system **E**. For example, in the line 25 Nitpick finds a countermodels that state formula $\bigcirc(go/\top)$ is logically independent from the set of formulas

$\{\bigcirc(tell/go), \bigcirc(\neg tell/\neg go), \neg go\}$.


Isabelle2018/HOL - Chisholm_Scenario.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
Chisholm_Scenario.thy (%USERPROFILE%\Dropbox\Mirel\isabelle\2-CTD\)
Documentation
Sidekick
State
Theories
Update
Output
Query
Sledgehammer
Symbols

```
  lemma True nitpick [satisfy,card=4]  oops
  lemma assumes
       "⌊◇(go ∧ tell)⌋" and "⌊◇(go ∧ ¬tell)⌋" and "⌊◇(¬go ∧ tell)⌋" and "⌊◇(¬go ∧ ¬tell)⌋"
        and limited: "(∀φ. (∃x. (φ)x) ⟶ (∃x. opt<φ>x))"
      shows True nitpick [satisfy,card=4]  oops
 end

(* Independence results are confirmed by Nitpick; countermodels are produced *)
lemma assumes "⌊○<go|T> ∧ ○<tell|go> ∧ ○<¬tell|¬go>⌋ι" shows "⌊¬go⌋ι" nitpick oops
lemma assumes "⌊○<tell|go> ∧ ○<¬tell|¬go> ∧ ¬go⌋ι" shows "⌊○<go|T>⌋ι" nitpick oops
lemma assumes "⌊○<go|T> ∧ ○<¬tell|¬go> ∧ ¬go⌋ι" shows "⌊○<tell|go>⌋ι" nitpick oops
lemma assumes "⌊○<go|T> ∧ ○<tell|go>∧ ¬go⌋ι" shows "⌊○<¬tell|¬go>⌋ι" nitpick oops
end
```

☑ Proof state ☑ Auto update Search: 100%

```
Nitpick found a model for card i = 4:

  Constants:
    go = (λx. _)(i1 := False, i2 := False, i3 := True, i4 := True)
    tell = (λx. _)(i1 := False, i2 := True, i3 := True, i4 := False)
    aw = i2
    (r) =
      (λx. _)
      ((i1, i1) := True, (i1, i2) := True, (i1, i3) := False, (i1, i4) := False,
       (i2, i1) := False, (i2, i2) := True, (i2, i3) := True, (i2, i4) := True,
       (i3, i1) := True, (i3, i2) := True, (i3, i3) := True, (i3, i4) := True,
       (i4, i1) := True, (i4, i2) := True, (i4, i3) := False, (i4, i4) := True)
```

20,23 (892/1371) (isabelle,isabelle,UTF-8-Isabelle) | nm r o UG 102/528MB 3:23 PM

FIGURE 6.3: The logically independence of Chisholm's statements in **E**

Reykjavic's scenario We look at another CTD example, due to Belzer [18]:

1. You should not tell the secret to Reagan.

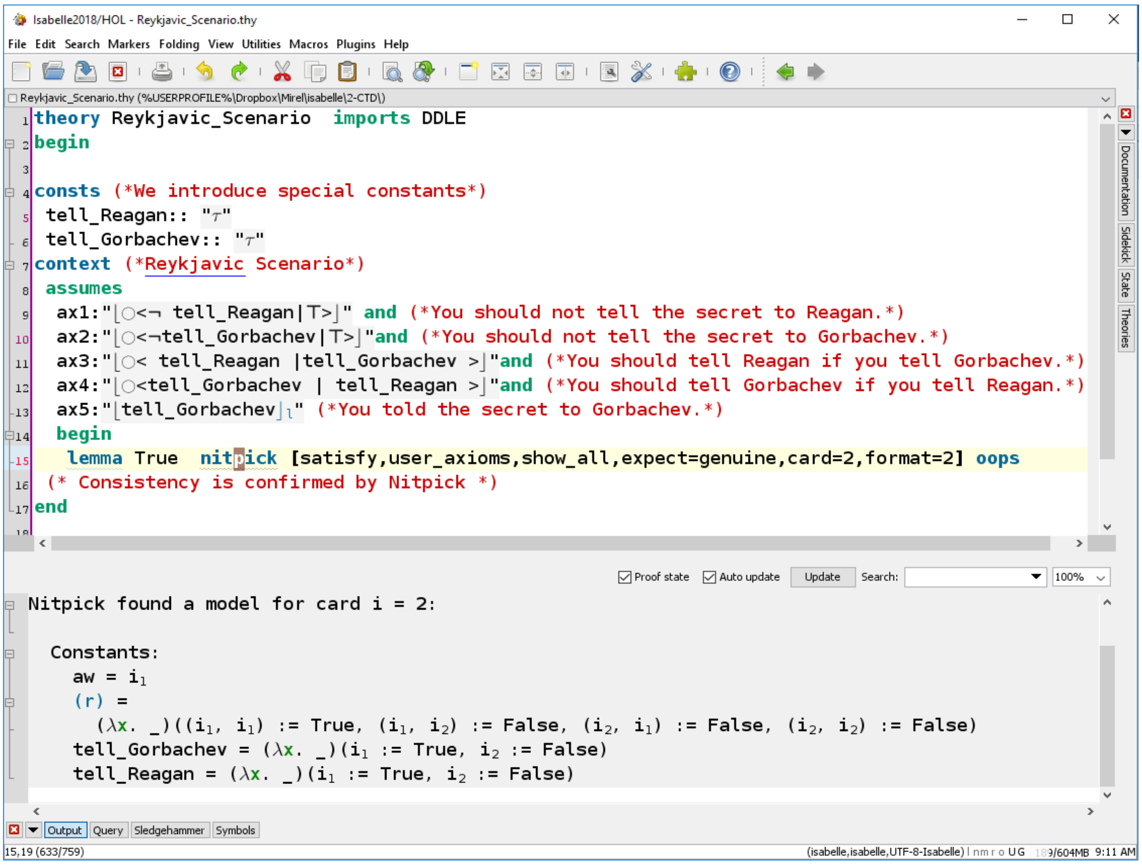
2. You should not tell the secret to Gorbachev.

3. You should tell Reagan if you tell Gorbachev.

4. You should tell Gorbachev if you tell Reagan.

5. You told the secret to Gorbachev.

There are two interpretations fo this paradox. Obligations 3 and 4 can be considered as overridden or CTD obligations [210]. We consider second interpretation here. Obligation $\bigcirc(tell\_Reagan/tell\_Gorbachev)$ is a CTD obligation of $\bigcirc(\neg tell\_Gorbachev/\top)$ and obligation $\bigcirc(tell\_Gorbachev\ /tell\_Reagan)$ is a CTD obligation of $\bigcirc(\neg tell\_Reagan/\top)$.



Isabelle2018/HOL - Reykjavic_Scenario.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
Reykjavic_Scenario.thy (%USERPROFILE%\Dropbox\Mirel\isabelle\2-CTD\)
Documentation
Sidekick
State
Theories

```
 1 theory Reykjavic_Scenario  imports DDLE
 2 begin
 3
 4 consts (*We introduce special constants*)
 5  tell_Reagan:: "τ"
 6  tell_Gorbachev:: "τ"
 7 context (*Reykjavic Scenario*)
 8  assumes
 9   ax1:"⌊○<¬ tell_Reagan|T>⌋" and (*You should not tell the secret to Reagan.*)
10   ax2:"⌊○<¬tell_Gorbachev|T>⌋"and (*You should not tell the secret to Gorbachev.*)
11   ax3:"⌊○< tell_Reagan |tell_Gorbachev >⌋"and (*You should tell Reagan if you tell Gorbachev.*)
12   ax4:"⌊○<tell_Gorbachev | tell_Reagan >⌋"and (*You should tell Gorbachev if you tell Reagan.*)
13   ax5:"⌊tell_Gorbachev⌋ι" (*You told the secret to Gorbachev.*)
14   begin
15    lemma True  nitpick [satisfy,user_axioms,show_all,expect=genuine,card=2,format=2] oops
16  (* Consistency is confirmed by Nitpick *)
17 end
```

```
Nitpick found a model for card i = 2:

  Constants:
    aw = i₁
    (r) =
      (λx. _)((i₁, i₁) := True, (i₁, i₂) := False, (i₂, i₁) := False, (i₂, i₂) := False)
    tell_Gorbachev = (λx. _)(i₁ := True, i₂ := False)
    tell_Reagan = (λx. _)(i₁ := True, i₂ := False)
```

Output
Query
Sledgehammer
Symbols
15,19 (633/759)
(isabelle,isabelle,UTF-8-Isabelle) | nm r o UG 189/604MB 9:11 AM

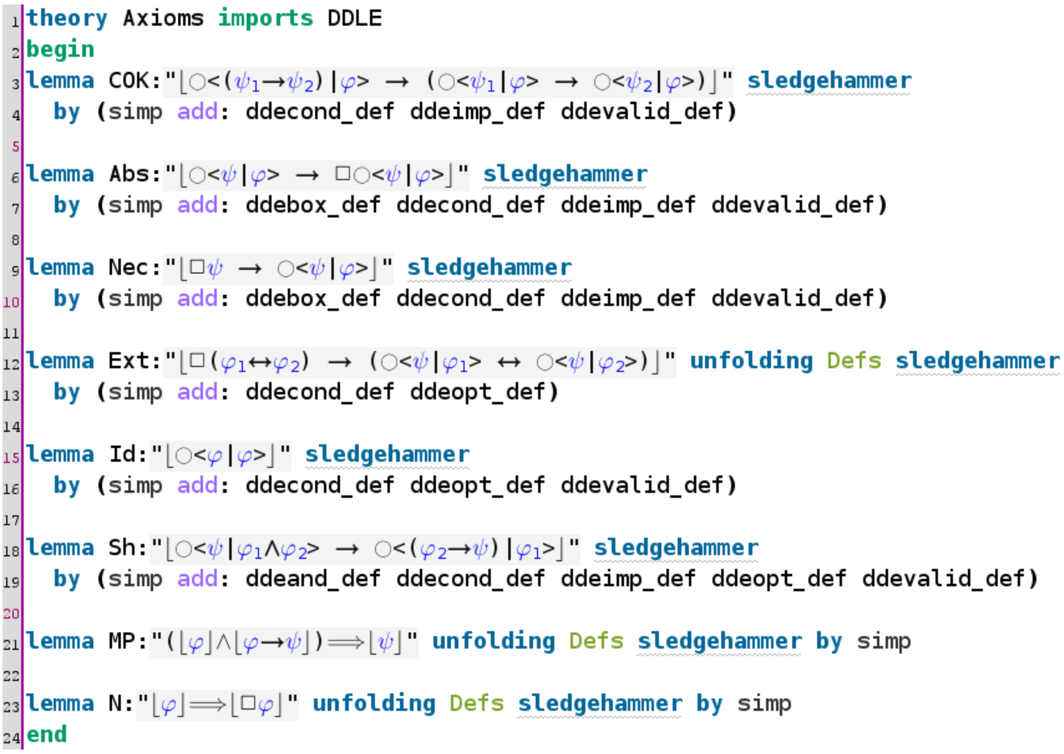
Figure 6.4: The Reykjavic's scenario encoded in system **E**

In Figure 6.4 Nitpick computes a model M consisting a set of possible words $\{i_1, i_2\}$ and the preference relation $\succeq = \{(i_1, i_1)\}$ in the line 15. The valuation function is $V(\mathit{tell\_Reagan}) = V(\mathit{tell\_Gorbachev}) = \{i_1\}$.

In this model, we have $\mathrm{opt}_{\succeq}(V(\top)) = \emptyset$. So, $M, i_1 \models \bigcirc(\neg \mathit{tell\_Reagan}/ \top)$ and $M, i_1 \models \bigcirc(\neg \mathit{tell\_Gorbachev}/\top)$ by the evaluation rule for $\bigcirc$. Moreover, $\mathrm{opt}_{\succeq}$ $(V(\mathit{tell\_Reagan}))$ $= \mathrm{opt}_{\succeq}(V(\mathit{tell\_Gorbachev})) = \{i_1\}$. So $M, i_1 \models \bigcirc(\mathit{tell\_Reagan}/ \mathit{tell\_Gorbachev})$ and $M, i_1 \models \bigcirc(\mathit{tell\_Gorbachev}\ /\mathit{tell\_Reagan})$.

6.3.3 Automatic verification of validities

Automatic verification of valid formulas is also possible. In Figure 6.5 Isabelle/HOL confirms the validity of each and every axiom and primitive rule of **E** by using the Sledgehammer tool [45] that gives access to automatic theorem provers (ATPs).



```
theory Axioms imports DDLE
begin
lemma COK:"⌊○<(ψ₁→ψ₂)|φ> → (○<ψ₁|φ> → ○<ψ₂|φ>)⌋" sledgehammer
  by (simp add: ddecond_def ddeimp_def ddevalid_def)

lemma Abs:"⌊○<ψ|φ> → □○<ψ|φ>⌋" sledgehammer
  by (simp add: ddebox_def ddecond_def ddeimp_def ddevalid_def)

lemma Nec:"⌊□ψ → ○<ψ|φ>⌋" sledgehammer
  by (simp add: ddebox_def ddecond_def ddeimp_def ddevalid_def)

lemma Ext:"⌊□(φ₁↔φ₂) → (○<ψ|φ₁> ↔ ○<ψ|φ₂>)⌋" unfolding Defs sledgehammer
  by (simp add: ddecond_def ddeopt_def)

lemma Id:"⌊○<φ|φ>⌋" sledgehammer
  by (simp add: ddecond_def ddeopt_def ddevalid_def)

lemma Sh:"⌊○<ψ|φ₁∧φ₂> → ○<(φ₂→ψ)|φ₁>⌋" sledgehammer
  by (simp add: ddeand_def ddecond_def ddeimp_def ddeopt_def ddevalid_def)

lemma MP:"(⌊φ⌋∧⌊φ→ψ⌋)⟹⌊ψ⌋" unfolding Defs sledgehammer by simp

lemma N:"⌊φ⌋⟹⌊□φ⌋" unfolding Defs sledgehammer by simp
end
```

FIGURE 6.5: Verifying the validity of the axioms and rules of system **E**

6.3.4 Nested dyadic deontic operator in system E

A particular focus of our experiments is on nested dyadic obligations. Belanyek et al. [17] proved any formula in system **G** is equivalent to some formula with no nesting based on the similar result for epistemic logic [148]. This is done based on the definition of the unnested disjunctive normal form (UDNF) of formulas [148], the following lemmas, and the reduction of a modal operator to a dyadic obligation operator as a particular case in system **G**. Similarly, by using the same form of formulas (UDNF), we can show that nested dyadic obligations in system **E** also can be eliminated by the help of Isabelle/HOL.

Here, we show that the reduction laws for nesting dyadic deontic logic hold for system **E** also.

Definition 6.6 (Unnested disjunctive normal form (**UDNF**))**.** Let $\mathcal{L}^O$ be the sublanguage of $\mathcal{L}$ (for system **E**) without formula containing the □-operator. We say that a formula $\psi \in \mathcal{L}^O$ is in Unnested Disjunctive Normal Form (**UDNF**) if it is a disjunction of conjunctions of the form
$\delta = \alpha \wedge \bigcirc(\psi_1/\varphi_1) \wedge ... \wedge \bigcirc(\psi_n/\varphi_n) \wedge \neg \bigcirc (\psi_{n+1}/\varphi_{n+1}) \wedge ... \wedge \neg \bigcirc (\psi_{n+k}/\varphi_{n+k})$
where $n; k \in \mathbb{N}$ and all of the formula $\alpha, \varphi_m, \psi_m (m \leq n+k)$ are propositional formulas

($\bot$ and $\top$ are considered propositional formula). The formula δ is called a canonical conjunction and the formula $\bigcirc(\psi_m/\varphi_m)$ is called prenex formula [148].

Lemma 6.7 (Meyer and van der Hoek [148] Lemma 1.7.6.2)**.** *If ψ is in **UDNF** and contains a prenex formula σ, then ψ is equivalent to a formula of the form $\psi = \pi \vee (\chi \wedge \sigma)$ where π, χ, and σ are all in **UDNF**.*

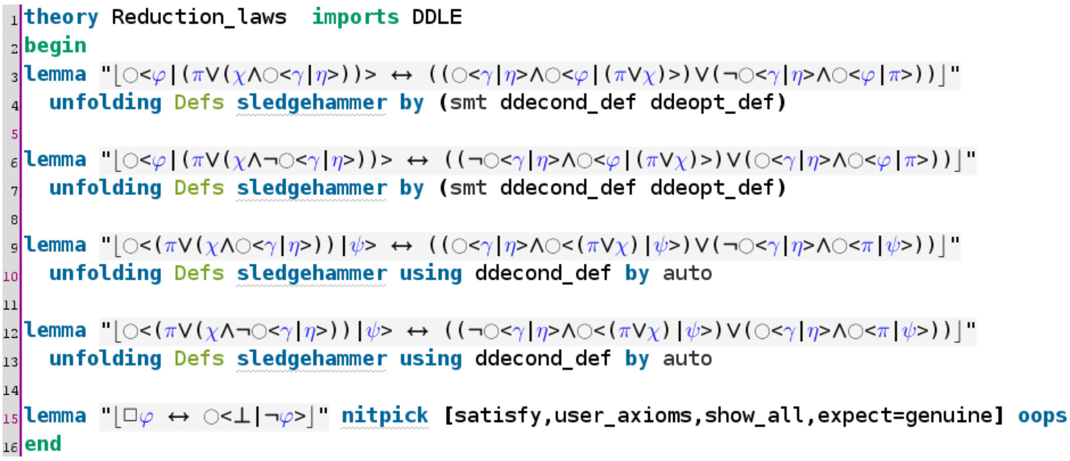
Proof. See [17] or Lemma 1.7.6.2 [148]. □

Lemma 6.8. *For arbitrary formula $\varphi, \pi, \chi, \gamma$ and η in system **E** we have:*

$$\bigcirc(\varphi/(\pi \vee (\chi \wedge \bigcirc(\gamma/\eta)))) \leftrightarrow ((\bigcirc(\gamma/\eta) \wedge \bigcirc(\varphi/(\pi \vee \chi))) \vee (\neg \bigcirc (\gamma/\eta) \wedge \bigcirc(\varphi/\pi)))$$
$$\bigcirc(\varphi/(\pi \vee (\chi \wedge \neg \bigcirc (\gamma/\eta)))) \leftrightarrow ((\neg \bigcirc (\gamma/\eta) \wedge \bigcirc(\varphi/(\pi \vee \chi))) \vee (\bigcirc(\gamma/\eta) \wedge \bigcirc(\varphi/\pi)))$$
$$\bigcirc(\pi \vee (\chi \wedge \bigcirc(\gamma/\eta))/\psi) \leftrightarrow ((\bigcirc(\gamma/\eta) \wedge \bigcirc(\pi \vee \chi/\psi)) \vee (\neg \bigcirc (\gamma/\eta) \wedge \bigcirc(\pi/\psi)))$$
$$\bigcirc(\pi \vee (\chi \wedge \neg \bigcirc (\gamma/\eta))/\psi) \leftrightarrow ((\neg \bigcirc (\gamma/\eta) \wedge \bigcirc(\pi \vee \chi/\psi)) \vee (\bigcirc(\gamma/\eta) \wedge \bigcirc(\pi/\psi)))$$

*Belanyek et al. [45] showed that these reduction laws hold in system **G**. Isabelle/HOL confirms that they also hold in the system **E**.*

Proof. The proof is confirmed by using the Sledgehammer tool [45] that give access to automatic theorem provers (ATPs) in Isabelle/HOL; cf. Fig. 6.6. Figure 6.6 gives the example of four "reduction" laws identified by Belanyek et al. [17]. They use these reduction laws to establish a more general result concerning iterated modalities in **G**, to the effect that any formula containing nested modal operators is equivalent to some formula with no nesting. On lines 3-13 in Figure 6.6, Isabelle/HOL confirms that the proofs of these equivalences carry over from **G** to **E**. □



```
1  theory Reduction_laws  imports DDLE
2  begin
3  lemma "⌊○<φ|(πV(χ∧○<γ|η>))> ↔ ((○<γ|η>∧○<φ|(πVχ)>)V(¬○<γ|η>∧○<φ|π>))⌋"
4    unfolding Defs sledgehammer by (smt ddecond_def ddeopt_def)
5
6  lemma "⌊○<φ|(πV(χ∧¬○<γ|η>))> ↔ ((¬○<γ|η>∧○<φ|(πVχ)>)V(○<γ|η>∧○<φ|π>))⌋"
7    unfolding Defs sledgehammer by (smt ddecond_def ddeopt_def)
8
9  lemma "⌊○<(πV(χ∧○<γ|η>))|ψ> ↔ ((○<γ|η>∧○<(πVχ)|ψ>)V(¬○<γ|η>∧○<π|ψ>))⌋"
10   unfolding Defs sledgehammer using ddecond_def by auto
11
12 lemma "⌊○<(πV(χ∧¬○<γ|η>))|ψ> ↔ ((¬○<γ|η>∧○<(πVχ)|ψ>)V(○<γ|η>∧○<π|ψ>))⌋"
13   unfolding Defs sledgehammer using ddecond_def by auto
14
15 lemma "⌊□φ ↔ ○<⊥|¬φ>⌋" nitpick [satisfy,user_axioms,show_all,expect=genuine] oops
16 end
```

FIGURE 6.6: Reduction laws in system **E**

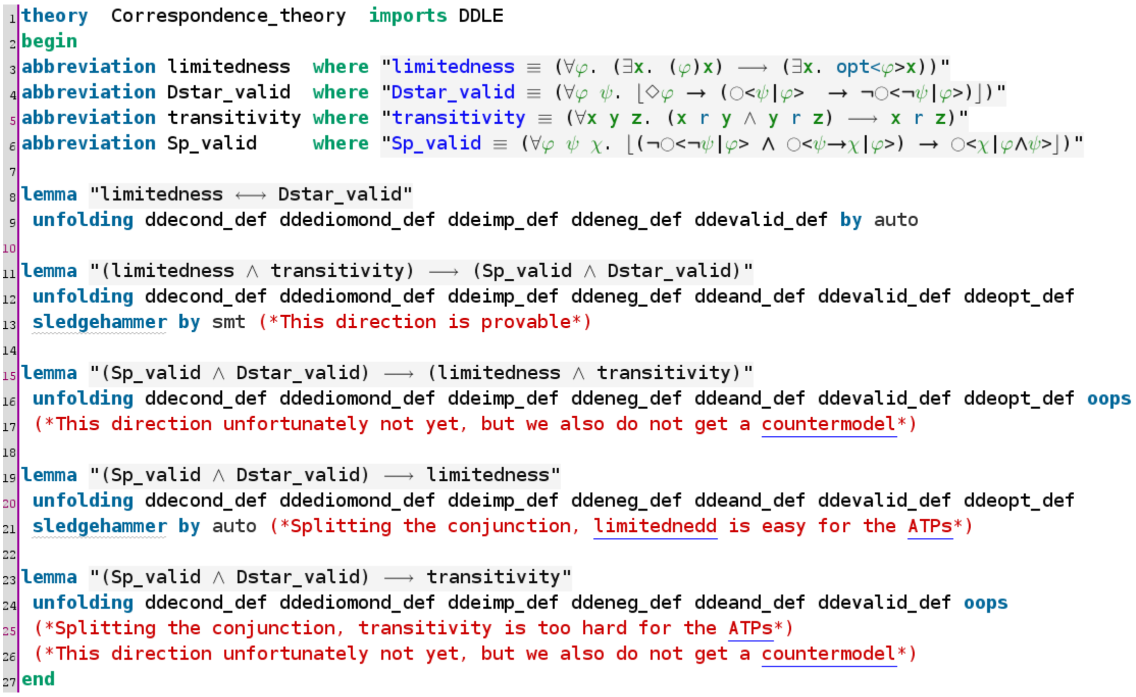
Theorem 6.9. *For every $\lambda \in \mathcal{L}^O$, there exist a formula λ' such that λ' is in **UDNF** and $\models \lambda \leftrightarrow \lambda'$.*

Proof. See [17]. □

However, the more general result concerning iterated modalities does not. To establish that one, the authors appeal to the fact that in **G**, $\Box$ is definable in terms of $\bigcirc(\_/\_)$: $\Box\varphi \leftrightarrow \bigcirc(\bot/\neg\varphi)$. Nitpick confirms that this equivalence is falsifiable in the class of all preference models (line 15).

6.3.5 Correspondence theory

The aim of correspondence theory is to establish connections between properties of Kripke frames and the formulas in modal logic that are true in all Kripke frames with these properties. Figure 6.7 shows some experimentations in correspondence theory. Lines 8–9 tell us that limitedness is equivalent with (and thus corresponds to) D^*. Lines 11–13 tell us that limitedness and transitivity are conjointly enough to get both D^* and Sp. However, on lines 15–16, Isabelle/HOL fails to show that they are necessary conditions. The problem is with the proof of the property of transitivity (lines 23–24). The good news is: we do not get a counter-model to the implication (calls for countermodel search with nitpick are not displayed here).



```
theory  Correspondence_theory  imports DDLE
begin
abbreviation limitedness  where "limitedness ≡ (∀φ. (∃x. (φ)x) ⟶ (∃x. opt<φ>x))"
abbreviation Dstar_valid  where "Dstar_valid ≡ (∀φ ψ. ⌊◇φ → (○<ψ|φ>  → ¬○<¬ψ|φ>)⌋)"
abbreviation transitivity where "transitivity ≡ (∀x y z. (x r y ∧ y r z) ⟶ x r z)"
abbreviation Sp_valid     where "Sp_valid ≡ (∀φ ψ χ. ⌊(¬○<¬ψ|φ> ∧ ○<ψ→χ|φ>) → ○<χ|φ∧ψ>⌋)"

lemma "limitedness ⟷ Dstar_valid"
 unfolding ddecond_def ddediomond_def ddeimp_def ddeneg_def ddevalid_def by auto

lemma "(limitedness ∧ transitivity) ⟶ (Sp_valid ∧ Dstar_valid)"
 unfolding ddecond_def ddediomond_def ddeimp_def ddeneg_def ddeand_def ddevalid_def ddeopt_def
 sledgehammer by smt (*This direction is provable*)

lemma "(Sp_valid ∧ Dstar_valid) ⟶ (limitedness ∧ transitivity)"
 unfolding ddecond_def ddediomond_def ddeimp_def ddeneg_def ddeand_def ddevalid_def ddeopt_def oops
 (*This direction unfortunately not yet, but we also do not get a countermodel*)

lemma "(Sp_valid ∧ Dstar_valid) ⟶ limitedness"
 unfolding ddecond_def ddediomond_def ddeimp_def ddeneg_def ddeand_def ddevalid_def ddeopt_def
 sledgehammer by auto (*Splitting the conjunction, limitednedd is easy for the ATPs*)

lemma "(Sp_valid ∧ Dstar_valid) ⟶ transitivity"
 unfolding ddecond_def ddediomond_def ddeimp_def ddeneg_def ddeand_def ddevalid_def oops
 (*Splitting the conjunction, transitivity is too hard for the ATPs*)
 (*This direction unfortunately not yet, but we also do not get a countermodel*)
end
```

FIGURE 6.7: Experiments in correspondence theory

6.4 A Brief Study of Unification Problem in Deontic Logic

We are interested in studying and comparing the implemented logics in this thesis with the semantics introduced by Kratzer, which seems is one of the best options for deontic semantic unification. Horty [112] compared Kratzer semantics to SDL and van Fraassen [216] system and found some connections between them. Comparison of the Kratzerian framework with other well-known deontic logics such as dyadic deontic logic by Åqvist is open. In this section, we study and compare Åqvist dyadic deontic logic with Kratzer's system KD [112]. Horty [112] found the relation between Kratzer's system KD and SDL.

6.4.1 Kratzer's system KD

We discussed Kratzer semantics in Section 1.3 and Section 3.1. A Kratzer model is a structure $M = \langle W, f, g, v \rangle$, where f and g are functions from worlds to set of propositions. f is the modal base function, and g is the ordering source function. g ranks the worlds as follows:

$$s \succeq_{g(w)} t \text{ iff, for all } X \in g(w), \text{ if } t \in X, \text{ then } s \in X.$$

$$(s \succ_{g(w)} t \text{ iff } s \succeq_{g(w)} t \text{ and } t \not\succeq_{g(w)} s\ )$$

Best worlds are defined as usual : $Best_{g(w)}(X) = \{s \in X : \neg\exists t \in X(t \succ_{g(w)} s)\}$. In this chapter we restrict Kratzer model with stoppered property. A Kratzer model is stoppered if and only if for all w if $s \in \bigcap f(w)$ then $s \in Best_{g(w)}(\bigcap f(w))$ or $\exists t.t \in Best_{g(w)}(\bigcap f(w)) \wedge t \succeq_{g(w)} s$ (moreover, we can add the consistency condition $\bigcap f(w) \neq \emptyset$).

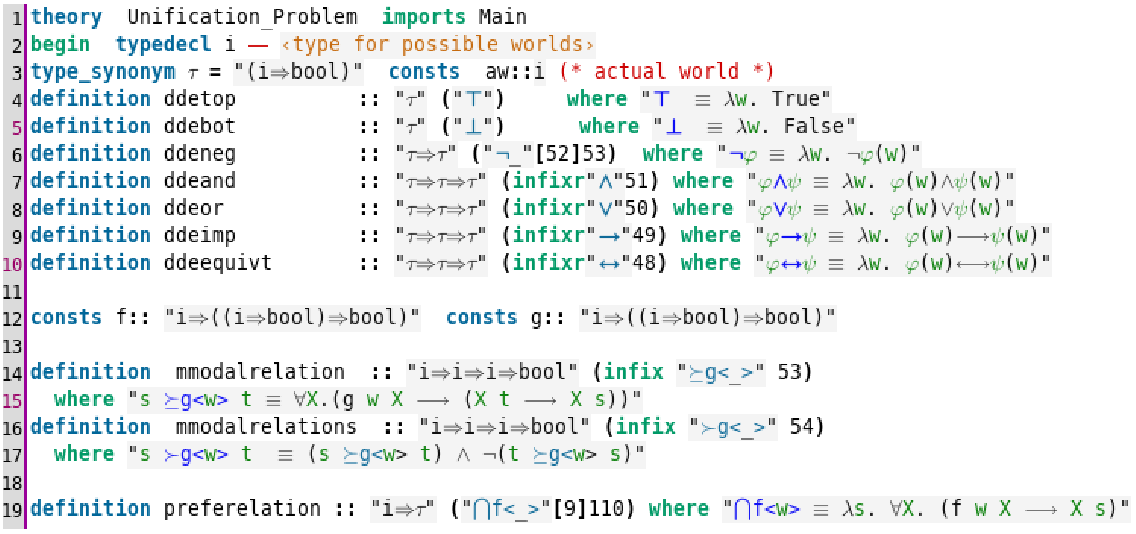
The satisfaction relation for monadic obligation [112] can be defined as follows:

$$M, w \models \bigcirc\varphi \quad \text{if and only if} \quad Best_{g(w)}(\bigcap f(w)) \subseteq V(\varphi)$$

and for dyadic obligation [112] as follows:

$$M, w \models \bigcirc(\psi/\varphi) \quad \text{if and only if} \quad Best_{g(w)}(\bigcap f(w) \cap V(\varphi)) \subseteq V(\psi)$$

Kratzer's system classical logical operators is implemented in Isabelle/HOL (cf. Fig. 6.8).



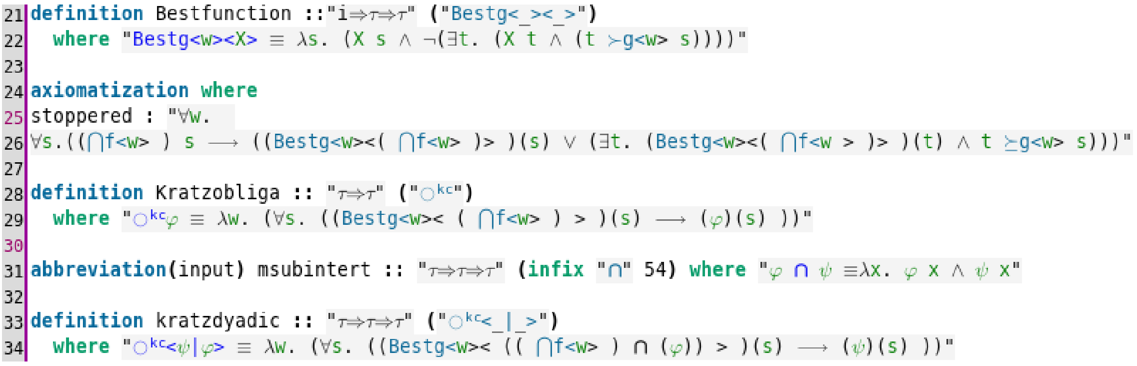
```
theory  Unification_Problem  imports Main
begin  typedecl i — ‹type for possible worlds›
type_synonym τ = "(i⇒bool)"  consts  aw::i (* actual world *)
definition ddetop           :: "τ" ("⊤")     where "⊤  ≡ λw. True"
definition ddebot           :: "τ" ("⊥")      where "⊥  ≡ λw. False"
definition ddeneg           :: "τ⇒τ" ("¬_"[52]53)  where "¬φ ≡ λw. ¬φ(w)"
definition ddeand           :: "τ⇒τ⇒τ" (infixr"∧"51) where "φ∧ψ ≡ λw. φ(w)∧ψ(w)"
definition ddeor            :: "τ⇒τ⇒τ" (infixr"∨"50) where "φ∨ψ ≡ λw. φ(w)∨ψ(w)"
definition ddeimp           :: "τ⇒τ⇒τ" (infixr"→"49) where "φ→ψ ≡ λw. φ(w)⟶ψ(w)"
definition ddeequivt        :: "τ⇒τ⇒τ" (infixr"↔"48) where "φ↔ψ ≡ λw. φ(w)⟷ψ(w)"

consts f:: "i⇒((i⇒bool)⇒bool)"  consts g:: "i⇒((i⇒bool)⇒bool)"

definition  mmodalrelation  :: "i⇒i⇒i⇒bool" (infix "⪰g<_>" 53)
  where "s ⪰g<w> t ≡ ∀X.(g w X ⟶ (X t ⟶ X s))"
definition  mmodalrelations  :: "i⇒i⇒i⇒bool" (infix "≻g<_>" 54)
  where "s ≻g<w> t  ≡ (s ⪰g<w> t) ∧ ¬(t ⪰g<w> s)"

definition preferelation :: "i⇒τ" ("⋂f<_>"[9]110) where "⋂f<w> ≡ λs. ∀X. (f w X ⟶ X s)"
```

FIGURE 6.8: Semantical embedding of Kratzer's system KD in Isabelle/HOL

Moreover, we can define Kratzer style monadic and dyadic deontic operators in HOL and consequently in Isabelle/HOL (cf. Fig. 6.9).

6.4.2 Comparing system E and Kratzer's system KD

We can characterize system **E** within the Kratzerian framework. It is based on using $\text{opt}_{\succeq}$ operator for defining both modal and ordering sources. The natural modal base for the current world is the most preferable accessible worlds i.e. $\bigcap f(w) = \text{opt}_{\succeq}(R(w))$ (cf. Fig. 6.10, line 35).

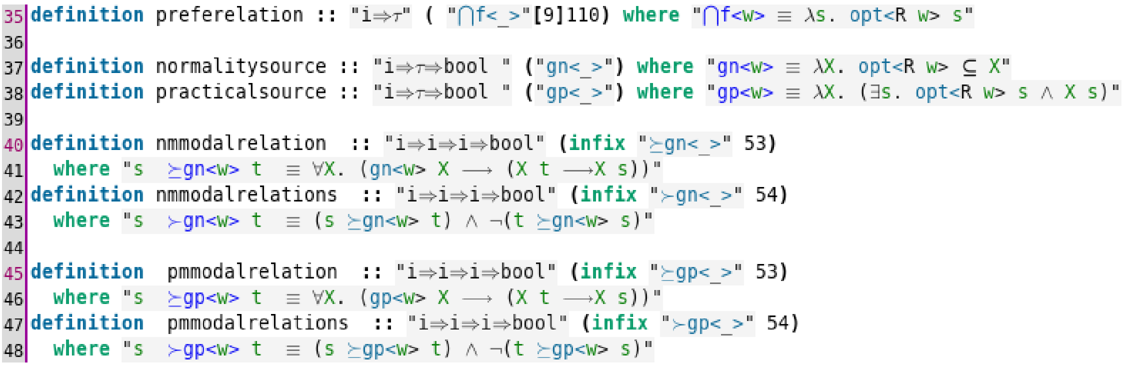


```
definition Bestfunction ::"i⇒τ⇒τ" ("Bestg<_><_>")
  where "Bestg<w><X> ≡ λs. (X s ∧ ¬(∃t. (X t ∧ (t ≻g<w> s))))"

axiomatization where
stoppered : "∀w.
∀s.((⋂f<w> ) s ⟶ ((Bestg<w><( ⋂f<w> )> )(s) ∨ (∃t. (Bestg<w><( ⋂f<w > )> )(t) ∧ t ≽g<w> s)))"

definition Kratzobliga :: "τ⇒τ" ("○kc")
  where "○kcφ ≡ λw. (∀s. ((Bestg<w>< ( ⋂f<w> ) > )(s) ⟶ (φ)(s) ))"

abbreviation(input) msubintert :: "τ⇒τ⇒τ" (infix "∩" 54) where "φ ∩ ψ ≡λx. φ x ∧ ψ x"

definition kratzdyadic :: "τ⇒τ⇒τ" ("○kc<_|_>")
  where "○kc<ψ|φ> ≡ λw. (∀s. ((Bestg<w>< (( ⋂f<w> ) ∩ (φ)) > )(s) ⟶ (ψ)(s) ))"
```

FIGURE 6.9: Semantical embedding of Kratzer's system KD in Isabelle/HOL

Moreover, there are two suggested ways of defining ordering sources (cf. Fig. 6.10, lines 37 and 38).

- Entailment : $g(w) = \{X : \text{opt}_{\succeq}(R(w)) \subseteq X\}$
- Compatibility: $g(w) = \{X : \text{opt}_{\succeq}(R(w)) \cap X \neq \emptyset\}$



```
35 definition preferelation :: "i⇒τ" ( "⋂f<_>"[9]110) where "⋂f<w> ≡ λs. opt<R w> s"
36
37 definition normalitysource :: "i⇒τ⇒bool " ("gn<_>") where "gn<w> ≡ λX. opt<R w> ⊆ X"
38 definition practicalsource :: "i⇒τ⇒bool " ("gp<_>") where "gp<w> ≡ λX. (∃s. opt<R w> s ∧ X s)"
39
40 definition nmmodalrelation  :: "i⇒i⇒i⇒bool" (infix "≽gn<_>" 53)
41   where "s  ≽gn<w> t  ≡ ∀X. (gn<w> X ⟶ (X t ⟶X s))"
42 definition nmmodalrelations  :: "i⇒i⇒i⇒bool" (infix "≻gn<_>" 54)
43   where "s  ≻gn<w> t  ≡ (s ≽gn<w> t) ∧ ¬(t ≽gn<w> s)"
44
45 definition  pmmodalrelation  :: "i⇒i⇒i⇒bool" (infix "≽gp<_>" 53)
46   where "s  ≽gp<w> t  ≡ ∀X. (gp<w> X ⟶ (X t ⟶X s))"
47 definition  pmmodalrelations  :: "i⇒i⇒i⇒bool" (infix "≻gp<_>" 54)
48   where "s  ≻gp<w> t  ≡ (s ≽gp<w> t) ∧ ¬(t ≽gp<w> s)"
```

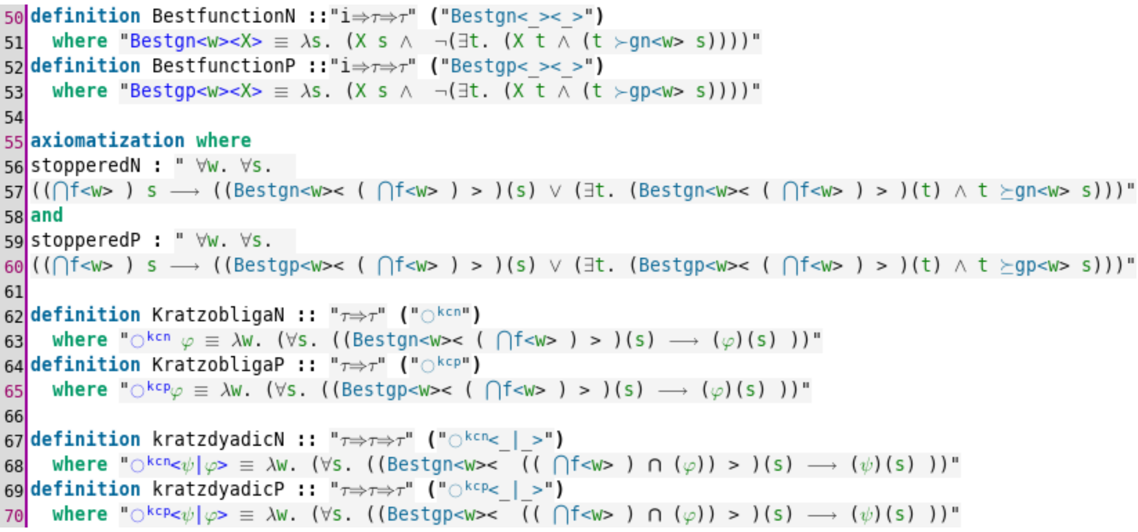
FIGURE 6.10: Comparing system **E** and Kratzer's system KD in Isabelle/HOL

We can define two different monadic and dyadic deontic operators (cf. Fig. 6.11 lines 62–65 and 67–70) based on two various ordering sources.

Identity property fails for both monadic operators and factual detachment holds for both dyadic operators (cf. Fig. 6.12).

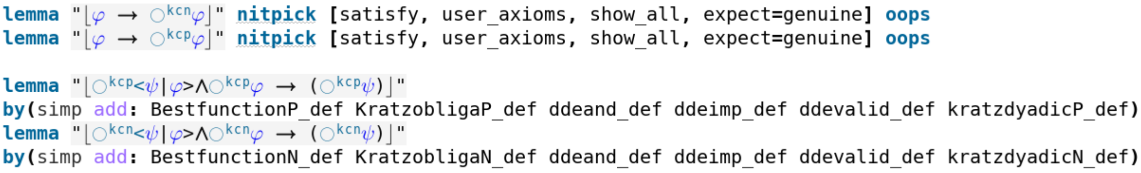
Isabelle/HOL shows that the initial dyadic operator in system **E** is the same as both newly introduced dyadic operators in each world. This guaranty our approach as a faithful generalization of Åqvist dyadic deontic logic in the Kratzerian framework (cf. Fig. 6.13). The results hold with removing the stoppered property.

The last interesting point is that both different ordering sources define the same monadic and dyadic deontic operators (cf. Fig. 6.14).



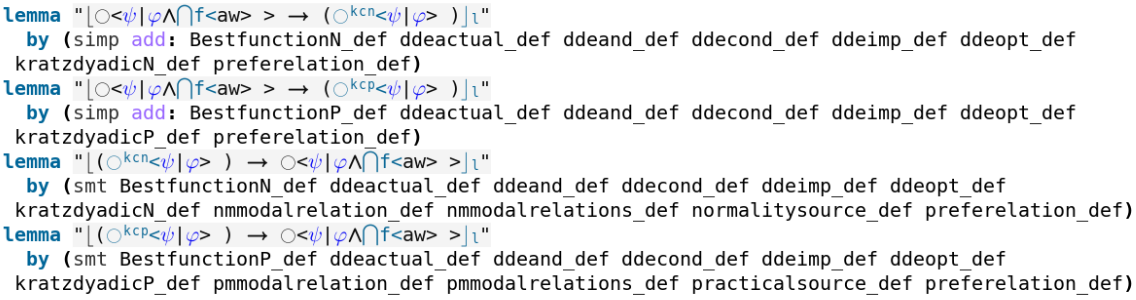
```
definition BestfunctionN ::"i⇒τ⇒τ"  ("Bestgn<_><_>")
  where "Bestgn<w><X> ≡ λs. (X s ∧  ¬(∃t. (X t ∧ (t ≻gn<w> s))))"
definition BestfunctionP ::"i⇒τ⇒τ"  ("Bestgp<_><_>")
  where "Bestgp<w><X> ≡ λs. (X s ∧  ¬(∃t. (X t ∧ (t ≻gp<w> s))))"

axiomatization where
stopperedN : " ∀w. ∀s.
((⋂f<w> ) s ⟶ ((Bestgn<w>< ( ⋂f<w> ) > )(s) ∨ (∃t. (Bestgn<w>< ( ⋂f<w> ) > )(t) ∧ t ≽gn<w> s)))"
and
stopperedP : " ∀w. ∀s.
((⋂f<w> ) s ⟶ ((Bestgp<w>< ( ⋂f<w> ) > )(s) ∨ (∃t. (Bestgp<w>< ( ⋂f<w> ) > )(t) ∧ t ≽gp<w> s)))"

definition KratzobligaN :: "τ⇒τ" ("○kcn")
  where "○kcn φ ≡ λw. (∀s. ((Bestgn<w>< ( ⋂f<w> ) > )(s) ⟶ (φ)(s) ))"
definition KratzobligaP :: "τ⇒τ" ("○kcp")
  where "○kcpφ ≡ λw. (∀s. ((Bestgp<w>< ( ⋂f<w> ) > )(s) ⟶ (φ)(s) ))"

definition kratzdyadicN :: "τ⇒τ⇒τ" ("○kcn<_|_>")
  where "○kcn<ψ|φ> ≡ λw. (∀s. ((Bestgn<w><  (( ⋂f<w> ) ∩ (φ)) > )(s) ⟶ (ψ)(s) ))"
definition kratzdyadicP :: "τ⇒τ⇒τ" ("○kcp<_|_>")
  where "○kcp<ψ|φ> ≡ λw. (∀s. ((Bestgp<w><  (( ⋂f<w> ) ∩ (φ)) > )(s) ⟶ (ψ)(s) ))"
```

Figure 6.11: Comparing system **E** and Kratzer's system KD in Isabelle/HOL



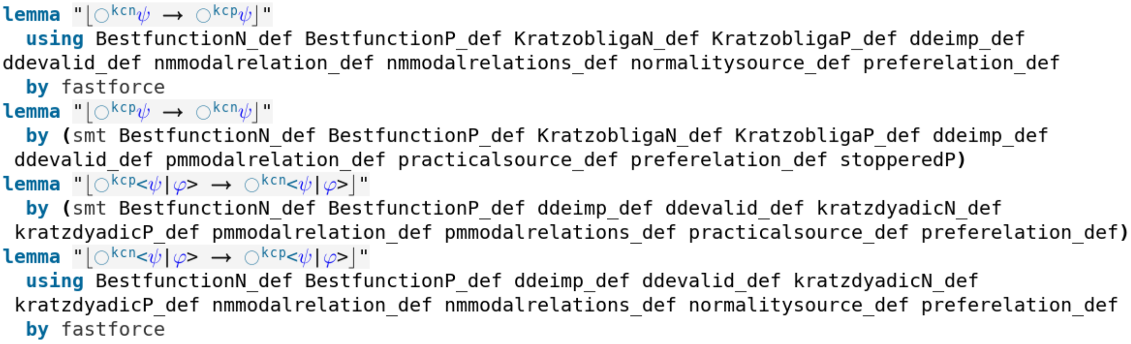
```
lemma "⌊φ ⟶ ○kcnφ⌋" nitpick [satisfy, user_axioms, show_all, expect=genuine] oops
lemma "⌊φ ⟶ ○kcpφ⌋" nitpick [satisfy, user_axioms, show_all, expect=genuine] oops

lemma "⌊○kcp<ψ|φ>∧○kcpφ ⟶ (○kcpψ)⌋"
by(simp add: BestfunctionP_def KratzobligaP_def ddeand_def ddeimp_def ddevalid_def kratzdyadicP_def)
lemma "⌊○kcn<ψ|φ>∧○kcnφ ⟶ (○kcnψ)⌋"
by(simp add: BestfunctionN_def KratzobligaN_def ddeand_def ddeimp_def ddevalid_def kratzdyadicN_def)
```

Figure 6.12: Comparing system **E** and Kratzer's system KD in Isabelle/HOL



```
lemma "⌊○<ψ|φ∧⋂f<aw> > ⟶ (○kcn<ψ|φ> )⌋ι"
  by (simp add: BestfunctionN_def ddeactual_def ddeand_def ddecond_def ddeimp_def ddeopt_def
 kratzdyadicN_def preferelation_def)
lemma "⌊○<ψ|φ∧⋂f<aw> > ⟶ (○kcp<ψ|φ> )⌋ι"
  by (simp add: BestfunctionP_def ddeactual_def ddeand_def ddecond_def ddeimp_def ddeopt_def
 kratzdyadicP_def preferelation_def)
lemma "⌊(○kcn<ψ|φ> ) ⟶ ○<ψ|φ∧⋂f<aw> >⌋ι"
  by (smt BestfunctionN_def ddeactual_def ddeand_def ddecond_def ddeimp_def ddeopt_def
 kratzdyadicN_def nmmodalrelation_def nmmodalrelations_def normalitysource_def preferelation_def)
lemma "⌊(○kcp<ψ|φ> ) ⟶ ○<ψ|φ∧⋂f<aw> >⌋ι"
  by (smt BestfunctionP_def ddeactual_def ddeand_def ddecond_def ddeimp_def ddeopt_def
 kratzdyadicP_def pmmodalrelation_def pmmodalrelations_def practicalsource_def preferelation_def)
```

Figure 6.13: Comparing system **E** and Kratzer's system KD in Isabelle/HOL



```
lemma "⌊○kcnψ ⟶ ○kcpψ⌋"
  using BestfunctionN_def BestfunctionP_def KratzobligaN_def KratzobligaP_def ddeimp_def
ddevalid_def nmmodalrelation_def nmmodalrelations_def normalitysource_def preferelation_def
  by fastforce
lemma "⌊○kcpψ ⟶ ○kcnψ⌋"
  by (smt BestfunctionN_def BestfunctionP_def KratzobligaN_def KratzobligaP_def ddeimp_def
 ddevalid_def pmmodalrelation_def practicalsource_def preferelation_def stopperedP)
lemma "⌊○kcp<ψ|φ> ⟶ ○kcn<ψ|φ>⌋"
  by (smt BestfunctionN_def BestfunctionP_def ddeimp_def ddevalid_def kratzdyadicN_def
 kratzdyadicP_def pmmodalrelation_def pmmodalrelations_def practicalsource_def preferelation_def)
lemma "⌊○kcn<ψ|φ> ⟶ ○kcp<ψ|φ>⌋"
  using BestfunctionN_def BestfunctionP_def ddeimp_def ddevalid_def kratzdyadicN_def
 kratzdyadicP_def nmmodalrelation_def nmmodalrelations_def normalitysource_def preferelation_def
  by fastforce
```

Figure 6.14: Comparing system **E** and Kratzer's system KD in Isabelle/HOL

6.5 Conclusion

A shallow semantical embedding of Åqvist's dyadic deontic logic **E** in classical higher-order logic has been presented and shown to be faithful (sound an complete). We have studied some meta-theoretical properties of systems **E**. A particular study was the characterization of the system **E** within the Kratzerian framework.

We end this chapter by listing a number of topics for future research. First, it would be worthwhile to study the shallow semantical embedding of the stronger systems **F** and **G** in HOL, and look at the three systems from the point of view of a semantics defining *best* in terms of maximality rather than optimality [164, 161]. We need to represent limitedness assumption into higher-order logic for a semantical embedding of system **F** and **G** in HOL. The translation of the limitedness assumption into HOL is not straightforward. We have suggested a replacement for this assumption that can easily be translated in HOL. The details are discussed in Appendix A.1. Second, we could employ our implementation to systematically study some meta-logical properties of these systems within Isabelle/HOL. Third, it would be interesting to consider the quantified extensions of system **E**, **F**, and **G**; previous work has focused on monadic modal logic and conditional logic [22, 23, 40].

Chapter 7

Deontic Modals, Circumstances, and Computation

> "*Our judgments about our actual duty in concrete situations, have none of the certainty that attaches to our recognition of general principles of duty [...] Where a possible act is seen to have two characteristics in virtue of one of which it is prima facie right and in virtue of the other prima facie wrong we are well aware that we are not certain whether we ought or ought not to do it. Whether we do it or not we are taking a moral risk.*"
>
> (Ross [81], p. 30)

A normative system generates prima facie obligations for agents. Normative systems for ideal agents may conflict in non-ideal circumstances. Then it is impossible to meet all the prima facie obligations they generate. Uncertainty is increased as to whether, in choosing x over y, the agent has done the right thing in facing a dilemma generated by prima facie obligations. As Carmo and Jones [67] put it

> "We need to be able to integrate in a single logical framework the ability to make deductions at two different levels: on the level of what *ideally* should be the case, and on the level of what *actually* should be the case, given the circumstances (where, of course, the circumstances might include the fact that what has happened deviates from the ideal). The simultaneous specification

> of both ideal behavior and of what to do when actual behavior deviates from the ideal is a central task of deontic logic.”

A particular dyadic deontic logic, tailored to so-called *prima facie* and *actual obligations* has been proposed by Carmo and Jones [67]. We shall refer to it as CJL in the remainder. CJL comes with a neighborhood semantics and a weakly complete axiomatization over the class of finite models. The framework is immune to the well-known contrary-to-duty paradoxes.

The chapter is structured as follows: Section 7.1 outlines CJL. The semantical embedding of CJL in HOL is then devised and studied in Section 7.2. This section also addresses soundness and completeness. Section 7.3 discusses the implementation and automation of the embedding in Isabelle/HOL [156] and Section 7.4 concludes the chapter.

7.1 The Dyadic Deontic Logic of Carmo and Jones

This section provides a concise introduction of CJL, the dyadic deontic logic proposed by Carmo and Jones. Definitions as required for the remainder are presented. For further details we refer to the literature [67, 66].

To define the formulas of CJL we start with an countable set of propositional symbols P, and we choose $\neg$ and $\vee$ as the only primitive connectives.

The set of *CJL formulas* is given as the smallest set of formulas obeying the following conditions:

- Each $p^i \in P$ is an (atomic) CJL formula.
- Given two arbitrary CJL formulas φ and ψ, then
  $\neg\varphi$ — *classical negation,*
  $\varphi \vee \psi$ — *classical disjunction,*
  $\bigcirc(\psi/\varphi)$ — *dyadic deontic obligation: “it ought to be ψ, given φ”,*
  $\Box\varphi$ — *in all worlds,*
  $\Box_a\varphi$ — *in all actual versions (open alternatives) of the current world,*
  $\Box_p\varphi$ — *in all potential versions of the current world,*
  $\bigcirc_a(\varphi)$ — *monadic deontic operator for actual obligation,* and
  $\bigcirc_p(\varphi)$ — *monadic deontic operator for primary obligation*

  are also CJL formulas.

Further logical connectives can be defined as usual. For example, we may define $\varphi \wedge \psi := \neg(\neg\varphi \vee \neg\psi)$, $\varphi \rightarrow \psi := \neg\varphi \vee \psi$, $\varphi \longleftrightarrow \psi := (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$, $\Diamond\varphi := \neg\Box\neg\varphi$, $\Diamond_a\varphi := \neg\Box_a\neg\varphi$, $\Diamond_p\varphi := \neg\Box_p\neg\varphi$, $\top := \neg q \vee q$, for some propositional symbol q, and $\bot := \neg\top$.

7.1.1 Semantics

A CJL *model* is a structure $M = \langle S, av, pv, ob, V\rangle$, where S is a non empty set of items called possible worlds, V is a function assigning a set of worlds to each atomic formula, that is, $V(p^j) \subseteq S$. av: $S \rightarrow P(S)$, where $P(S)$ denotes the power set of S, is a function mapping worlds to sets of worlds such that $av(s) \neq \emptyset$. $av(s)$ denotes the set of actual versions of the world s. pv: $S \rightarrow P(S)$ is another, similar mapping such that $av(s) \subseteq pv(s)$ and $s \in pv(s)$. $pv(s)$ denotes the set of potential versions of the world s. ob: $P(S) \rightarrow P(P(S))$, which denotes the set of propositions that are obligatory in context $\bar{X} \subseteq S$, is a function mapping set of worlds to sets of sets of worlds. The following conditions hold for ob (where $\bar{X}, \bar{Y}, \bar{Z}$ designate arbitrary subsets of S):

1. $\emptyset \notin ob(\bar{X})$.
2. If $\bar{Y} \cap \bar{X} = \bar{Z} \cap \bar{X}$, then $\bar{Y} \in ob(\bar{X})$ if and only if $\bar{Z} \in ob(\bar{X})$.
3. Let $\bar{\beta} \subseteq ob(\bar{X})$ and $\bar{\beta} \neq \emptyset$. If $(\cap\bar{\beta}) \cap \bar{X} \neq \emptyset$
   (where $\cap\bar{\beta} = \{s \in S \mid \text{for all } \bar{Z} \in \bar{\beta} \text{ we have } s \in \bar{Z}\}$), then $(\cap\bar{\beta}) \in ob(\bar{X})$.
4. If $\bar{Y} \subseteq \bar{X}$ and $\bar{Y} \in ob(\bar{X})$ and $\bar{X} \subseteq \bar{Z}$, then $(\bar{Z} \setminus \bar{X}) \cup \bar{Y} \in ob(\bar{Z})$.
5. If $\bar{Y} \subseteq \bar{X}$ and $\bar{Z} \in ob(\bar{X})$ and $\bar{Y} \cap \bar{Z} \neq \emptyset$, then $\bar{Z} \in ob(\bar{Y})$.

Satisfiability of a formula φ for a model $M = \langle S, av, pv, ob, V\rangle$ and a world $s \in S$ is denoted by $M, s \models \varphi$ and we define $V^M(\varphi) = \{s \in S \mid M, s \models \varphi\}$. In order to simplify the presentation, whenever the model M is obvious from context, we write $V(\varphi)$ instead of $V^M(\varphi)$. Moreover, we often use "iff" as shorthand for "if and only if".

$$
\begin{array}{lll}
M, s \models p^j & \text{iff} & s \in V(p^j) \\
M, s \models \neg\varphi & \text{iff} & M, s \not\models \varphi \,\text{(that is, not } M, s \models \varphi) \\
M, s \models \varphi \vee \psi & \text{iff} & M, s \models \varphi \text{ or } M, s \models \psi \\
M, s \models \Box\varphi & \text{iff} & V(\varphi) = S \\
M, s \models \Box_a\varphi & \text{iff} & av(s) \subseteq V(\varphi) \\
M, s \models \Box_p\varphi & \text{iff} & pv(s) \subseteq V(\varphi) \\
M, s \models \bigcirc(\psi/\varphi) & \text{iff} & V(\psi) \in ob(V(\varphi)) \\
M, s \models \bigcirc_a\varphi & \text{iff} & V(\varphi) \in ob(av(s)) \text{ and } av(s) \cap V(\neg\varphi) \neq \emptyset \\
M, s \models \bigcirc_p\varphi & \text{iff} & V(\varphi) \in ob(pv(s)) \text{ and } pv(s) \cap V(\neg\varphi) \neq \emptyset
\end{array}
$$

Our evaluation rule for $\bigcirc(\_/\_)$ is a simplified version of the one used by Carmo and Jones. Given the constraints placed on ob, both rules are equivalent (cf. [23, result II-2-2]).

As usual, a CJL formula φ is *valid in a CJL model* $M = \langle S, av, pv, ob, V\rangle$, denoted as $M \models^{CJL} \varphi$, if and only if for all worlds $s \in S$ holds $M, s \models \varphi$. A formula φ is *valid*, denoted $\models^{CJL} \varphi$, if and only if it is valid in every CJL model.

7.1.2 Axiomatization

Carmo and Jones provided an axiomatization [67] for their semantics as follows:

- Characterization of $\Box$:

  (1) $\Box$ is a normal modal operator of type $S5$

- Characterization of O: Reference label

  (2) $\bigcirc(\psi/\varphi) \rightarrow \Diamond(\varphi \wedge \psi)$ $(\bigcirc \rightarrow \Diamond)$

  (3) $\Diamond(\varphi \wedge \psi \wedge \chi) \wedge \bigcirc(\psi/\varphi) \wedge \bigcirc(\chi/\varphi) \rightarrow \bigcirc(\psi \wedge \chi/\varphi)$ $(\bigcirc - C)$

- (4) The principle of strengthening of the antecedent:

  $\Box(\varphi \rightarrow \psi) \wedge \Diamond(\varphi \wedge \chi) \wedge \bigcirc(\chi/\psi) \rightarrow \bigcirc(\chi/\varphi)$ $(\bigcirc - SA)$

- (5) The RE-axiom wrt the antecedent:

  $\Box(\varphi \leftrightarrow \psi) \rightarrow (\bigcirc(\chi/\varphi) \leftrightarrow \bigcirc(\chi/\psi))$ $(\bigcirc - REA)$

- (6) the contextual RE-axiom wrt the consequent:

  $\Box(\chi \to (\varphi \leftrightarrow \psi)) \to (\bigcirc(\varphi/\chi) \leftrightarrow \bigcirc(\psi/\chi))$ $(\bigcirc - CONT - REC)$

  (7) $\bigcirc(\psi/\varphi) \to \Box \bigcirc (\psi/\varphi)$ $(\bigcirc \to \Box\bigcirc)$

  (8) $\bigcirc(\psi/\varphi) \to \bigcirc(\varphi \to \psi/\top)$ $(\bigcirc \to \bigcirc \to)$

- Characterization of $\Box_p$:

  (9) $\Box_p$ is a normal modal operator of type **KT**

- Characterization of $\Box_a$:

  (10) $\Box_a$ is a normal modal operator of type **KD**

- Relationships between $\Box$, $\Box_p$ and $\Box_a$:

  (11) $\Box\varphi \to \Box_p\varphi$ $(\Box \to \Box_p)$ (12) $\Box_p\varphi \to \Box_a\varphi$ $(\Box_p \to \Box_a)$

- Relationships between $\bigcirc_a(\bigcirc_p)$ and $\Box$ $(\Box_p)$

  (13)$\Box_a\varphi \to (\neg \bigcirc_a \varphi \wedge \neg \bigcirc_a \neg\varphi)$ $(\neg\bigcirc_a)$

  $\Box_p\varphi \to (\neg \bigcirc_p \varphi \wedge \neg \bigcirc_p \neg\varphi)$ $(\neg\bigcirc_p)$

  (14) $\Box_a(\varphi \leftrightarrow \psi) \to (\bigcirc_a\varphi \leftrightarrow \bigcirc_a\psi)$ $(\leftrightarrow \bigcirc_a)$

  $\Box_p(\varphi \leftrightarrow \psi) \to (\bigcirc_p\varphi \leftrightarrow \bigcirc_p\psi)$ $(\leftrightarrow \bigcirc_p)$

- Relationships between $\bigcirc$, $\bigcirc_a$ $(\bigcirc_p)$ and $\Box_a$ $(\Box_p)$ - factual detachment axioms:

  (15) $\bigcirc(\psi/\varphi) \wedge \Box_a\varphi \wedge \Diamond_a\psi \wedge \Diamond_a\neg\psi \to \bigcirc_a\psi$ $(\bigcirc_a - FD)$

  $\bigcirc(\psi/\varphi) \wedge \Box_p\varphi \wedge \Diamond_p\psi \wedge \Diamond_p\neg\psi \to \bigcirc_p\psi$ $(\bigcirc_p - FD)$

- (16) Rules to consistently add $\bigcirc(\_/\_)$ formulas:

  $(\bigcirc_a - \Box_a\bigcirc)$- rule: if the propositional symbol q does not occur in any of the formulas $\psi_1, ..., \psi_n, \varphi$
  and $\vdash \psi_1 \wedge ... \wedge \psi_n \to \neg\Box(\bigcirc_a\varphi \to \Box_a q \wedge \bigcirc(\varphi/q))$
  then $\vdash \psi_1 \wedge ... \wedge \psi_n \to \neg\Diamond \bigcirc_a \varphi$ (i.e. $\vdash \psi_1 \wedge ... \wedge \psi_n \to \Box\neg \bigcirc_a \varphi$).

  $(\bigcirc_p - \Box_p\bigcirc)$- rule: if the propositional symbol q does not occure in any of the formulas $\psi_1, ..., \psi_n, \varphi$
  and $\vdash \psi_1 \wedge ... \wedge \psi_n \to \neg\Box(\bigcirc_p\varphi \to \Box_p q \wedge \bigcirc(\varphi/q))$
  then $\vdash \psi_1 \wedge ... \wedge \psi_n \to \neg\Diamond \bigcirc_p \varphi$.

7.2 Modeling CJL as a Fragment of HOL

This section, as the core contribution of this article, presents a shallow semantical embedding of CJL in HOL and proves its soundness and completeness.

7.2.1 Semantical embedding

CJL formulas are identified in our semantical embedding with certain HOL terms (predicates) of type $i \to o$. They can be applied to terms of type i, which are assumed to denote possible worlds. That is, the HOL type i is now identified with a (non-empty) set of worlds. The HOL signature is assumed to contain the constant symbol $av_{i\to\tau}$, $pv_{i\to\tau}$ and $ob_{\tau\to\tau\to o}$. Moreover, for each propositional symbol p^j of CJL, the HOL signature must contain a respective constant symbols p^j_τ. Without loss of generality, we assume that besides those symbols and the primitive logical connectives of HOL, no other constant symbols are given in the signature of HOL.

The mapping $\lfloor\cdot\rfloor$ translates CJL formulas s into HOL terms $\lfloor\varphi\rfloor$ of type τ. The mapping is recursively defined:

$$\begin{array}{lcl}
\lfloor p^j \rfloor & = & p^j_\tau \\
\lfloor \neg\varphi \rfloor & = & \neg_{\tau\to\tau} \lfloor\varphi\rfloor \\
\lfloor \varphi \vee \psi \rfloor & = & \vee_{\tau\to\tau\to\tau} \lfloor\varphi\rfloor \lfloor\psi\rfloor \\
\lfloor \Box\varphi \rfloor & = & \Box_{\tau\to\tau} \lfloor\varphi\rfloor \\
\lfloor \bigcirc(\psi/\varphi) \rfloor & = & \bigcirc_{\tau\to\tau\to\tau} \lfloor\varphi\rfloor \lfloor\psi\rfloor \\
\lfloor \Box_a\varphi \rfloor & = & \Box^a_{\tau\to\tau} \lfloor\varphi\rfloor \\
\lfloor \Box_p\varphi \rfloor & = & \Box^p_{\tau\to\tau} \lfloor\varphi\rfloor \\
\lfloor \bigcirc_a\varphi \rfloor & = & \bigcirc^a_{\tau\to\tau} \lfloor\varphi\rfloor \\
\lfloor \bigcirc_p\varphi \rfloor & = & \bigcirc^p_{\tau\to\tau} \lfloor\varphi\rfloor
\end{array}$$

$\neg_{\tau\to\tau}$, $\vee_{\tau\to\tau\to\tau}$, $\Box_{\tau\to\tau}$, $\bigcirc_{\tau\to\tau\to\tau}$, $\Box^{a}_{\tau\to\tau}$, $\Box^{p}_{\tau\to\tau}$, $\bigcirc^{a}_{\tau\to\tau}$ and $\bigcirc^{p}_{\tau\to\tau}$ thereby abbreviate the following HOL terms:

$$\begin{aligned}
\neg_{\tau\to\tau} &= \lambda A_\tau \lambda X_i \neg(A\,X)\\
\vee_{\tau\to\tau\to\tau} &= \lambda A_\tau \lambda B_\tau \lambda X_i (A\,X \vee B\,X)\\
\Box_{\tau\to\tau} &= \lambda A_\tau \lambda X_i \forall Y_i (A\,Y)\\
\bigcirc_{\tau\to\tau\to\tau} &= \lambda A_\tau \lambda B_\tau \lambda X_i (ob\,A\,B)\\
\Box^{a}_{\tau\to\tau} &= \lambda A_\tau \lambda X_i \forall Y_i (\neg(av\,X\,Y) \vee A\,Y)\\
\Box^{p}_{\tau\to\tau} &= \lambda A_\tau \lambda X_i \forall Y_i (\neg(pv\,X\,Y) \vee A\,Y)\\
\bigcirc^{a}_{\tau\to\tau} &= \lambda A_\tau \lambda X_i ((ob\,(av\,X)\,A) \wedge \exists Y_i (av\,X\,Y \wedge \neg(A\,Y)))\\
\bigcirc^{p}_{\tau\to\tau} &= \lambda A_\tau \lambda X_i ((ob\,(pv\,X)\,A) \wedge \exists Y_i (pv\,X\,Y \wedge \neg(A\,Y)))
\end{aligned}$$

Analyzing the truth of a translated formula $\lfloor\varphi\rfloor$ in a world represented by term w_i corresponds to evaluating the application $(\lfloor\varphi\rfloor\, w_i)$. In line with previous work [40], we define $\text{vld}_{\tau\to o} = \lambda A_\tau \forall S_i (A\,S)$. With this definition, validity of a CJL formula s in CJL corresponds to the validity of formula $(\text{vld}\,\lfloor\varphi\rfloor)$ in HOL, and vice versa.

7.2.2 Soundness and completeness

To prove the soundness and completeness, that is, faithfulness, of the above embedding, a mapping from CJL models into Henkin models is employed.

Definition 7.1 (Henkin model H^M for CJL model M)**.** For any CJL model $M = \langle S, av, pv, ob, V\rangle$, we define corresponding Henkin models H^M. Thus, let a CJL model $M = \langle S, av, pv, ob, V\rangle$ be given. Moreover, assume that $p^1, ..., p^m \in P$, for $m \geq 1$, are the only propositional symbols of CJL. Remember that our embedding requires the corresponding signature of HOL to provide constant symbols p^j_τ such that $\lfloor p^j\rfloor = p^j_\tau$ for $j = 1, \ldots, m$.

A Henkin model $H^M = \langle \{D_\alpha\}_{\alpha\in T}, I\rangle$ for M is now defined as follows: D_i is chosen as the set of possible worlds S; all other sets $D_{\alpha\to\beta}$ are chosen as (not necessarily full) sets of functions from D_α to D_β. For all $D_{\alpha\to\beta}$ the rule that every term $t_{\alpha\to\beta}$ must have a denotation in $D_{\alpha\to\beta}$ must be obeyed (Denotatpflicht). In particular, it is required that D_τ, $D_{i\to\tau}$ and $D_{\tau\to\tau\to o}$ contain the elements Ip^j_τ, $Iav_{i\to\tau}$, $Ipv_{i\to\tau}$ and $Iob_{\tau\to\tau\to o}$. The interpretation function I of H^M is defined as follows:

1. For $j = 1, \ldots, m$, $Ip^j_\tau \in D_\tau$ is chosen such that $Ip^j_\tau(s) = T$ iff $s \in V(p^j)$ in M.

2. $Iav_{i\to\tau} \in D_{i\to\tau}$ is chosen such that $Iav_{i\to\tau}(s,u) = T$ iff $u \in av(s)$ in M.
3. $Ipv_{i\to\tau} \in D_{i\to\tau}$ is chosen such that $Ipv_{i\to\tau}(s,u) = T$ iff $u \in pv(s)$ in M.
4. $Iob_{\tau\to\tau\to o} \in D_{\tau\to\tau\to o}$ is chosen such that $Iob_{\tau\to\tau\to o}(\bar{X},\bar{Y}) = T$ iff $\bar{Y} \in ob(\bar{X})$ in M.
5. For the logical connectives $\neg$, $\vee$, Π and $=$ of HOL the interpretation function I is defined as usual (see the previous section).

Since we assume that there are no other symbols (besides the p^i_τ, av, pv, ob and $\neg$, $\vee$, Π, and $=$) in the signature of HOL, I is a total function. Moreover, the above construction guarantees that H^M is a Henkin model: $\langle D, I\rangle$ is a frame, and the choice of I in combination with the Denotatpflicht ensures that for arbitrary assignments g, $\|.\|^{H^M,g}$ is an total evaluation function.

Lemma 7.2. *Let H^M be a Henkin model for a CJL model M. In H^M we have for all $s \in D_i$ and all $\bar{X},\bar{Y},\bar{Z} \in D_\tau$ (cf. the conditions CJL models as stated on page 3):*[1]

(av) $Iav_{i\to\tau}(s) \neq \emptyset$.

(pv1) $Iav_{i\to\tau}(s) \subseteq Ipv_{i\to\tau}(s)$.

(pv2) $s \in Ipv_{i\to\tau}(s)$.

(ob1) $\emptyset \notin Iob_{\tau\to\tau\to o}(\bar{X})$.

(ob2) *If* $\bar{Y} \cap \bar{X} = \bar{Z} \cap \bar{X}$, *then* ($\bar{Y} \in Iob_{\tau\to\tau\to o}(\bar{X})$ *iff* $\bar{Z} \in Iob_{\tau\to\tau\to o}(\bar{X})$).

(ob3) *Let* $\bar{\beta} \subseteq Iob_{\tau\to\tau\to o}(\bar{X})$ *and* $\bar{\beta} \neq \emptyset$.
If $(\cap\bar{\beta}) \cap \bar{X} \neq \emptyset$, *where* $\cap\bar{\beta} = \{s \in S \mid$ *for all* $\bar{Z} \in \bar{\beta}$ *we have* $s \in \bar{Z}\}$,
then $(\cap\bar{\beta}) \in Iob_{\tau\to\tau\to o}(\bar{X})$.

(ob4) *If* $\bar{Y} \subseteq \bar{X}$ *and* $\bar{Y} \in Iob_{\tau\to\tau\to o}(\bar{X})$ *and* $\bar{X} \subseteq \bar{Z}$,
then $(\bar{Z} \setminus \bar{X}) \cup \bar{Y} \in Iob_{\tau\to\tau\to o}(\bar{Z})$.

(ob5) *If* $\bar{Y} \subseteq \bar{X}$ *and* $\bar{Z} \in Iob_{\tau\to\tau\to o}(\bar{X})$ *and* $\bar{Y} \cap \bar{Z} \neq \emptyset$,
then $\bar{Z} \in Iob_{\tau\to\tau\to o}(\bar{Y})$.

Proof. Each statement follows by construction of H^M for M.
(av): By definition of av for $s \in S$ in M, $av(s) \neq \emptyset$; hence, there is $u \in S$ such that $u \in av(s)$. By definition of H^M, $Iav_{i\to\tau}(s,u) = T$, so $u \in Iav_{i\to\tau}(s)$ and hence $Iav_{i\to\tau}(s) \neq \emptyset$ in H^M.
(pv1): By definition of av and pv for $s \in S$ in M, $av(s) \subseteq pv(s)$; hence, for every $u \in av(s)$ we have $u \in pv(s)$. In H^M this means, if $Iav_{i\to\tau}(s,u) = T$, then $Ipv_{i\to\tau}(s,u) = T$. So, $Iav_{i\to\tau}(s) \subseteq Ipv_{i\to\tau}(s)$ in H^M.

[1] In the proof we implicitly employ currying and uncurrying, and we associate sets with their characteristic functions. This analogously applies to the remainder of this article.

(pv2): This case is similar to (av).

(ob1): By definition of ob, we have $\emptyset \notin ob(\bar{X})$; hence, in H^M, $Iob_{\tau\to\tau\to o}(\bar{X}, \emptyset) = F$, that is $\emptyset \notin Iob_{\tau\to\tau\to o}(\bar{X})$.

(ob2): Suppose $\bar{Y} \cap \bar{X} = \bar{Z} \cap \bar{X}$. In M we have $\bar{Y} \in ob(\bar{X})$ iff $\bar{Z} \in ob(\bar{X})$. By definition of H^M we have $Iob_{\tau\to\tau\to o}(\bar{X}, \bar{Y}) = T$ iff $Iob_{\tau\to\tau\to o}(\bar{X}, \bar{Z}) = T$. Hence, $\bar{Y} \in Iob_{\tau\to\tau\to o}(\bar{X})$ iff $\bar{Z} \in Iob_{\tau\to\tau\to o}(\bar{X})$ in H^M.

(ob3): Suppose $\bar{\beta} \subseteq Iob_{\tau\to\tau\to o}(\bar{X})$ and $\bar{\beta} \neq \emptyset$. If $(\cap\bar{\beta}) \cap \bar{X} \neq \emptyset$, by definition of ob in M we have $(\cap\bar{\beta}) \in ob(\bar{X})$. Hence, in H^M, $Iob_{\tau\to\tau\to o}(\bar{X}, (\cap\bar{\beta})) = T$ and then $(\cap\bar{\beta}) \in Iob_{\tau\to\tau\to o}(\bar{X})$.

(ob4) and (ob5) are similar to (ob2). □ □

Lemma 7.3. *Let $H^M = \langle \{D_\alpha\}_{\alpha\in T}, I\rangle$ be a Henkin model for a CJL model M. We have $H^M \models^{HOL} \Sigma$ for all $\Sigma \in \{AV, PV1, PV2, OB1, ..., OB5\}$, where*

$$\begin{array}{lll}
AV & \textit{is} & \forall W_i \exists V_i (av_{i\to\tau} W_i V_i) \\
PV1 & \textit{is} & \forall W_i \forall V_i (av_{i\to\tau} W_i V_i \to pv_{i\to\tau} W_i V_i) \\
PV2 & \textit{is} & \forall W_i (pv_{i\to\tau} W_i W_i) \\
OB1 & \textit{is} & \forall X_\tau \neg ob_{\tau\to\tau\to o} X_\tau (\lambda X_\tau \bot) \\
OB2 & \textit{is} & \forall X_\tau Y_\tau Z_\tau (\ (\forall W_i ((Y_\tau W_i \wedge X_\tau W_i) \longleftrightarrow (Z_\tau W_i \wedge X_\tau W_i))) \\
& & \qquad\qquad \to (ob_{\tau\to\tau\to o} X_\tau Y_\tau \longleftrightarrow ob_{\tau\to\tau\to o} X_\tau Z_\tau)) \\
OB3 & \textit{is} & \forall \beta_{\tau\to\tau\to o} \forall X_\tau \\
& & (\ ((\forall Z_\tau (\beta_{\tau\to\tau\to o} Z_\tau \to ob_{\tau\to\tau\to o} X_\tau Z_\tau)) \wedge \exists Z_\tau (\beta_{\tau\to\tau\to o} Z_\tau)) \\
& & \quad \to (\ (\exists Y_i (((\lambda W_i \forall Z_\tau (\beta_{\tau\to\tau\to o} Z_\tau \to Z_\tau W_i))\, Y_i) \wedge X_\tau Y_i)) \\
& & \qquad \to ob_{\tau\to\tau\to o} X_\tau (\lambda W_i \forall Z_\tau (\beta_{\tau\to\tau\to o} Z_\tau \to Z_\tau W_i)))) \\
OB4 & \textit{is} & \forall X_\tau Y_\tau Z_\tau \\
& & (\ (\forall W_i (Y_\tau W_i \to X_\tau W_i) \wedge ob_{\tau\to\tau\to o} X_\tau Y_\tau \wedge \forall X_\tau (X_\tau W_i \to Z_\tau W_i)) \\
& & \quad \to ob_{\tau\to\tau\to o} Z_\tau (\lambda W_i ((Z_\tau W_i \wedge \neg X_\tau W_i) \vee Y_\tau W_i))) \\
OB5 & \textit{is} & \forall X_\tau Y_\tau Z_\tau \\
& & (\ (\forall W_i (Y_\tau W_i \to X_\tau W_i) \wedge ob_{\tau\to\tau\to o} X_\tau Z_\tau \wedge \exists W_i (Y_\tau W_i \wedge Z_\tau W_i)) \\
& & \quad \to ob_{\tau\to\tau\to o} Y_\tau Z_\tau)
\end{array}$$

Proof. See Appendix A.2. □

Lemma 7.4. *Let H^M be a Henkin model for a CJL model M. For all CJL formulas δ, arbitrary variable assignments g and worlds s it holds:*

$$M, s \models \delta \textit{ if and only if } \|\lfloor\delta\rfloor\, S_i\|^{H^M, g[s/S_i]} = T$$

Proof. See Appendix A.2. □

Lemma 7.5. *For every Henkin model $H = \langle \{D_\alpha\}_{\alpha \in T}, I \rangle$ such that $H \models^{HOL} \Sigma$ for all $\Sigma \in \{AV, PV1, PV2, OB1, \ldots, OB5\}$, there exists a corresponding CJL model M. Corresponding means that for all CJL formulas δ and for all assignment g and worlds s, $\| \lfloor \delta \rfloor S \|^{H, g[s/S_i]} = T$ if and only if $M, s \vDash \delta$.*

Proof. Suppose that $H = \langle \{D_\alpha\}_{\alpha \in T}, I \rangle$ is a Henkin model such that $H \models^{\mathrm{HOL}} \Sigma$ for all $\Sigma \in \{$AV, PV1, PV2, OB1,..,OB5$\}$. Without loss of generality, we can assume that the domains of H are denumerable [110]. We construct the corresponding CJL model M as follows:

- $S = D_i$.
- $u \in av(s)$ for $s, u \in S$ iff $Iav_{i \to \tau}(s, u) = T$.
- $u \in pv(s)$ for $s, u \in S$ iff $Ipv_{i \to \tau}(s, u) = T$.
- $\bar{Y} \in ob(\bar{X})$ for $\bar{X}, \bar{Y} \in D_i \longrightarrow D_o$ iff $Iob_{\tau \to \tau \to o}(\bar{X}, \bar{Y}) = T$.
- $s \in V(p^j)$ iff $Ip^j_\tau(s) = T$ for all p^j.

Since $H \models^{\mathrm{HOL}} \Sigma$ for all $\Sigma \in \{$AV, PV1, PV2, OB1, .., OB5$\}$, it is straightforward (but tedious) to verify that av, pv and ob satisfy the conditions as required for a CJL model.

Moreover, the above construction ensures that H is a Henkin model H^M for CJL model M. Hence, Lemma 7.4 applies. This ensures that for all CJL formulas δ, for all assignment g and all worlds s we have $\| \lfloor \delta \rfloor S \|^{H, g[s/S_i]} = T$ if and only if $M, s \vDash \delta$. □

Theorem 7.6 (Soundness and Completeness of the Embedding)**.**

$$\models^{CJL} \varphi \text{ if and only if } \{AV,\ PV1,\ PV2,\ OB1,..,OB5\} \models^{HOL} vld \lfloor \varphi \rfloor$$

Proof. (Soundness, $\leftarrow$) The proof is by contraposition. Assume $\not\models^{CJL} \varphi$, that is, there is a CJL model $M = \langle S, av, pv, ob, V \rangle$, and world $s \in S$, such that $M, s \not\vDash \varphi$. Now let H^M be a Henkin model for CJL model M. By Lemma 7.4, for an arbitrary assignment g, it holds that $\| \lfloor \varphi \rfloor S \|^{H^M, g[s/S_i]} = F$. Thus, by definition of $\|.\|$, it holds that $\| \forall S_i (\lfloor \varphi \rfloor S_i) \|^{H^M, g} = \| \mathrm{vld} \lfloor \varphi \rfloor \|^{H^M, g} = F$. Hence, $H^M \not\models^{\mathrm{HOL}} \mathrm{vld} \lfloor \varphi \rfloor$. Furthermore, $H^M \models^{\mathrm{HOL}} \Sigma$ for all $\Sigma \in \{$AV, PV1, PV2, OB1,...,OB5$\}$ by Lemma 7.3. Thus, $\{$AV, PV1, PV2, OB1,..,OB5$\} \not\models^{\mathrm{HOL}} \mathrm{vld} \lfloor \varphi \rfloor$.

(Completeness, $\rightarrow$) The proof is again by contraposition. Assume $\{$AV, PV1, PV2, OB1,..,OB5$\} \not\models^{\mathrm{HOL}} \mathrm{vld} \lfloor \varphi \rfloor$, that is, there is a Henkin model $H =$

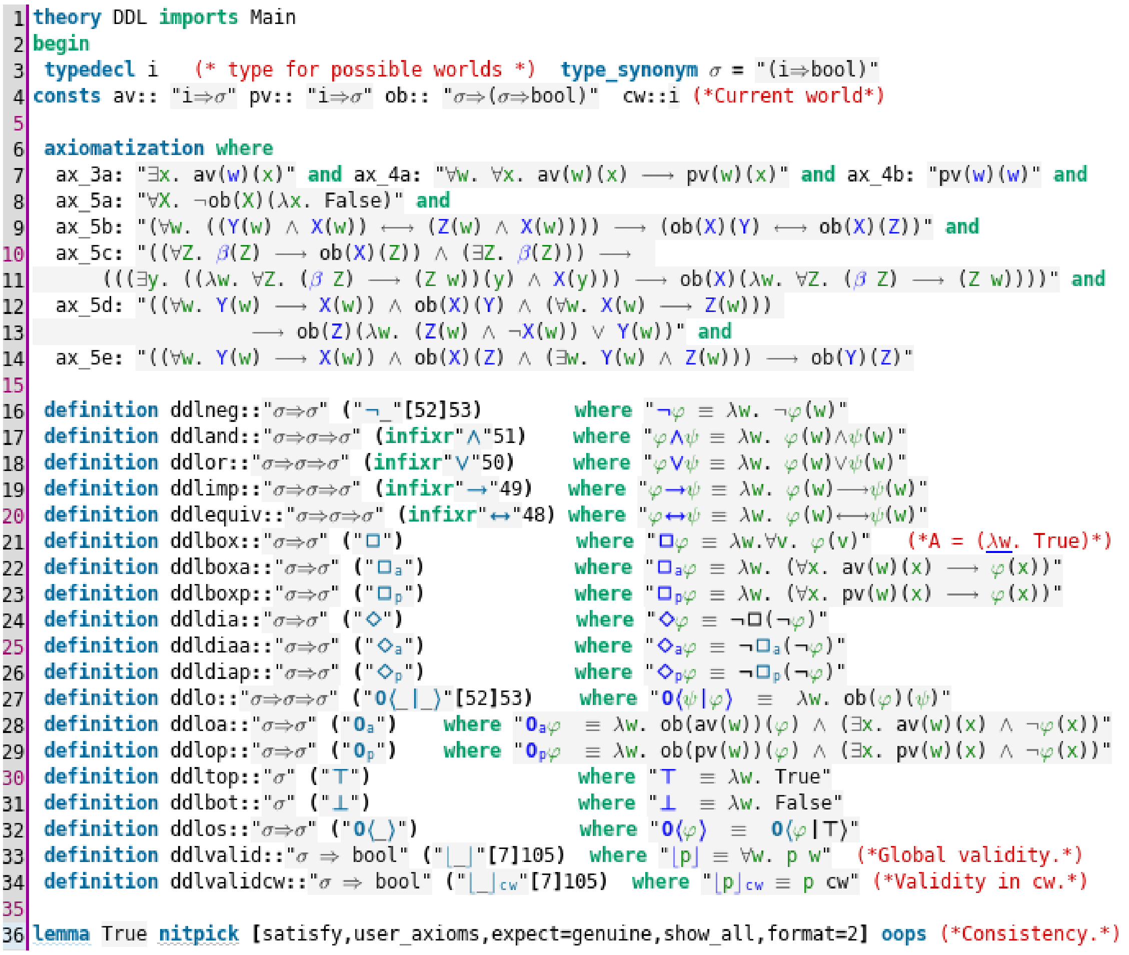
$\langle \{D_\alpha\}_{\alpha \in T}, I\rangle$ such that $H \models^{\mathrm{HOL}} \Sigma$ for all $\Sigma \in \{\mathrm{AV}, \mathrm{PV1}, \mathrm{PV2}, \mathrm{OB1},..,\mathrm{OB5}\}$, but $\|\mathrm{vld}\, \lfloor\varphi\rfloor\|^{H,g} = F$ for some assignment g . By Lemma 7.5, there is a CJL model M such that $M \not\models^{CJL} \varphi$. Hence, $\not\models^{CJL} \varphi$.

□

The theorem characterizes CJL as a natural fragment of HOL.

7.3 Implementation in Isabelle/HOL

The semantical embedding as devised in Section 7.2 has been implemented in the higher-order proof assistant Isabelle/HOL [156]. Figure 7.1 displays the respective encoding.

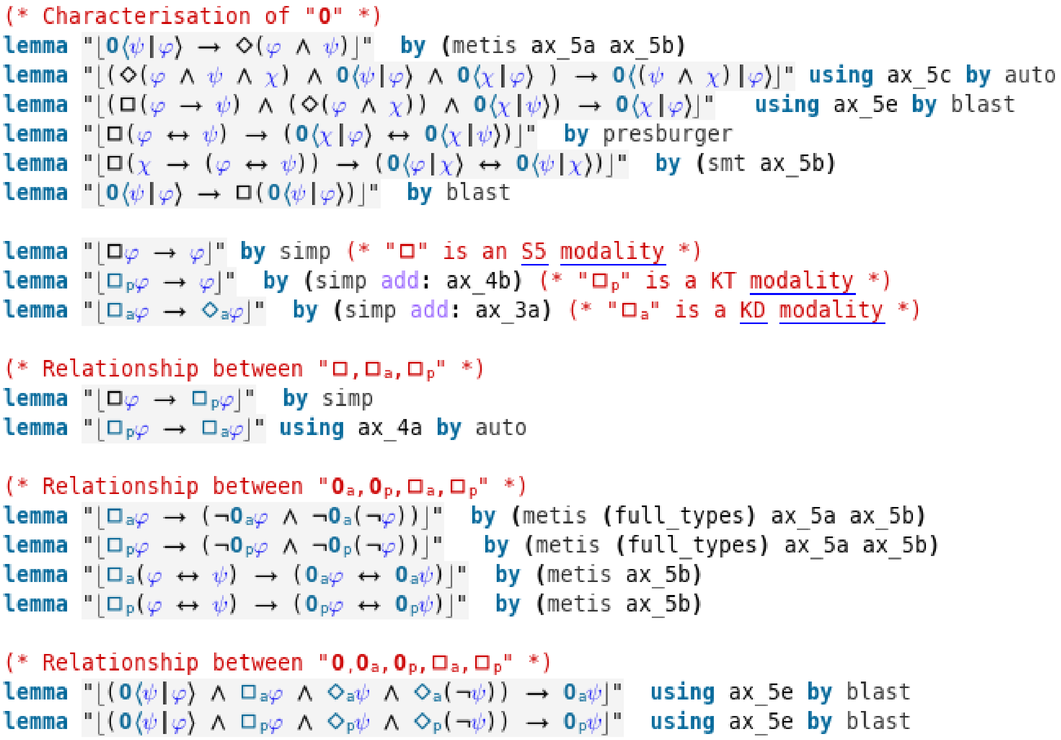


```
theory DDL imports Main
begin
 typedecl i   (* type for possible worlds *)  type_synonym σ = "(i⇒bool)"
consts av:: "i⇒σ" pv:: "i⇒σ" ob:: "σ⇒(σ⇒bool)"  cw::i (*Current world*)

 axiomatization where
  ax_3a: "∃x. av(w)(x)" and ax_4a: "∀w. ∀x. av(w)(x) ⟶ pv(w)(x)" and ax_4b: "pv(w)(w)" and
  ax_5a: "∀X. ¬ob(X)(λx. False)" and
  ax_5b: "(∀w. ((Y(w) ∧ X(w)) ⟷ (Z(w) ∧ X(w)))) ⟶ (ob(X)(Y) ⟷ ob(X)(Z))" and
  ax_5c: "((∀Z. β(Z) ⟶ ob(X)(Z)) ∧ (∃Z. β(Z))) ⟶
      (((∃y. ((λw. ∀Z. (β Z) ⟶ (Z w))(y) ∧ X(y))) ⟶ ob(X)(λw. ∀Z. (β Z) ⟶ (Z w))))" and
  ax_5d: "((∀w. Y(w) ⟶ X(w)) ∧ ob(X)(Y) ∧ (∀w. X(w) ⟶ Z(w)))
                 ⟶ ob(Z)(λw. (Z(w) ∧ ¬X(w)) ∨ Y(w))" and
  ax_5e: "((∀w. Y(w) ⟶ X(w)) ∧ ob(X)(Z) ∧ (∃w. Y(w) ∧ Z(w))) ⟶ ob(Y)(Z)"

 definition ddlneg::"σ⇒σ" ("¬_"[52]53)        where "¬φ ≡ λw. ¬φ(w)"
 definition ddland::"σ⇒σ⇒σ" (infixr"∧"51)     where "φ∧ψ ≡ λw. φ(w)∧ψ(w)"
 definition ddlor::"σ⇒σ⇒σ" (infixr"∨"50)      where "φ∨ψ ≡ λw. φ(w)∨ψ(w)"
 definition ddlimp::"σ⇒σ⇒σ" (infixr"→"49)    where "φ→ψ ≡ λw. φ(w)⟶ψ(w)"
 definition ddlequiv::"σ⇒σ⇒σ" (infixr"↔"48) where "φ↔ψ ≡ λw. φ(w)⟷ψ(w)"
 definition ddlbox::"σ⇒σ" ("□")               where "□φ ≡ λw.∀v. φ(v)"    (*A = (λw. True)*)
 definition ddlboxa::"σ⇒σ" ("□ₐ")             where "□ₐφ ≡ λw. (∀x. av(w)(x) ⟶ φ(x))"
 definition ddlboxp::"σ⇒σ" ("□ₚ")             where "□ₚφ ≡ λw. (∀x. pv(w)(x) ⟶ φ(x))"
 definition ddldia::"σ⇒σ" ("◇")               where "◇φ ≡ ¬□(¬φ)"
 definition ddldiaa::"σ⇒σ" ("◇ₐ")             where "◇ₐφ ≡ ¬□ₐ(¬φ)"
 definition ddldiap::"σ⇒σ" ("◇ₚ")             where "◇ₚφ ≡ ¬□ₚ(¬φ)"
 definition ddlo::"σ⇒σ⇒σ" ("O⟨_|_⟩"[52]53)    where "O⟨ψ|φ⟩  ≡  λw. ob(φ)(ψ)"
 definition ddloa::"σ⇒σ" ("Oₐ")     where "Oₐφ  ≡ λw. ob(av(w))(φ) ∧ (∃x. av(w)(x) ∧ ¬φ(x))"
 definition ddlop::"σ⇒σ" ("Oₚ")     where "Oₚφ  ≡ λw. ob(pv(w))(φ) ∧ (∃x. pv(w)(x) ∧ ¬φ(x))"
 definition ddltop::"σ" ("⊤")                 where "⊤  ≡ λw. True"
 definition ddlbot::"σ" ("⊥")                 where "⊥  ≡ λw. False"
 definition ddlos::"σ⇒σ" ("O⟨_⟩")             where "O⟨φ⟩  ≡  O⟨φ|⊤⟩"
 definition ddlvalid::"σ ⇒ bool" ("⌊_⌋"[7]105)  where "⌊p⌋ ≡ ∀w. p w"  (*Global validity.*)
 definition ddlvalidcw::"σ ⇒ bool" ("⌊_⌋cw"[7]105)  where "⌊p⌋cw ≡ p cw" (*Validity in cw.*)

lemma True nitpick [satisfy,user_axioms,expect=genuine,show_all,format=2] oops (*Consistency.*)
```

Figure 7.1: Shallow semantical embedding of CJL in Isabelle/HOL

- Line 3: the primitive type i for possible words and the derived type σ for formulas is introduced.
- Line 4: the constants av, pv, ob and cw are introduced. av and pv are the HOL counterparts of the accessibility relations used to define $\Box_a$, $\Box_p$. ob is used to define dyadic obligation operator. the constant cw encodes the current world.
- Lines 6–14: the axioms for the accessibility relation av, pv and ob are postulated.
- Lines 16–20: the Boolean logical connectives are defined.
- Lines 21–23: the three necessity operators $\Box$, $\Box_a$, and $\Box_p$ are introduced.
- Lines 24–26: the three possibility operators $\Diamond$, $\Diamond_a$, and $\Diamond_p$ are introduced.
- Line 27: the dyadic obligation operator is defined.
- Lines 28 and 29: the actual and primary obligation operators $\bigcirc_a$ and $\bigcirc_p$ are introduced.
- Line 32: the monadic obligation operator is defined.
- Lines 33–34: the notions of global validity (i.e, truth in all worlds) and local validity (truth at the actual world) are introduced.
- Line 36: the model finder Nitpick [44] confirms the consistency of the logical system.

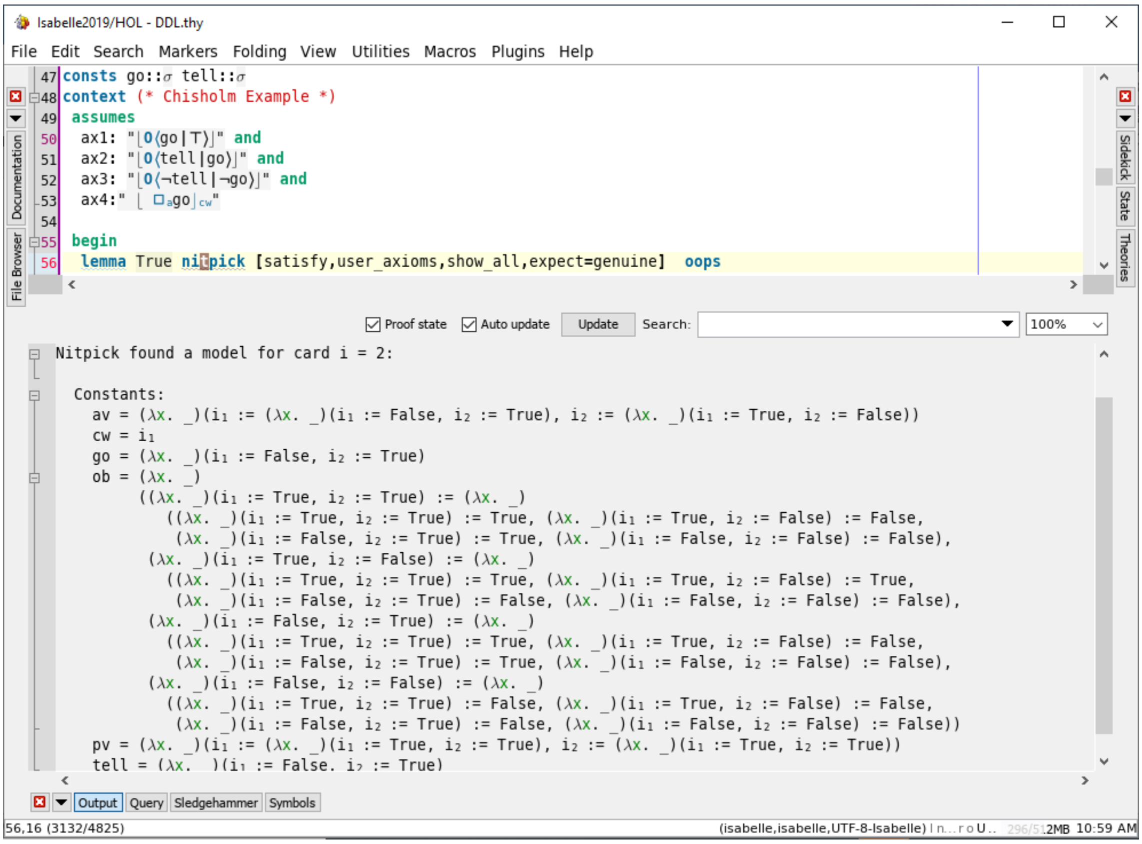
In Figure 7.2 Isabelle/HOL confirms the validity of some axioms by using the access to automated theorem provers.



```
(* Characterisation of "O" *)
lemma "⌊O⟨ψ|φ⟩ → ◇(φ ∧ ψ)⌋"  by (metis ax_5a ax_5b)
lemma "⌊(◇(φ ∧ ψ ∧ χ) ∧ O⟨ψ|φ⟩ ∧ O⟨χ|φ⟩ ) → O⟨(ψ ∧ χ)|φ⟩⌋" using ax_5c by auto
lemma "⌊(□(φ → ψ) ∧ (◇(φ ∧ χ)) ∧ O⟨χ|ψ⟩) → O⟨χ|φ⟩⌋"   using ax_5e by blast
lemma "⌊□(φ ↔ ψ) → (O⟨χ|φ⟩ ↔ O⟨χ|ψ⟩)⌋"  by presburger
lemma "⌊□(χ → (φ ↔ ψ)) → (O⟨φ|χ⟩ ↔ O⟨ψ|χ⟩)⌋"  by (smt ax_5b)
lemma "⌊O⟨ψ|φ⟩ → □(O⟨ψ|φ⟩)⌋"  by blast

lemma "⌊□φ → φ⌋" by simp (* "□" is an S5 modality *)
lemma "⌊□ₚφ → φ⌋"  by (simp add: ax_4b) (* "□ₚ" is a KT modality *)
lemma "⌊□ₐφ → ◇ₐφ⌋"  by (simp add: ax_3a) (* "□ₐ" is a KD modality *)

(* Relationship between "□,□ₐ,□ₚ" *)
lemma "⌊□φ → □ₚφ⌋"  by simp
lemma "⌊□ₚφ → □ₐφ⌋" using ax_4a by auto

(* Relationship between "Oₐ,Oₚ,□ₐ,□ₚ" *)
lemma "⌊□ₐφ → (¬Oₐφ ∧ ¬Oₐ(¬φ))⌋"  by (metis (full_types) ax_5a ax_5b)
lemma "⌊□ₚφ → (¬Oₚφ ∧ ¬Oₚ(¬φ))⌋"   by (metis (full_types) ax_5a ax_5b)
lemma "⌊□ₐ(φ ↔ ψ) → (Oₐφ ↔ Oₐψ)⌋"  by (metis ax_5b)
lemma "⌊□ₚ(φ ↔ ψ) → (Oₚφ ↔ Oₚψ)⌋"  by (metis ax_5b)

(* Relationship between "O,Oₐ,Oₚ,□ₐ,□ₚ" *)
lemma "⌊(O⟨ψ|φ⟩ ∧ □ₐφ ∧ ◇ₐψ ∧ ◇ₐ(¬ψ)) → Oₐψ⌋"  using ax_5e by blast
lemma "⌊(O⟨ψ|φ⟩ ∧ □ₚφ ∧ ◇ₚψ ∧ ◇ₚ(¬ψ)) → Oₚψ⌋"  using ax_5e by blast
```

Figure 7.2: Some validities in CJL

Figure 7.3 applies this encoding to Chisholm's scenario (cf. [72]). Chisholm's statements can be given a consistent formalisation in CJL, see Fig. 7.3. This is confirmed by the model finder Nitpick [44] integrated with Isabelle/HOL. Nitpick computes an intuitive, small model for the scenario consisting of two possible worlds i_1 and i_2.


Isabelle2019/HOL - DDL.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
Documentation
File Browser
Sidekick
State
Theories
☑ Proof state ☑ Auto update
Update
Search:
100%
Output
Query
Sledgehammer
Symbols

```
consts go::σ tell::σ
context (* Chisholm Example *)
 assumes
  ax1: "⌊O⟨go|⊤⟩⌋" and
  ax2: "⌊O⟨tell|go⟩⌋" and
  ax3: "⌊O⟨¬tell|¬go⟩⌋" and
  ax4:" ⌊ □ₐgo⌋cw"

 begin
  lemma True nitpick [satisfy,user_axioms,show_all,expect=genuine]  oops
```

```
Nitpick found a model for card i = 2:

  Constants:
    av = (λx. _)(i₁ := (λx. _)(i₁ := False, i₂ := True), i₂ := (λx. _)(i₁ := True, i₂ := False))
    cw = i₁
    go = (λx. _)(i₁ := False, i₂ := True)
    ob = (λx. _)
         ((λx. _)(i₁ := True, i₂ := True) := (λx. _)
            ((λx. _)(i₁ := True, i₂ := True) := True, (λx. _)(i₁ := True, i₂ := False) := False,
             (λx. _)(i₁ := False, i₂ := True) := True, (λx. _)(i₁ := False, i₂ := False) := False),
          (λx. _)(i₁ := True, i₂ := False) := (λx. _)
            ((λx. _)(i₁ := True, i₂ := True) := True, (λx. _)(i₁ := True, i₂ := False) := True,
             (λx. _)(i₁ := False, i₂ := True) := False, (λx. _)(i₁ := False, i₂ := False) := False),
          (λx. _)(i₁ := False, i₂ := True) := (λx. _)
            ((λx. _)(i₁ := True, i₂ := True) := True, (λx. _)(i₁ := True, i₂ := False) := False,
             (λx. _)(i₁ := False, i₂ := True) := True, (λx. _)(i₁ := False, i₂ := False) := False),
          (λx. _)(i₁ := False, i₂ := False) := (λx. _)
            ((λx. _)(i₁ := True, i₂ := True) := False, (λx. _)(i₁ := True, i₂ := False) := False,
             (λx. _)(i₁ := False, i₂ := True) := False, (λx. _)(i₁ := False, i₂ := False) := False))
    pv = (λx. _)(i₁ := (λx. _)(i₁ := True, i₂ := True), i₂ := (λx. _)(i₁ := True, i₂ := True))
    tell = (λx. _)(i₁ := False, i₂ := True)
```

```
56,16 (3132/4825)        (isabelle,isabelle,UTF-8-Isabelle) I n...r o U..  296/512MB 10:59 AM
```

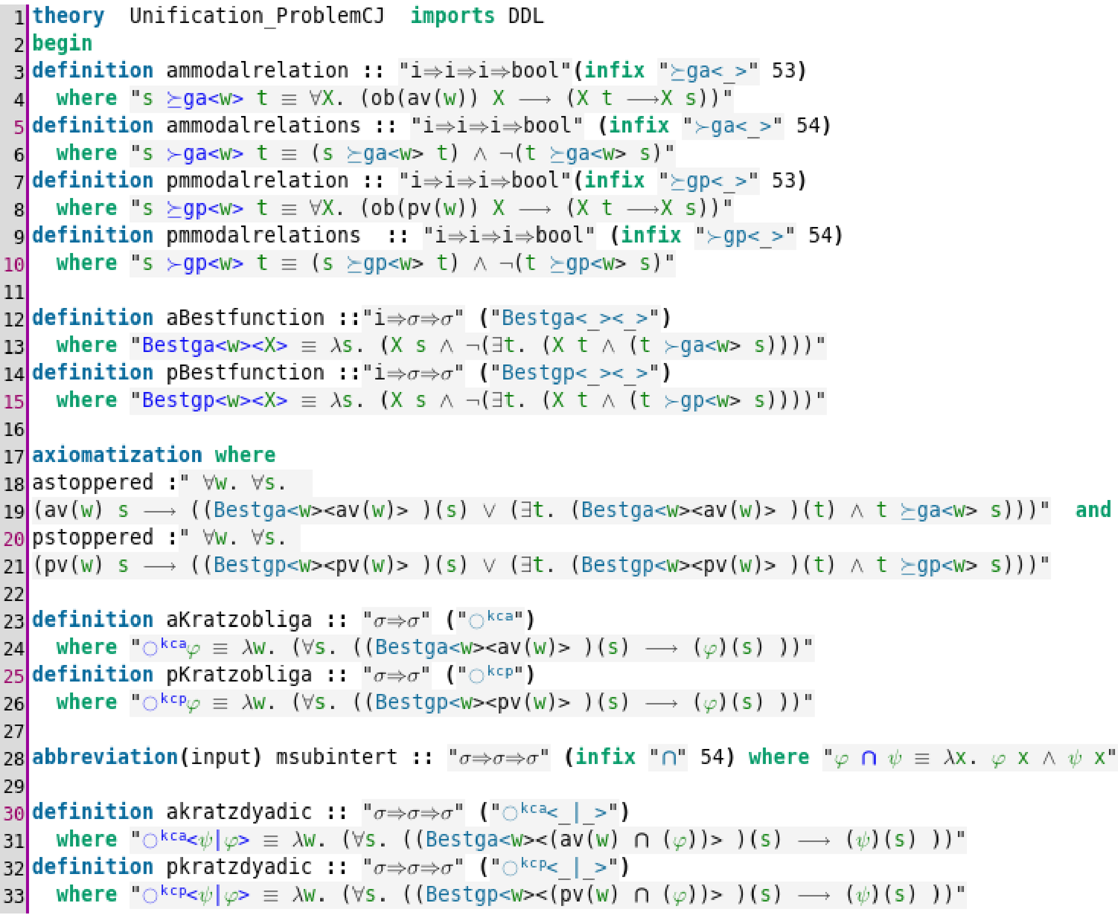
FIGURE 7.3: The Chisholm's scenario encoded in CJL

Function ob is interpreted in this model as: $ob(\{i_1, i_2\}) = \{\{i_1, i_2\}, \{i_2\}\}$, $ob(\{i_1\}) = \{\{i_1, i_2\}, \{i_1\}\}$, $ob(\{i_2\}) = \{\{i_1, i_2\}, \{i_2\}\}$ and $ob(\emptyset) = \emptyset$. The designated current world in the given model is i_1, in which Jones doesn't go to assist his neighbors and doesn't tell them that he is coming. In the other possible world i_2, Jones is going to assist them and he also tells them that he his coming. That is, $V(go) = V(tell) = \{i_2\}$. Also, we have $\{i_2\} \in ob(\{i_1, i_2\})$. So, $i_1 \models \bigcirc go$ by the evaluation rule for $\bigcirc$. Similarly, $\{i_1\} \in ob(\{i_1\})$ implies $i_1 \models \bigcirc(tell/go)$, and $\{i_2\} \in ob(\{i_2\})$ implies $i_1 \models \bigcirc(\neg tell/\neg go)$.

7.3.1 Comparing CJL system and Kratzer's system KD

In this section we study and compare Carmo and Jones deontic system with Kratzerain framework.

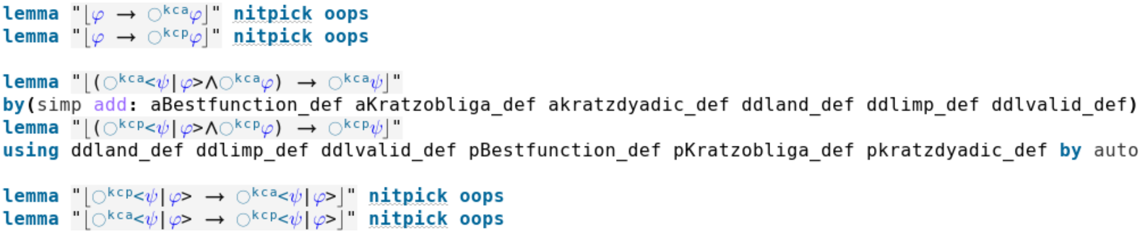
First scenario: In the first scenario, we consider actual and potential accessibility relations as two different modal bases. Moreover, we define two different ordering sources by means of application dyadic operation *ob* on the actual versions of current world $ob(av(w))$ (cf. Fig. 7.4, lines 3–6) or potential versions of current world $ob(pv(w))$ (cf. Fig. 7.4, lines 7–10). We have two kinds of monadic and dyadic deontic operators that could play the same role as actual and potential obligations in the initial CJL system.



```
 1 theory  Unification_ProblemCJ  imports DDL
 2 begin
 3 definition ammodalrelation :: "i⇒i⇒i⇒bool"(infix "⪰ga<_>" 53)
 4   where "s ⪰ga<w> t ≡ ∀X. (ob(av(w)) X ⟶ (X t ⟶X s))"
 5 definition ammodalrelations :: "i⇒i⇒i⇒bool" (infix "≻ga<_>" 54)
 6   where "s ≻ga<w> t ≡ (s ⪰ga<w> t) ∧ ¬(t ⪰ga<w> s)"
 7 definition pmmodalrelation :: "i⇒i⇒i⇒bool"(infix "⪰gp<_>" 53)
 8   where "s ⪰gp<w> t ≡ ∀X. (ob(pv(w)) X ⟶ (X t ⟶X s))"
 9 definition pmmodalrelations  :: "i⇒i⇒i⇒bool" (infix "≻gp<_>" 54)
10   where "s ≻gp<w> t ≡ (s ⪰gp<w> t) ∧ ¬(t ⪰gp<w> s)"
11
12 definition aBestfunction ::"i⇒σ⇒σ" ("Bestga<_><_>")
13   where "Bestga<w><X> ≡ λs. (X s ∧ ¬(∃t. (X t ∧ (t ≻ga<w> s))))"
14 definition pBestfunction ::"i⇒σ⇒σ" ("Bestgp<_><_>")
15   where "Bestgp<w><X> ≡ λs. (X s ∧ ¬(∃t. (X t ∧ (t ≻gp<w> s))))"
16
17 axiomatization where
18 astoppered :" ∀w. ∀s.
19 (av(w) s ⟶ ((Bestga<w><av(w)> )(s) ∨ (∃t. (Bestga<w><av(w)> )(t) ∧ t ⪰ga<w> s)))"  and
20 pstoppered :" ∀w. ∀s.
21 (pv(w) s ⟶ ((Bestgp<w><pv(w)> )(s) ∨ (∃t. (Bestgp<w><pv(w)> )(t) ∧ t ⪰gp<w> s)))"
22
23 definition aKratzobliga :: "σ⇒σ" ("○kca")
24   where "○kcaφ ≡ λw. (∀s. ((Bestga<w><av(w)> )(s) ⟶ (φ)(s) ))"
25 definition pKratzobliga :: "σ⇒σ" ("○kcp")
26   where "○kcpφ ≡ λw. (∀s. ((Bestgp<w><pv(w)> )(s) ⟶ (φ)(s) ))"
27
28 abbreviation(input) msubintert :: "σ⇒σ⇒σ" (infix "∩" 54) where "φ ∩ ψ ≡ λx. φ x ∧ ψ x"
29
30 definition akratzdyadic :: "σ⇒σ⇒σ" ("○kca<_|_>")
31   where "○kca<ψ|φ> ≡ λw. (∀s. ((Bestga<w><(av(w) ∩ (φ))> )(s) ⟶ (ψ)(s) ))"
32 definition pkratzdyadic :: "σ⇒σ⇒σ" ("○kcp<_|_>")
33   where "○kcp<ψ|φ> ≡ λw. (∀s. ((Bestgp<w><(pv(w) ∩ (φ))> )(s) ⟶ (ψ)(s) ))"
```

FIGURE 7.4: Comparing CJL and Kratzer's system KD in Isabelle/HOL (scenario-1)

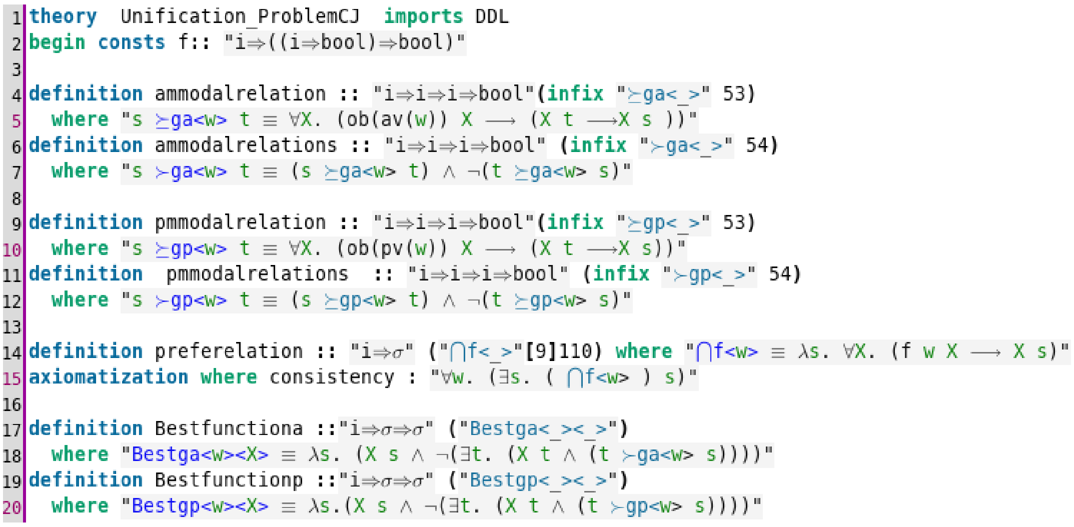
Nitpick confirms that the identity condition does not hold for both new monadic deontic operators. Also, Isabelle/HOL proves factual detachment property for both dyadic deontic operators (cf. Fig. 7.5).



```
lemma "⌊φ ⟶ ○kcaφ⌋" nitpick oops
lemma "⌊φ ⟶ ○kcpφ⌋" nitpick oops

lemma "⌊(○kca<ψ|φ>∧○kcaφ) ⟶ ○kcaψ⌋"
by(simp add: aBestfunction_def aKratzobliga_def akratzdyadic_def ddland_def ddlimp_def ddlvalid_def)
lemma "⌊(○kcp<ψ|φ>∧○kcpφ) ⟶ ○kcpψ⌋"
using ddland_def ddlimp_def ddlvalid_def pBestfunction_def pKratzobliga_def pkratzdyadic_def by auto

lemma "⌊○kcp<ψ|φ> ⟶ ○kca<ψ|φ>⌋" nitpick oops
lemma "⌊○kca<ψ|φ> ⟶ ○kcp<ψ|φ>⌋" nitpick oops
```

FIGURE 7.5: Comparing CJL and Kratzer's system KD in Isabelle/HOL (scenario-1)

Second scenario: In the second scenario, the actual and potential accessibility relations only play a role in the ordering source. The application of dyadic operation on *av* and *pv* provide two different ordering sources (cf. Fig. 7.6, lines 4–7 and 9–12). We have fixed the modal base source by a given modal base function *f* (cf. Fig. 7.6, line 2) (same as primary modal base function in the Kratzer's system) with consistency condition.

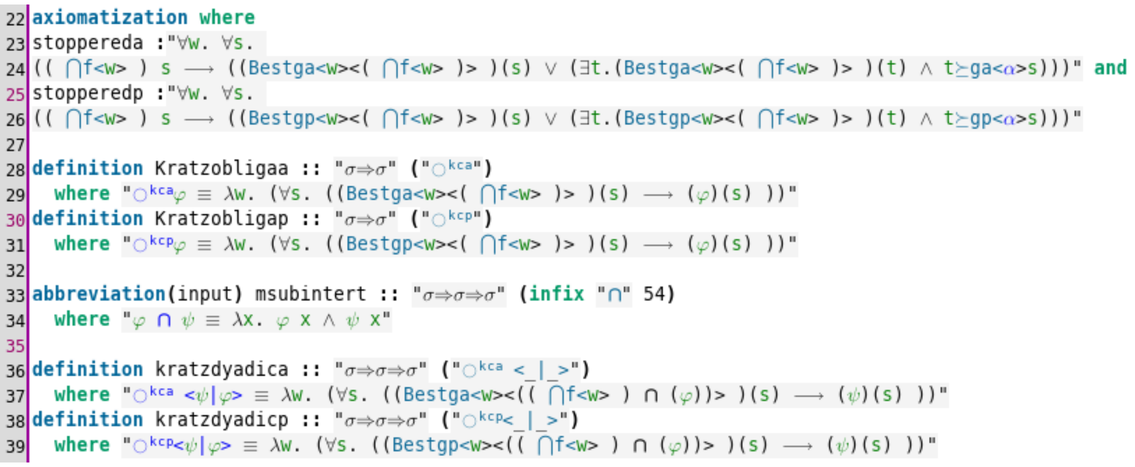


```
 1 theory  Unification_ProblemCJ  imports DDL
 2 begin consts f:: "i⇒((i⇒bool)⇒bool)"
 3
 4 definition ammodalrelation :: "i⇒i⇒i⇒bool"(infix "≿ga<_>" 53)
 5   where "s ≿ga<w> t ≡ ∀X. (ob(av(w)) X ⟶ (X t ⟶X s ))"
 6 definition ammodalrelations :: "i⇒i⇒i⇒bool" (infix "≻ga<_>" 54)
 7   where "s ≻ga<w> t ≡ (s ≿ga<w> t) ∧ ¬(t ≿ga<w> s)"
 8
 9 definition pmmodalrelation :: "i⇒i⇒i⇒bool"(infix "≿gp<_>" 53)
10   where "s ≿gp<w> t ≡ ∀X. (ob(pv(w)) X ⟶ (X t ⟶X s))"
11 definition  pmmodalrelations  :: "i⇒i⇒i⇒bool" (infix "≻gp<_>" 54)
12   where "s ≻gp<w> t ≡ (s ≿gp<w> t) ∧ ¬(t ≿gp<w> s)"
13
14 definition preferelation :: "i⇒σ" ("⋂f<_>"[9]110) where "⋂f<w> ≡ λs. ∀X. (f w X ⟶ X s)"
15 axiomatization where consistency : "∀w. (∃s. ( ⋂f<w> ) s)"
16
17 definition Bestfunctiona ::"i⇒σ⇒σ" ("Bestga<_><_>")
18   where "Bestga<w><X> ≡ λs. (X s ∧ ¬(∃t. (X t ∧ (t ≻ga<w> s))))"
19 definition Bestfunctionp ::"i⇒σ⇒σ" ("Bestgp<_><_>")
20   where "Bestgp<w><X> ≡ λs.(X s ∧ ¬(∃t. (X t ∧ (t ≻gp<w> s))))"
```

FIGURE 7.6: Comparing CJL and Kratzer's system KD in Isabelle/HOL (scenario-2)

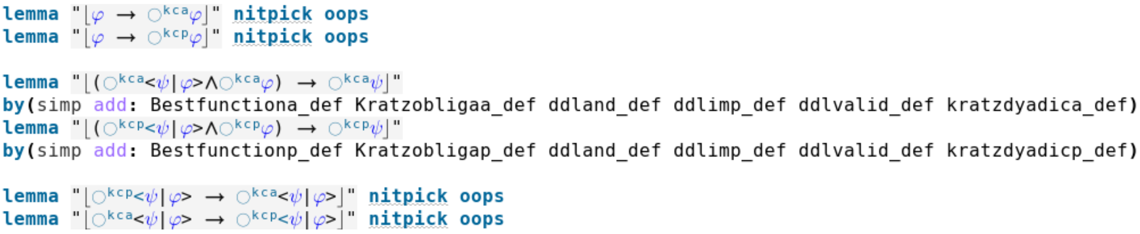
We have defined two monadic and dyadic deontic operators that are different only in the ordering sources. So the distinguishing feature of the actual obligation and potential obligations is their ordering sources (cf. Fig. 7.7).

Factual detachment holds for both operators in these new systems, and identity condition fails for actual and potential obligations (cf. Fig. 7.8).



```
axiomatization where
stoppereda :"∀w. ∀s.
(( ⋂f<w> ) s ⟶ ((Bestga<w><( ⋂f<w> )> )(s) ∨ (∃t.(Bestga<w><( ⋂f<w> )> )(t) ∧ t≥ga<α>s)))" and
stopperedp :"∀w. ∀s.
(( ⋂f<w> ) s ⟶ ((Bestgp<w><( ⋂f<w> )> )(s) ∨ (∃t.(Bestgp<w><( ⋂f<w> )> )(t) ∧ t≥gp<α>s)))"

definition Kratzobligaa :: "σ⇒σ" ("○kca")
  where "○kcaφ ≡ λw. (∀s. ((Bestga<w><( ⋂f<w> )> )(s) ⟶ (φ)(s) ))"
definition Kratzobligap :: "σ⇒σ" ("○kcp")
  where "○kcpφ ≡ λw. (∀s. ((Bestgp<w><( ⋂f<w> )> )(s) ⟶ (φ)(s) ))"

abbreviation(input) msubintert :: "σ⇒σ⇒σ" (infix "∩" 54)
  where "φ ∩ ψ ≡ λx. φ x ∧ ψ x"

definition kratzdyadica :: "σ⇒σ⇒σ" ("○kca <_|_>")
  where "○kca <ψ|φ> ≡ λw. (∀s. ((Bestga<w><(( ⋂f<w> ) ∩ (φ))> )(s) ⟶ (ψ)(s) ))"
definition kratzdyadicp :: "σ⇒σ⇒σ" ("○kcp<_|_>")
  where "○kcp<ψ|φ> ≡ λw. (∀s. ((Bestgp<w><(( ⋂f<w> ) ∩ (φ))> )(s) ⟶ (ψ)(s) ))"
```

FIGURE 7.7: Comparing CJL and Kratzer's system KD in Isabelle/HOL (scenario-2)



```
lemma "⌊φ → ○kcaφ⌋" nitpick oops
lemma "⌊φ → ○kcpφ⌋" nitpick oops

lemma "⌊(○kca<ψ|φ>∧○kcaφ) → ○kcaψ⌋"
by(simp add: Bestfunctiona_def Kratzobligaa_def ddland_def ddlimp_def ddlvalid_def kratzdyadica_def)
lemma "⌊(○kcp<ψ|φ>∧○kcpφ) → ○kcpψ⌋"
by(simp add: Bestfunctionp_def Kratzobligap_def ddland_def ddlimp_def ddlvalid_def kratzdyadicp_def)

lemma "⌊○kcp<ψ|φ> → ○kca<ψ|φ>⌋" nitpick oops
lemma "⌊○kca<ψ|φ> → ○kcp<ψ|φ>⌋" nitpick oops
```

FIGURE 7.8: Comparing CJL and Kratzer's system KD in Isabelle/HOL (scenario-2)

7.4 Conclusion

A shallow semantical embedding of Carmo and Jones's logic of contrary-to-duty conditionals in classical higher-order logic has been presented, and shown to be faithful (sound an complete). This implementation constitutes the first work in part of the larger LogiKEy project [37]. The introduced framework used to systematically analyses the properties of Carmo and Jones's dyadic deontic logic within Isabelle/HOL. A particular study was exploring Carmo and Jones's neighborhood semantics within the Kratzerian framework. The provided framework could be extended to study and support first-order and higher-order variants of the framework [91].

Chapter 8

Conclusion and Further Work

In this section, we summarize the result of this thesis and give a couple of suggestions for further research.

8.1 Discursive Input/Output Logic

In this thesis, we have introduced a non-adjunctive variant of input/output logic. Non-adjunctive logical systems are those where deriving the conjunctive formula $\varphi \wedge \psi$ from the set $\{\varphi, \psi\}$ fails [77, 78]. These systems are especially suited for modeling discursive reasoning because joining the discourse participants' opinions is not always possible. We build two groups of I/O operations for deriving permissions and obligations over Boolean algebras. The main difference between the two operations is similar to the possible world semantics characterization of box and diamond, where box is closed under AND, $((\Box\varphi\wedge\ \Box\psi) \rightarrow \Box(\varphi \wedge \psi))$, and diamond not. The new framework is a unification of the two main approaches for deontic logic: *modal logic* and *norm-based* approaches. There are other frameworks, such as adaptive logic [197, 198], that combine both approaches. The novelty of our approach is *semantical unification*. The semantical unification is based on bringing the core semantical elements of both approaches into a single unit.

8.1.1 Input/output logic for permission: Removing AND rule

Von Wright [220] defined permission as the primitive concept and obligation as the dual of it. Later, in the main literature of deontic logic, obligation introduced as the primitive concept and permission defined as the dual concept, as well in the original input/output logic for permission [142]. Moreover, in the I/O literature, permission based on derogation is studied by Stople [195, 194] and based on constraints by Boella and van der Torre [54].

In the main literature of input/output logic developed by Makinson and van der Torre [140], Parent, Gabbay, and van der Torre [163], Parent and van der Torre [165, 167, 169], and Stolpe [192, 193, 196] at least one form of AND inference rule is present (see the related table in Subsection 2.2.2). Sun [199] analyzed norms derivations rules of input/output logic in isolation. Still, it is not clear how we can combine them and build new logical systems, specifically systems that do not admit the rule of AND. For building a primitive operation for producing permissible propositions, we need to remove the AND rule from the proof system. We have characterized a class of proof systems over Boolean algebras, Heyting algebras, and any abstract logic where the rule of AND is absent for a sat of explicitly given norms as follows:

$derive_i$	Rules
$derive_R$	{EQO}
$derive_L$	{EQI}
$derive_0$	{EQI, EQO}
$derive_I$	{SI, EQO}
$derive_{II}$	{WO, EQI}
$derive_1$	{SI, WO}
$derive_2$	{SI, WO, OR}
$derive_3$	{SI, WO, T}

$$\text{EQO}\ \frac{(a,x) \qquad x=y}{(a,y)} \qquad\qquad \text{WO}\ \frac{(a,x) \qquad x\leq y}{(a,y)}$$

$$\text{EQI}\ \frac{(a,x) \qquad a=b}{(b,x)} \qquad\qquad \text{OR}\ \frac{(a,x) \qquad (b,x)}{(a\vee b,x)}$$

$$\text{SI}\ \frac{(a,x) \qquad b\leq a}{(b,x)} \qquad\qquad \text{T}\ \frac{(a,x) \qquad (x,y)}{(a,y)}$$

The corresponding output operations are defined as follows:

1. $out_R(N,A) = Eq(N(A))$
2. $out_L(N,A) = N(Eq(A))$
3. $out_I(N,A) = Eq(N(Eq(A)))$
4. $out_I(N,A) = Eq(N(Up(A)))$
5. $out_{II}(N,A) = Up(N(Eq(A)))$
6. $out_1(N,A) = Up(N(Up(A)))$
7. $out_2(N,A) = \bigcap\{Up(N(V)), A\subseteq V, V\text{is saturated}\}$
8. $out_3(N,A) = \bigcap\{Up(N(V)), A\subseteq V = Up(V)\supseteq N(V)\}$

The main characteristic difference in the new output operations and old ones [140] are using "Up" and "Eq" operators instead of "Cn" and saturated sets instead of complete sets.

There are other approaches that do not validate deontic aggregation principles. More interestingly, Ciabattoni et al. [76] introduced a proof-theoretic approach for reasoning about deontic modals and handling specificity that does not validate AND. It is an extension of minimal deontic logic [71]. A limitation of this approach is that the underlying non-normal deontic logics are relatively weak. As an advantage of our approach, we can add the rule of AND and cumulative transitivity (CT) to our system; see the next subsection.

8.1.2 Input/output logic for obligation: Adding AND rule

We have shown how we can rebuild four systems $derive_1^{AND}$, $derive_2^{AND}$, $derive_1^{CT,AND}$ and $derive_1^{CT,OR,AND}$ are introduced by Makinson and van der Torre [140] for reasoning about obligatory norms.

$derive_i^X$	Rules
$derive_1^{AND}$	{SI, WO, AND }
$derive_2^{AND}$	{SI, WO, OR, AND }
$derive_1^{CT,AND}$	{SI, WO, CT, AND }
$derive_1^{CT,OR,AND}$	{SI, WO, CT, OR, AND}

$$\text{AND}\ \frac{(a,x) \qquad (a,y)}{(a,x\wedge y)}$$

$$\text{CT}\ \frac{(a,x) \qquad (a\wedge x,y)}{(a,y)}$$

The output characterization, for instance in the case AND, has the following form:

$$\begin{aligned}
out_i^{AND^0}(N,A) &= out_i(N,A)\\
out_i^{AND^{n+1}}(N,A) &= out_i^{AND^n}(N,A) \cup\\
&\quad \{y\wedge z : y,z \in out_i^{AND^n}(N,\{a\}),\ \ a\in A\}\\
out_i^{AND}(N,A) &= \textstyle\bigcup_{n\in N} out_i^{AND^n}(N,A)
\end{aligned}$$

The proof is based on the reversibility of inference rules [140]. The same method can be used to define and characterize other rule-based systems.

$derive_i^X$	Rules
$derive_{II}^{AND}$	{WO, EQI, AND}
$derive_1^{AND}$	{SI, WO, AND}
$derive_2^{AND}$	{SI, WO, OR, AND}
$derive_I^{CT}$	{SI, EQO, CT}
$derive_{II}^{CT}$	{WO, EQI, CT}
$derive_1^{CT}$	{SI, WO, CT}
$derive_1^{CT,AND}$	{SI, WO, CT, AND}

$derive_i^X$	Rules
$derive_I^{OR}$	{SI, EQO, OR}
$derive_I^{CT,OR}$	{SI, EQO, CT, OR}
$derive_1^{CT,OR}$	{SI, WO, CT, OR}
$derive_1^{CT,OR,AND}$	{SI, WO, CT, OR, AND}

8.1.3 Semantic unification: Integrating input/output logic into the Kratzerian framework

Kratzerian framework has two contextual components: a *modal base* and an *ordering source*. Formally these are both functions, called conversational background, from evaluation worlds to sets of propositions. The *modal base* determines the set of accessible worlds and the *ordering source* induces the ordering on worlds [128, 218]. We employ these contextual components, the *modal base* and *ordering source* functions, from the Kratzerian framework [128, 129] and the *detachment* approach [164] from I/O framework, instead of quantification, for deriving deontic modals. As an advantage of the detachment approach, we can characterize derivation systems that do not admit, for example, weakening of the output (WO) or strengthening of the input (SI).

In input/output logic, the main semantical construct for normative propositions is the output operation, which represents the set of normative propositions related to the normative system N, regarding the state of affairs A, namely $out(N, A)$. *Detachment* is the basic idea of the semantics of input/output logic [164]. The interpretation of "x is obligatory if a" is that "x can be detached in context a". In a discourse, the context is represented by a modal base or an ordering source in the Kratzerian framework. To unify the norm-based semantics with the classic semantics, in each world w, we can detach what we allowed to or have to as the output of what we know (as the input set) represented as $\bigcap f(w)$, the intersection of the propositions given by the modal base, and the corresponding normative system N.

Consistent premise sets: Suppose $\bigcap f(w) \neq \emptyset$

$$[[\text{be-allowed-to}]]^{w,f} = \lambda N^P \lambda x \ (x \in out(N^P, \{\bigcap f(w)\}))$$

$$[[\text{have-to}]]^{w,f} = \lambda N^O \lambda x \ (x \in out(N^O, \{\bigcap f(w)\}))$$

In this case, deontic modals are evaluated with reference to a set of propositions given by the modal base and a normative system in each possible world. The modal bases are always factual. Whenever there are possible inconsistencies, we can take the content as an ordering source. If the set of $g(w)$ is not consistent, we can draw conclusions by looking at maximal consistent subsets.

Inconsistent premise sets: Suppose $\bigcap g(w) = \emptyset$,
and $\text{Maxfamily}^{\bigcap}(g(w)) = \{\bigcap A | A \subseteq g(w)\ \text{ and }\ A\ \text{ is consistent and maximal}\}$
$[[\text{be-allowed-to}]]^{w,g} = \lambda N^P \lambda x\ (x \in \mathit{out}(N^P, \text{Maxfamily}^{\bigcap}(g(w))))$
$[[\text{have-to}]]^{w,g} = \lambda N^O \lambda x\ (x \in \mathit{out}(N^O, \text{Maxfamily}^{\bigcap}(g(w))))$

Horty [112] raised the issue of unification for deontic logic. For example, he showed how van Fraassen's account [217] (as a norm-based approach) and SDL could be interpreted within the Kratzerian framework. He tried to show the norm-based approach's advantages, such as resolving normative conflicts and handling specificity, compared to the apparent advantages of modal logic, such as compositionality. We are still missing a semantics based on both the modal logic and the norm-based approaches. Our semantical unification is based on bringing the core semantical elements of both approaches into a single unit. For deriving (monadic) deontic modals the fragment is input/output logic based on the interaction between normative reasoning and informational and motivational modalities. We use the Kratzerian framework for representing different sets of information and motivation. For dyadic obligations and permissions, we add another fragment besides input/output logic based on the basic idea in classical semantics (or the Kratzerian framework): a preference ordering on possible worlds, and what ought to be the case, is determined by what is the case in all the best of the accessible worlds. Horty [112] pointed that with conditional obligations, at last, there is a real difference between the norm-based and modal logic approaches. As he discussed, there are two senses for conditional obligations: *constrained maximization* sense (classical semantics) and *resultant* sense (norm-based semantics) [112]. Our introduced compositional framework combines these two senses; see the next section.

8.1.4 Input/output methodology: Secretarial assistant

Makinson and van der Torre [140] introduced input/output logic as "secretarial assistant to an arbitrary process transforming propositional inputs into propositional outputs." The only input/output logics investigated in the literature so far are built on top of classical propositional logic [140] and intuitionist propositional logic [163]. It has been shown that we can build the input/output version of any abstract logic. An abstract logic [87] is a pair $\mathcal{A} = \langle \mathcal{L}, C \rangle$ where $\mathcal{L} = \langle L, ... \rangle$ is an algebra and C is a closure operator defined on the power set of its universe, that means for all $A, B \subseteq L$:

- $A \subseteq C(A)$
- $A \subseteq B \Rightarrow C(A) \subseteq C(B)$

- $C(A) = C(C(A))$

There is a similar result for building the simple-minded I/O operation over Tarskian consequence relations [68] (see the discussion about abstract input/output logic [199]).

There are important similarities between input/output logic and the theory of joining systems, such as: studying normative systems as deductive systems and representing norms as ordered pairs. Moreover, both frameworks can generally be built on top of algebraic structures such as Boolean algebras and lattices. While the focus in input/output logic is deontic and factual detachment in the theory of joining systems, the central theme is intermediate concepts and representing normative systems as a network of subsystems and relations between them; for more detail, see [136].

8.2 A Compositional Theory of Conditional Obligation and Permission

Following Thomason [204] and Bonevac [57], we have introduced a compositional theory of conditional obligation which separate the contributions of *if* and *ought*.

> "At the very least, a theorist using a conditional obligation operator owes us an explanation of how the semantics of the operator depends on the semantics for obligation and the conditional simpliciter. Sentences expressing conditional obligations are intelligible to anyone understanding should (or ought to) and if. The combination of these words is no idiom. The meanings of such sentences, therefore, should be explicable in terms of the meanings of if and should construed independently." (Bonevac [57], p. 37)

First, based on our results for building I/O operations over Boolean algebras, I/O operations' neighborhood characterization was presented.

It is a well-known fact that Boolean algebras are the algebraic models of (classical) propositional logic. We proved that the extension of propositional logic with a set of conditional norms is sound and complete respect to the class of Boolean algebras that the corresponding I/O operation holds through all of them.

$$(\varphi, \psi) \in derive_i^{\mathbf{Fm}(X)}(N)$$

if and only if

$$V(\psi) \in out_i^{\mathcal{B}}(N^V, \{V(\varphi)\}) \text{ for every } \mathcal{B} \in \mathbf{BA}, \text{ for every valuation } V \text{ on } \mathcal{B}$$

The valuation functions from propositional logic $\mathbf{Fm}(X)$ into the class of Boolean algebras $\mathcal{B}$ play the role of possible worlds, and the order over them supports the theory of conditionals.

We have analyzed conditional obligation sentences which have the form $a > \bigcirc x$, where $>$ is a classic conditional connective [131]. Given the set of obligatory norms N^O and suppose $(a, x) \in N^O$, we defined the new conditionals as follows:

$$a > \bigcirc x \text{ holds iff } (a, x) \in derive_i(N^O) \text{ and } a > x \text{ holds}$$

where $derive_i(N^O)$ is an appropriate derivation system for obligation. The proof system of the new compositional system is based on the included I/O proof system and the dyadic operator. For example, if we combine simple-minded output operation with the preference relation, the resulting compositional operator satisfies the rule of WO since both components satisfy this rule. More specifically, the set of new conditionals, $derive_i^O$, is not reflexive, it is not necessary that $a > \bigcirc x \in derive_i^O(N^O)$, for $(a, x) \in N^O$. Moreover, it does not validate the rule of SI due to the dyadic operator. Draping these two properties makes more intuitive input/output logic in facing contrary-to-duty statements, for the full discussion see Section 4.1.

For a given set of permissive norms N^P, by choosing a plausible derivation system $derive_i(N^P)$ for permission similar to the definition of conditional obligation, we can define the conditional permission.

$$a > Px \text{ holds iff } (a, x) \in derive_i(N^P) \text{ and } \neg(a > \neg x) \text{ holds}$$

The following theorems show that how the proposed system compose the theory of conditionals and our input/output logic for reasoning about conditional obligation and permission.

$\varphi > \bigcirc\psi \in derive_i^O(N^O)$
if and only if
$(\varphi, \psi) \in derive_i^{\mathbf{Fm}(X)}(N^O)$ and
For every preference Boolean algebra
$M = \langle \mathcal{B}, \mathcal{V}, \succeq_f \rangle$,
for every valuation $V_i \in opt_{\succeq_f}(\varphi)$ we have
$V_i(\psi) = 1_{\mathcal{B}}$

$\varphi > P\psi \in derive_i^P(N^P)$
if and only if
$(\varphi, \psi) \in derive_i^{\mathbf{Fm}(X)}(N^P)$ and
For every preference Boolean algebra
$M = \langle \mathcal{B}, \mathcal{V}, \succeq_f \rangle$,
there is a valuation $V_i \in opt_{\succeq_f}(\varphi)$ such that
$V_i(\psi) = 1_{\mathcal{B}}$

We have presented a semantical characterization for constrained I/O logic. Here, constraints are preferences. There are syntactical ([141], Section 6) and proof theoretical ([198], Section 3) characterizations for constrained I/O logic. Our semantical characterization is more flexible than the syntactical characterization since our approach does not necessarily depend on the rules AND, SI, and (EQI, EQO) required for syntactic characterization of the I/O operations in modal logic [198].

Moreover, we can use input/output logic as a non-monotonic defeat mechanism with Hansson and Lewis's conditional theory [109, 132] to detect dilemmas. A problem in most solutions (see [214, 213]) for detecting dilemmas is that the set of formulas $N = \{\bigcirc(\neg\psi/\varphi_1), \bigcirc(\psi/\varphi_2)\}$ is inconsistent or $\varphi_1 \wedge \varphi_2$ is impossible [213]. In our setting the set of formulas N is not necessary inconsistent, given $\{\varphi_1, \varphi_2\}$ consistent. We detect dilemmas by using detachment in the output operations.

8.3 A Computational Tool for Deontic Reasoning

8.3.1 Faithful embedding of some deontic logics in HOL

Another technical achievement of this paper is faithful embeddings of some deontic logics in higher-order logic. We have aligned Henkin models with neighborhood and preference models besides Kripke models.

Henkin models	Kripke models
Henkin models	Neighborhood models
Henkin models	Preference models

We studied how for a logical language $\mathbb{L}_s$ based on Kripke, neighborhood, and preference semantics, we could align these models with Henkin models such that:

$$\forall \Gamma \subseteq \mathbb{L}_s, \psi \in \mathbb{L}_s \quad (\Gamma, \psi) \in \vDash_s \text{ if and only if } (vld(\lfloor \Gamma \rfloor), vld(\lfloor \psi \rfloor)) \in \vDash_{HOL}.$$

$vld : HOL \to HOL$ is higher-order equivalent term of validity and $\lfloor . \rfloor : \mathbb{L}_s \to HOL$ is the translation function.

Moreover, it has been shown (Theorem 4.10) that for input/output derivation systems over propositional logic, $(\varphi, \psi) \in derive_i^{\mathbf{Fm}(X)}(N)$ holds if and only if $V(\psi) \in out_i^{\mathcal{B}}(N^V, \{V(\varphi)\})$ holds in all Boolean normative models $(\mathcal{N} = \langle \mathcal{B}, V, N^V \rangle)$. It has been proven that we can align Boolean normative models and Henkin models. In this way, we have introduced a faithful embedding of input/output logics in HOL.

8.3.2 Isabelle/HOL: An infrastructure for deontic reasoning

Deontic logic is developed within three fundamental frameworks: Modal-based deontic logic, Preference-based deontic logic, and Norm-based deontic logic. We presented the automation of at least one instance of each category that determines how each framework's automation will work. In Isabelle/HOL, we have provided a dataset of deontic logic implementations for legal and ethical reasoning tasks [27].

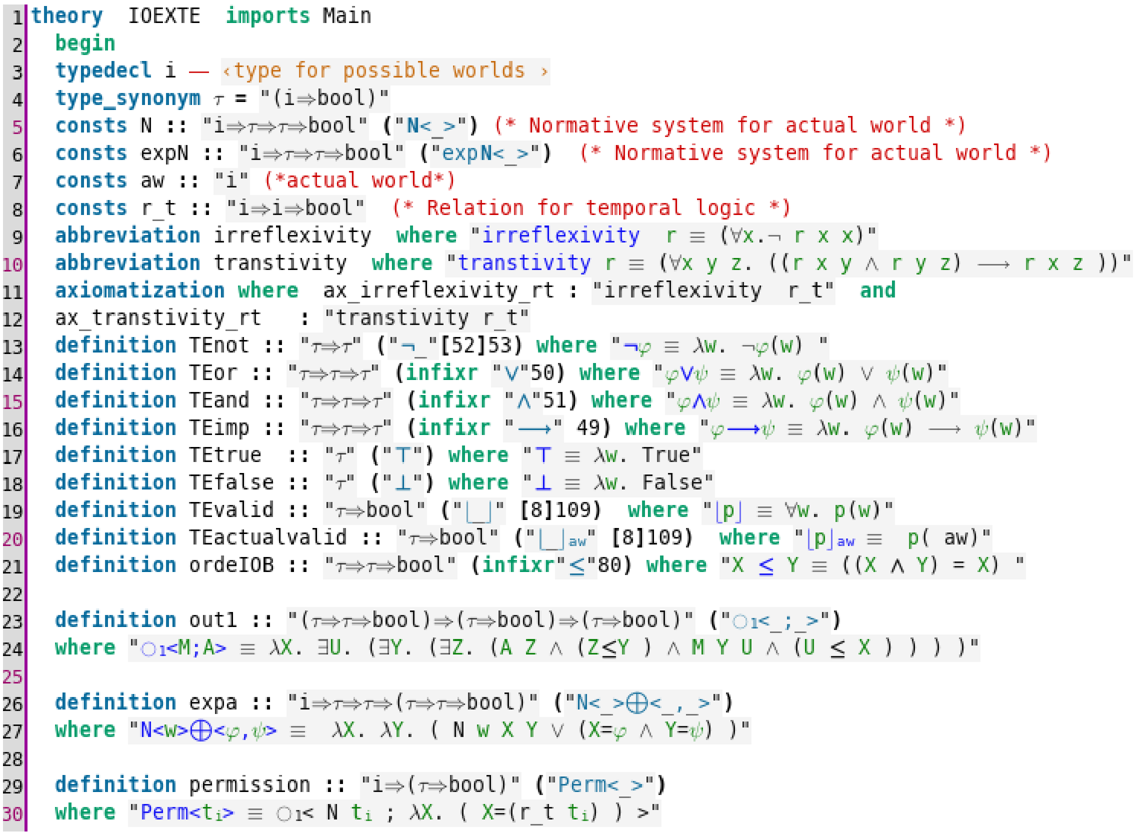
8.4 Future Work Through Discursive Input/Output Logic

8.4.1 A full characterization of rule-based logics

Makinson [139], in his seminal book *Bridges from Classical to Nonmonotonic Logic*, systematically shows that how we can, by using additional assumption, restrict valuations, and using additional rules define new consequence relations from classical consequence relation. In this thesis, we have defined a couple of different input/output rule-based logical systems, given an abstract logic $\mathcal{A} = \langle \mathcal{L}, C \rangle$ and a set of rules $R \subseteq \mathcal{L} \times \mathcal{L}$. It is worth studying the limits of these kinds of rule-based logical systems and finding full characterization for them. We pose combining I/O operations with consequence relations that do not satisfy inclusion or idempotence property [139] as a further research question. For example, combining the original I/O framework with a consequence relation that does not satisfy inclusion where the consequence relation is an input/output closure ($A \in out(N, A)$ is not necessary) was explored by Sun and van der Torre [201]. Moreover, we showed how we can add rules and restrict valuations simultaneously. It is worth studying the limitations and advantages of combining these three approaches.

8.4.2 Toward a unified logical framework for deontic reasoning

The thesis introduced a new logical framework for normative reasoning. It is a unification of the two main approaches for deontic logic: modal logic and norm-based approaches, see Section 1.3 and Section 2.2. An advantage of the modal logic approach is the capability to extend with other modalities such as epistemic or temporal operators, and advantages of the norm-based approach include the ability to explicitly represent normative codes such as legal systems and using non-monotonic logic techniques of common sense reasoning. For example, we can design a normative temporal system, which changes over time. The temporal reasoning comes from the advantage of the modal logic part and changes operators (expansion, contraction) from the norm-based part. For example, we have implemented a temporal logic in which the normative system expands over time. The formulation of the example is illustrated in Fig. 8.1. Lines 26–27 is used for defining the expansion function. Lines 23–24 and 29–30 define the input/output operation for permission over time. A further research area is exploring the expressivity power of the introduced unified semantics. Moreover, it is worth investigating the philosophical and conceptual advantages of integrating the norm-based semantics into the classic semantics [218, 112].



```
 1 theory  IOEXTE  imports Main
 2   begin
 3   typedecl i — ‹type for possible worlds ›
 4   type_synonym τ = "(i⇒bool)"
 5   consts N :: "i⇒τ⇒τ⇒bool" ("N<_>") (* Normative system for actual world *)
 6   consts expN :: "i⇒τ⇒τ⇒bool" ("expN<_>")  (* Normative system for actual world *)
 7   consts aw :: "i" (*actual world*)
 8   consts r_t :: "i⇒i⇒bool"  (* Relation for temporal logic *)
 9   abbreviation irreflexivity  where "irreflexivity  r ≡ (∀x.¬ r x x)"
10   abbreviation transtivity  where "transtivity r ≡ (∀x y z. ((r x y ∧ r y z) ⟶ r x z ))"
11   axiomatization where  ax_irreflexivity_rt : "irreflexivity  r_t"  and
12   ax_transtivity_rt    : "transtivity r_t"
13   definition TEnot :: "τ⇒τ" ("¬_"[52]53) where "¬φ ≡ λw. ¬φ(w)  "
14   definition TEor :: "τ⇒τ⇒τ" (infixr "∨"50) where "φ∨ψ ≡ λw. φ(w) ∨ ψ(w)"
15   definition TEand :: "τ⇒τ⇒τ" (infixr "∧"51) where "φ∧ψ ≡ λw. φ(w) ∧ ψ(w)"
16   definition TEimp :: "τ⇒τ⇒τ" (infixr "⟶" 49) where "φ⟶ψ ≡ λw. φ(w) ⟶ ψ(w)"
17   definition TEtrue  :: "τ" ("⊤") where "⊤ ≡ λw. True"
18   definition TEfalse :: "τ" ("⊥") where "⊥ ≡ λw. False"
19   definition TEvalid :: "τ⇒bool" ("⌊_⌋" [8]109)  where "⌊p⌋ ≡ ∀w. p(w)"
20   definition TEactualvalid :: "τ⇒bool" ("⌊_⌋aw" [8]109)  where "⌊p⌋aw ≡  p( aw)"
21   definition ordeIOB :: "τ⇒τ⇒bool" (infixr"≤"80) where "X ≤ Y ≡ ((X ∧ Y) = X) "
22
23   definition out1 :: "(τ⇒τ⇒bool)⇒(τ⇒bool)⇒(τ⇒bool)" ("○₁<_;_>")
24   where "○₁<M;A> ≡ λX. ∃U. (∃Y. (∃Z. (A Z ∧ (Z≤Y ) ∧ M Y U ∧ (U ≤ X ) ) ) )"
25
26   definition expa :: "i⇒τ⇒τ⇒(τ⇒τ⇒bool)" ("N<_>⨁<_,_>")
27   where "N<w>⨁<φ,ψ> ≡  λX. λY. ( N w X Y ∨ (X=φ ∧ Y=ψ) )"
28
29   definition permission :: "i⇒(τ⇒bool)" ("Perm<_>")
30   where "Perm<tᵢ> ≡ ○₁< N tᵢ ; λX. ( X=(r_t tᵢ) ) >"
```

FIGURE 8.1: A temporal normative system in Isabelle/HOL

8.5 Further Work Through Logical Implementations

A direct application of implementing logical systems in theorem provers is studying the source logic's meta-logical properties. For example, we proved that nested dyadic obligations in system **E** can be eliminated by the help of Isabelle/HOL. This result is just a piece of evidence for studying embedded logics within theorem provers; another one is cut-elimination for quantified conditional logic by Benzmüller [23]. Combing deontic logics with other implemented logical systems in HOL, such as temporal and epistemic logics and studying some meta-logical properties of these combinations, are other important research directions. So, we can study possible developments of deontic logical systems based on using HOL theorem provers and the logical implementations. Moreover, we can extend the expressivity of the implemented deontic logics in HOL by using the expressive power of HOL. Here we mention two possible directions.

8.5.1 Improving normative expressivity of the implemented logics in HOL

In the modern approach of studying deontic logic, the concentration is on the inference patterns in normative systems [166].

pattern	**names**
$\bigcirc\varphi_1, \bigcirc\varphi_2 / \bigcirc(\varphi_1 \wedge \bigcirc\varphi_2)$	AND
$\bigcirc\varphi_1 / \bigcirc(\varphi_1 \vee \varphi_2)$	W
$\bigcirc(\psi/\varphi_1) / \bigcirc(\psi/\varphi_1 \wedge \varphi_2)$	SA
$\bigcirc(\psi/\varphi \wedge \chi), \bigcirc(\varphi/\chi) / \bigcirc(\psi/\chi)$	CT

"Inference pattern" comes to the more general term of "property".

> "An *inference pattern* describes a *property* of a certain form." [166]

"Properties" are the main difference of normative systems. Parent and van der Torre [166] categorized the main deontic properties as follows:

Basic Properties of normative systems	Factual detachment and Violation detection: These two basic properties distinguish deontic logic from other intentional logics
Logical properties of normative systems	Substitution, Replacements of logical equivalence, Implication, Paraconsistency
Methodological properties of normative systems	Aggregation, Factual monotony, Norm monotony, Norm induction

Parent and van der Torre [166] argued that the most fundamental properties of normative systems are *factual detachment* and *violation detection.*

Factual detachment	**Violation detection**
• Weak sense: "if there is a norm with precisely the context as antecedent, then the output contains the consequent."[166] • Strong sense: "imposing detachment when the antecedent is implied by the context."[166]	• "Violation is the distinctive feature of norms and obligations with respect to other types intentional concepts. Violation detection is a property to make sure that violated obligations do not drown."[166]

Deontic properties as HOL terms The main dark side of preference-based deontic logic is the lack of factual detachment and violation detection properties. We can added these two properties as higher-order terms to our logical implementation.Van der Torre [210] inspiring KLM recognized the two most basic (violation detection) inference patterns in defeasible dyadic deontic logic.

- $\bigcirc(\psi/\varphi)$ is an overriding obligation (based on specificity) of $\bigcirc(\chi/\delta)$ iff $\psi \wedge \chi$ is inconsistent, and φ is more specific than δ.
- $\bigcirc(\psi/\varphi)$ is a contrary-to-duty obligation of $\bigcirc(\chi/\delta)$ iff $\varphi \wedge \chi$ is inconsistent.

We could represent *overriding obligation* and *contrary-to-duty obligation* as higher-order terms.

- Overiding-obligation$(\bigcirc(\psi/\varphi), \bigcirc(\chi/\delta))$
  iff $(\psi \wedge \chi \rightarrow \bot) \wedge (\delta \rightarrow \varphi)$

- Contrary-to-duty-obligation$(\bigcirc(\psi/\varphi), \bigcirc(\chi/\delta))$
  iff $\varphi \wedge \chi \rightarrow \bot$

Which could represented in HOL as follows:

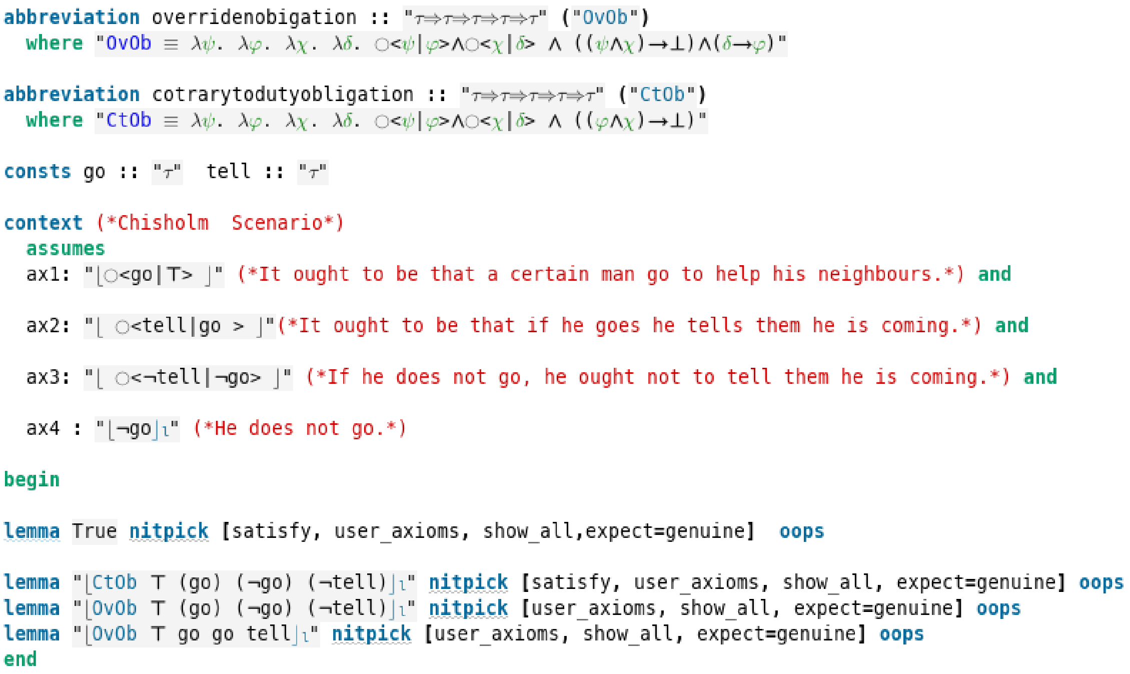
- Overiding-obligation $\equiv \lambda\psi.\lambda\varphi.\lambda\chi.\lambda\delta. \bigcirc (\psi/\varphi) \wedge \bigcirc(\chi/\delta) \wedge ((\psi \wedge \chi) \rightarrow \bot) \wedge (\delta \rightarrow \varphi)$

- Contrary-to-duty-obligation $\equiv \lambda\psi.\lambda\varphi.\lambda\chi.\lambda\delta. \bigcirc (\psi/\varphi) \wedge \bigcirc(\chi/\delta) \wedge ((\varphi \wedge \chi) \rightarrow \bot)$

Also, we could represent the property of factual detachment as follows:

$$\text{Factual-detachment}(\bigcirc\psi, \bigcirc(\psi/\varphi)) \quad \text{iff} \quad \bigcirc(\psi/\varphi) \wedge \varphi$$

We have implemented both definitions of overriding obligation and contrary-to-duty for the Chisholm's scenario in Isabelle/HOL, cf. Fig. 8.2.



```
abbreviation overridenobigation :: "τ⇒τ⇒τ⇒τ⇒τ" ("OvOb")
  where "OvOb ≡ λψ. λφ. λχ. λδ. ○<ψ|φ>∧○<χ|δ> ∧ ((ψ∧χ)→⊥)∧(δ→φ)"

abbreviation cotrarytodutyobligation :: "τ⇒τ⇒τ⇒τ⇒τ" ("CtOb")
  where "CtOb ≡ λψ. λφ. λχ. λδ. ○<ψ|φ>∧○<χ|δ> ∧ ((φ∧χ)→⊥)"

consts go :: "τ"  tell :: "τ"

context (*Chisholm  Scenario*)
  assumes
  ax1: "⌊○<go|T> ⌋" (*It ought to be that a certain man go to help his neighbours.*) and

  ax2: "⌊ ○<tell|go > ⌋"(*It ought to be that if he goes he tells them he is coming.*) and

  ax3: "⌊ ○<¬tell|¬go> ⌋" (*If he does not go, he ought not to tell them he is coming.*) and

  ax4 : "⌊¬go⌋ι" (*He does not go.*)

begin

lemma True nitpick [satisfy, user_axioms, show_all,expect=genuine]  oops

lemma "⌊CtOb T (go) (¬go) (¬tell)⌋ι" nitpick [satisfy, user_axioms, show_all, expect=genuine] oops
lemma "⌊OvOb T (go) (¬go) (¬tell)⌋ι" nitpick [user_axioms, show_all, expect=genuine] oops
lemma "⌊OvOb T go go tell⌋ι" nitpick [user_axioms, show_all, expect=genuine] oops
end
```

FIGURE 8.2: Improving dyadic deontic logic

8.5.2 Higher-order deontic logic

> "[...] quantifiers seem to me indispensable for any satisfactory analysis of the notions with which every system of deontic logic is likely to be concerned."
>
> (Hintikka [111], p.4)

Adding quantifier improve the expressivity of deontic logic. For example, we use quantifiers for most ethical and legal sentences.

Ethical expressions	**Legal expresions**
• Everyone ought to tell the truth. • Everyone ought to be honest.	• Everyone ought to pay the tax. • Everyone ought to pay the insurance.

Most presentations of deontic logic are restricted to propositional logic. For building a quantified deontic logic, there are some natural and essential questions.

The scope of 'Ought'	• De re $(\exists x) \bigcirc \varphi(x)$ • De dicto $\bigcirc(\exists x)\varphi(x)$
Substitution	From $t_1 = t_2$ and $\bigcirc\varphi[t_1]$ to infer $\bigcirc\varphi[t_2/t_1]$
Rules of extensional generalization	• From $\bigcirc\varphi[t]$ to infer $\exists x \bigcirc \varphi(x)$ • From $\forall x \bigcirc \varphi(x)$ to infer $\bigcirc\varphi[t]$

Goble [99] argues that obligations, unlike epistemic concepts such as "Knows that" and other modal concepts is an extensional concept.

> "Obligation, what one ought to do, pertains to individuals and their relations to others, and that seems to be independent of how these individuals are described or conceived." (Goble [99])

The definition of extensionality [19, 63] is based on the concept of equality. In simple type theory, there are two ways to introduce the equation.

- We could start with only primitive equality in the signature (for all types α) and introduce all other logical connectives as abbreviations based on it.
- We could remove primitive equality from the above signature since equality can be defined in HOL from these other logical connectives by exploiting Leibniz's principle, expressing that two objects are equal if they share the same properties.

$$= \ := \ (\lambda X_\alpha Y_\alpha \forall P_{\alpha\to o}(P_{\alpha\to o} X_\alpha \to P_{\alpha\to o} Y_\alpha))$$

In the second way in order to ensure the Henkin completeness [110], we need the functional extensionality axiom:

$$\forall f_{\alpha\to\beta} \forall g_{\alpha\to\beta}(\forall x_\beta(f_{\alpha\to\beta} x_\beta = g_{\alpha\to\beta} x_\beta) \to f_{\alpha\to\beta} = g_{\alpha\to\beta})$$

and the axiom for Boolean extensionality:

$$\forall p_o \forall q_o((p_o \leftrightarrow q_o) \to p_o = q_o)$$

Our deontic logic embeddings are faithful for both simple type theory with or without extensionality. There are implemented theorem provers such as LEO-II and Leo-III based on simple type theory with extensionality axiom.

As we have seen during the chapters, we have presented the different deontic operators as higher-order terms. So, in the HOL systems that we have the axiom of extensionality, our deontic operator should be extensional. Therefore, it could be reasonable that we add quantifiers over embedded deontic logics in HOL.

Also, for reasoning about individuals, we could add one more domain for individuals. Fuenmayor and Benzmüller [91] extended the CJL logic presented in Chapter 7 with a domain of individuals and a domain of context and presented some deontic quantified formulas for representing the Principle of Generic Consistency.

Moreover, the application of quantifiers could provide robust hybrid systems for legal and ethical knowledge representation. Following, we mention two examples from the literature that specify some ethical and legal procedures or concepts.

Example 1: Supported computer ethics Hoven and Lokhorst [209] presented a hybrid system for moral discourse, which makes use of deontic, epistemic, and action logic. The following are four axioms for this logical model that combine deontic operator $\bigcirc$, epistemic operator K, and action operator $[a\text{STIT} : \varphi]$ by using quantifier over the domain of individuals.

- $\bigcirc(K_a\varphi \rightarrow \forall x K_x\varphi)$: it ought to be the case that everybody knows what a knows.
- $\bigcirc[a\text{STIT} : \forall x K_x\varphi]$: a ought to see to it that everybody knows that φ.
- $[a\text{STIT} : \forall x(Fx \rightarrow B_x \bigcirc \varphi)]$: a sees to it that everybody who is F believes that φ is obligatory.
- $\neg(P[a\text{STIT} : \varphi] \rightarrow \forall x P[x\text{STIT} : \varphi])$

Example 2: Relativised deontic modalities for legal concepts Royakkers [183] extended deontic logic with a domain for individuals and introduced three kinds of obligations. Relativised deontic modalities are useful for expressing legal concepts such as right and power.

- a personal obligation: an obligation for a specific individual i: $\bigcirc_i(\varphi)$
- a general obligation: an obligation for all individuals: $\forall i \in I \bigcirc_i (\varphi)$ $(\forall i \in IP_i(\varphi))$
- an unspecific obligation: an obligation for some individual: $\exists i \in I \bigcirc_i (\varphi)$ $(\exists i \in IP_i(\varphi))$

Fo example, this formula

$$\neg(\bigcirc_i(\varphi) \wedge \exists j \in I \bigcirc_j (\neg\varphi))$$

express that "all obligations for all individuals can jointly be realised and that a permission for some individual should not be in conflict with his obligations."

8.6 Further Work Through Technological Developments

In this section we discuss a couple of related interdisciplinary domains such as normative multi-agent systems [73, 170], security [53], rational architecture [60, 61, 50], and artificial intelligence [37], how and why this thesis can contribute to them. This is mainly can be done by using theorem provers for deontic logic and introduced discursive input/output logic.

Logical, and mathematical frameworks, in general, have been shown to be useful for the design, specification, and for verification of systems. In particular, normative reasoning is essential in the development of ethical and legal systems where violation or exception of rules could happen. In this section, we review some possible technological environments that could be developed based on the expressivity power of normative systems.

8.6.1 A knowledge representation technology for normative multi-agent systems

> "Deontic logic is one of the formal tools needed in the design and specification of normative systems, where the latter are understood to be sets of agents (human or artificial) whose interactions can fruitfully be regarded as norm-governed; the norms prescribe how the agents ideally should and should not behave, [...] Importantly, the norms allow for the possibility that actual behavior may at times deviate from the ideal, i.e., that violations of obligations, or of agents' rights, may occur"
>
> (Carmo and Jones [66])

A multi-agent system (MAS) is a system that involves several autonomous entities that act in the same environment. The entities are called agents.

> "A normative multiagent system is a multiagent system organized by means of mechanisms to represent, communicate, distribute, detect, create, modify, and enforce norms, and mechanisms to deliberate about norms and detect norm violation and fulfillment." [55]

Rules are tools for collaboration and coordination for agents.

> "Some rules regulate antecedently existing forms of behaviour. For example, the rules of polite table behaviour regulate eating, but eating exists independently of these rules. Some rules, on the other hand, do not merely regulate an antecedently existing activity called playing chess; they, as it were, create the possibility of or define that activity. The activity of playing chess is constituted by action in accordance with these rules. The institutions of marriage, money, and promising are like the institutions of baseball and chess in that they are systems of such constitutive rules or conventions" (Searle [187], p. 131)

Constitutive norms are "count as" conditionals for the legal specification of concepts, for example, paper as money. Constitutive norms describe the creation of institutional facts like property, marriage.

Regulative norms are conditionals that refer to the legal classification of concepts and the exceptions in classifications. Regulative norms describe obligations, prohibitions, and permissions.

Deontic logic is useful for designing normative multi-agent systems [211, 170]. I/O mechanism can be used for designing the internal mechanism of multi-agent systems. Boella and van der Torre represented both constitutive and regulative norms in I/O setting [51, 170]. It is a direct research question to check the application of new input/output systems for multi-agent normative reasoning. Moreover, we can use the proposed logical implementations for automating the interaction of normative agents in computer.

8.6.2 A knowledge representation technology for privacy policies

Knowledge management (KM) is a crucial requirement of managing social networks and virtual communities [53].

> "Knowledge management is the systematic, holistic approach for sustainably improving the handling of knowledge on all levels of an organization

> (individual, group, organizational and inter-organizational level) in order to support the organization's business goals, such innovation, quality, cost effectiveness, etc." [215]

Beolla and van der Torre represented members of virtual communities as normative systems that interact with each other and poses prohibitions and permissions about access to their knowledge [53]. For sharing the knowledge between members of a virtual community, we need to distribute the security also. The past ten years have witnessed a considerable increase of attention to privacy in the media and, as a result, a massive increase in privacy policies with increasing complexity. Beolla and van der Torre based on normative multi-agent systems developed a model secure KM system with following requirements [53]:

- "First, participants should not give up their autonomy to prohibit access to knowledge to users they do not trust, even when the users satisfy the security rules of the virtual community."
- "Second, the rules of policies for managing knowledge in a secure way do not concern only what knowledge the users are prohibited or permitted to access, but they also concern which regulations their members are allowed or obliged to enforce."

The agents should decide to respect norms or not, according to rational balance between the advantage of not respecting a norm and the disadvantage of being sanctioned. Beolle and van der Torre developed a formal game-theoretic model using I/O framework for representing access control in KM. Another approach is based on developing deontic logic in a dynamic setting by using an epistemic operator for representing privacy policies [14]. We can implement both approaches in HOL and Isabelle/HOL. Moreover, we can explore the expressivity of I/O framework for knowledge management systems through discursive I/O operations.

8.6.3 A knowledge representation technology for rational architecture

BOID architecture

The BOID architecture is based on the cooperation and conflicts between beliefs, obligations, intentions, and desires, which represent the following mental states [60, 61].

Beliefs "represent the information of an agent about the current state of the world. All observations are turned into beliefs."	**Obligations** "represent attitudes that reflect the social nature of agents. Obligations can be violated, because agents are autonomous."
Intentions "are considered here as the generated goals that the agent has selected in its previous deliberations."	**Desires** "are long term preferences, and when a desire is triggered by an observation or belief, then a short-term desire is turned into a goal."

Agent architecture can be classified according to agent types. For example, an agent is realistic if the agent's beliefs override its obligations, intentions, or desires. BOID architecture use I/O mechanism for representing and resolving the conflicts among mental attitudes.

BOID Architecture	**input-output components**
Mental attitudes	Conditionals
Goal generation	Extension generation
Overriding	priority function

Normative architecture

> "a norm is that kind of guide to action that is supported by social sanctions."
>
> (Goffman [104])

Normative systems for the capability of controlling and regulating its behavior is autonomous and may be called a normative agent. We can attribute mental attitudes to normative systems [49, 50].

Regulative norms as desires: "obligations of the agents can be formalized as desires or goals of the normative agent. This representation may be paraphrased as "Your wish is my command", because the desires or wishes of the normative agent are the obligations or commands of the other agents. The goals of the normative system describe the ideal behavior of the system." [56]

Constitutive norms as beliefs: "we believe that the role of constitutive rules is not limited to the creation of an activity and the construction of new abstract categories. Constitutive norms specify both the behavior of a system and the evolution of the system: the normative system **n** itself specifies by means of its belief rules how its beliefs, desires and goals can be changed, who can change them, and the limits of the possible changes depending on the role played by an agent."[56]

As an advantage of attributing mental attitudes, we can model the interaction of an agent and a normative system as a game [50, 212].

> "If agent A is obliged to a, then agent N may decide that the absence of a counts as a violation of some norm n and that agent A must be sanctioned, and:
>
> 1 Agent A believes that agent N desires that A does a.
>
> 2 Agent A believes that agent N desires $\neg V(n)$, that there is no violation of norm n, but if agent N believes $\neg a$ then it has the goal $V(n)$, it counts as a violation.
>
> 3 Agent A believes that agent N desires $\neg s$, not to sanction, but if agent N decides $V(n)$ then it has as a goal that it sanctions agent A by doing s. Agent N only sanctions in case of violation. Moreover, agent A believes that agent N has a way to apply the sanction.
>
> 4 Agent A desires $\neg s$: it does not like the sanction.
>
> Symmetrically, permission can be modeled as an exceptional situation which does not count as a violation."[50]

The application of non-adjunctive input/output logic for representing and implementation the normative architecture is a further research area. For example, we have characterized different derivation systems including EQO or EQI (the rules for logical equivalences) that are basic rules for constitutive norms [120, 52]. As an example, we can characterize the system of rules $\{AND, EQI\}$ according to the reversibility of AND and EQI.

8.6.4 Online legal guidance systems

Artificial intelligence is growing up fast, and this leads to developing human-like intelligence technologies. According to the capabilities of AI systems for searching and giving advice, lawyers need new directions. This point is mentioned by Susskind [202].

> “what if we could find new, innovative ways of allowing our clients to tap into our knowledge and expertise? In particular, of course, what if we, as lawyers, could make our knowledge and expertise available through a wide range of online legal services, whether for the drafting of documents or for the resolution of disputes? If we can find online methods of enabling access to our experience and the service is thereby less costly, less cumbersome, more convenient, and quicker, then I suggest that clients, oppressed as they are by the more-for-less challenge, would welcome these services with arms flung open.” (Susskind [202])

Our deontic infrastructure could be a means of legal reasoning. Recently, Libal and Steen [134] developed an online website for legal reasoning based on theorem provers and deontic logic. Therefore, using deontic logic in automated deduction systems for designing legal systems in computers is a possible way for lawyers to use the advantages of new developments in AI. More specifically, input/output logic can be used for legal reasoning tasks. For example, reified input/output logic is a suitable formalism for expressing legal statements like those in the General Data Protection Regulation, for more details see the DAPRECO knowledge base [179]. It is promising to use the I/O framework within semantic web technologies, and the HOL theorem proves to build, in a reasonable time, extensive knowledge bases of formulas from legal texts.

In addition, the transparency of a legal system can be related to the capacity of exemplification of the laws that characterize that legal system. What we find in practicing a legal system are often specific rules. We need to find a reasoning pattern that allows us to deduce a general rule from each legal act’s specific rules. For example, in Islamic law (within *fiqh*) *qiyās*, known as correlational inference, provide a rational ground for the application of a juridical ruling to a given case not yet considered by the original juridical sources similar to the civil law. The methodology is studied in the computational framework of constructive type theory (CTT) by Shahid Rahman and his students [177]. CTT is expressive enough formalizing obligation [176] as the normative agent's goals, which conceptually is very similar to [50, 212]. It would be interesting to apply correlational inference for input/output framework with an appropriate computational framework such as CTT. Where logical systems provide a top-down approach for computational laws, correlational inference provides a bottom-up vision for the computational framework.

8.6.5 Autonomous cars

1 "A robot may not injure a human being or, through inaction, allow a human being to come to harm.

2 A robot must obey the orders given to it by human beings, except where such orders would conflict with the First Law.

3 A robot must protect its own existence, as long as such protection does not conflict with the First or Second Law."

(Isaac Asimov [13])

Autonomous vehicles are designed based on the control system. The classic paradigm for the control system is the optimal theory.

> "Optimal control deals with the problem of finding a control law for a given system such that a certain optimality criterion is achieved. A control problem includes a cost functional that is a function of state and control variables. An optimal control is a set of differential equations describing the paths of the control variables that minimize the cost function." (Wikipedia)

Deontic logic can be used to represent both deontological[1], and utilitarian[2] ethical frameworks. Ethics can be used positively as a control system for human behavior. Therefore, deontology and utilitarianism (or consequentialism) provide two types of algorithms for our reaction in ethical situations. We can draw an analogy between cast functions for an optimal control system and consequentialism.

Cost Functions	**Consequentialism**
Minimizing the error between the path taken by the vehicle and the desired path [98]	Minimizing the hazard of all objects in the environment present to the vehicle [98]

In dilemma situations, we need to prioritize the cost functions. Some cost functions have higher priority than others (not comparable) cost functions such as human life. So we could consider these functions as constraints for the system. Vehicles should satisfy the required constraints, but sometimes these constraints need to be violated. Some of these constraints are:

1. "An automated vehicle should not collide with a pedestrian or cyclist.
2. An automated vehicle should not collide with another vehicle, except where avoiding such a collision would conflict with the First Law.
3. An automated vehicle should not collide with any other object in the environment, except where avoiding such a collision would conflict with the First or Second Law." [97]

It makes sense that we draw an analogy between cost functions and deontology [97]. For example, stop signs and speed limits are deontological constraints for the traffic law, but they can be violated by ambulances in the emergency. However, we need to make a utilitarian decision for the smooth traffic flow or efficiency of traffic. It is open that how could we put the ethical and legal frameworks as control algorithms for autonomous vehicles? [97]. A possible way of combining deontology and consequentialism based on deontic logical systems is suggested by Baniasadi et

[1] Deontology is an approach to ethics that focuses on the rightness or wrongness of actions themselves, as opposed to the rightness or wrongness of the consequences of those actions. Modal-based deontic logic and norm-based deontic logic are base on deontology.

[2] Utilitarianism or consequentialism is an approach to ethics that argues that the morality of an action is contingent on the action's outcome or consequence [85]. Preference-based deontic logic is base on utilitarianism.

al. [16]. The LogiKEy methodology can be used for designing and implementing such kinds of systems.

8.6.6 Further feasible technologies

Communication Systems Jones and Parent presented a convention-based formal setting for communication [118]. They provide a multi-modal framework with doxastic and normative operators. This system can be implemented in the proposed deontic infrastructure and used for analyzing the communication networks.

Health-care advisor A theory of diagnosis is used for reasoning about violations. In particular, it reasons about the past with incomplete knowledge. A theory of diagnosis uses deontic logic to represent system rules and violations of these system rules. Sadeghzadeh argues that diagnostic-therapeutic knowledge is a kind of practical knowledge consisting of conditional obligations, and clinical medicine belongs to the realm of medical deontic [186].

> “Clinical-practical knowledge is deontic-procedural knowledge concerned with diagnosis, therapy, and prevention. [...] Diagnostic-therapeutic knowledge of this type is representable by conditional obligations. A conditional obligation, as an ought-to-do action rule, does not logically follow from descriptive or explanatory research that describes how things are, and explains why they occur. It is not justified by purely empirical research and evidence either. It comes from social institutions, i.e., medical communities in the present case, and is justified in comparison with alternative action rules by demonstrating that it is better than the latter. The comparative predicate “is better than”, however, is an evaluative one and has something to do with human values, intentions, and goals. Based on the considerations above, medical practice and research may be viewed as deontic disciplines that necessitate appropriate methods of inquiry termed medical deontics in preceding sections. The deontic character of medicine is exemplified by demonstrating that prototype diseases are deontic-social constructs. They are delimited as out-to-be-treated categories of states of affairs on the basis of common morality that requires

> the members of a society to charitably act in humanitarian emergency situations." (Sadegh-Zadeh [186])

Health technologies are enhancing by AI. Our implemented deontic logic dataset could be used for diagnostic-therapeutic knowledge representation for health care reasoning.

Fault-tolerant systems Deontic logic is a logical setting for the specification of normal and abnormal behavior. Abnormal behavior is the result of a violation. Faults in software systems may produce incorrect behavior, and therefore, corrective or recovery actions are needed. Deontic logic can specify fault-tolerant systems that can be executed adequately despite the occurrence of logic faults [69]. The presented deontic infrastructure and dataset could be used to detect faults of systems and specifications of fault-tolerant systems.

Verification tool for rule-based systems Input/output logic is a symbolic theory for the specification of rule-based systems. Logic programming is a rule-based system with the following rules based on a given language $\mathcal{L}$:

$$\varphi \leftarrow \psi_1, ..., \psi_n \quad \text{not } \delta_1, ... \text{ not } \delta_m$$

where $\varphi, \psi_1, ..., \psi_n, \delta_1, ..., \delta_m \in \mathcal{L}$.

The stable model semantics [95] is an important semantics for logic programming. Gonçalves and Alferes [105] provide a corresponding between I/O logic semantics and stable model semantics. Therefore, the implementation of I/O logic in theorem provers can be used as a verification tool for this class of logic programming.

Appendix A

Proofs

A.1 Appendix for Chapter 6

Proof for Lemma 6.3

The proof is similar to Lemma 7.4. We just check the case $\delta = \bigcirc(\psi/\varphi)$. We have the following chain of equivalences:

$\|\lfloor\bigcirc(\psi/\varphi)\rfloor S\|^{H^M,g[s/S_i]} = T$

$\Leftrightarrow$ $\|(\lambda X\forall W(\,(\lambda V(\lfloor\varphi\rfloor\,V \wedge (\forall Y(\lfloor\varphi\rfloor\,Y \to r\,V\,Y))))\,W \to \lfloor\psi\rfloor\,W))S\|^{H^M,g[s/S_i]} = T$

$\Leftrightarrow$ $\|\forall W(\,(\lambda V(\lfloor\varphi\rfloor\,V \wedge (\forall Y(\lfloor\varphi\rfloor\,Y \to r\,V\,Y))))\,W \to \lfloor\psi\rfloor\,W)\|^{H^M,g[s/S_i]} = T$

$\Leftrightarrow$ For all $u \in D_i$ we have:
$\|(\lambda V(\lfloor\varphi\rfloor\,V \wedge (\forall Y(\lfloor\varphi\rfloor\,Y \to r\,V\,Y))))\,W \to \lfloor\psi\rfloor\,W\|^{H^M,g[s/S_i][u/W_i]} = T$

$\Leftrightarrow$ For all $u \in D_i$ we have:
If $\|(\lambda V(\lfloor\varphi\rfloor\,V \wedge (\forall Y(\lfloor\varphi\rfloor\,Y \to r\,V\,Y))))\,W\|^{H^M,g[s/S_i][u/W_i]} = T$,
then $\|\lfloor\psi\rfloor W\|^{H^M,g[s/S_i][u/W_i]} = T$

$\Leftrightarrow$ For all $u \in D_i$ we have:
If $\|\lfloor\varphi\rfloor W\|^{H^M,g[s/S_i][u/W_i]} = T$ and
$\|\forall Y(\lfloor\varphi\rfloor\,Y \to r\,W\,Y)\|^{H^M,g[s/S_i][u/W_i]} = T$,
then $\|\lfloor\psi\rfloor V\|^{H^M,g[s/S_i][u/W_i]} = T$

$\Leftrightarrow$ For all $u \in D_i$ we have:
If $\|\lfloor\varphi\rfloor W\|^{H^M,g[s/S_i][u/W_i]} = T$ and
for all $t \in D_i$ we have $\|\lfloor\varphi\rfloor\,Y \to r\,W\,Y\|^{H^M,g[s/S_i][u/W_i][t/Y_i]} = T$,
then $\|\lfloor\psi\rfloor W\|^{H^M,g[s/S_i][u/W_i]} = T$

$\Leftrightarrow$ For all $u \in D_i$ we have:
If $\| \lfloor \varphi \rfloor W \|^{H^M, g[s/S_i][u/W_i]} = T$ and
for all $t \in D_i$ we have $\| \lfloor \varphi \rfloor Y \|^{H^M, g[s/S_i][u/W_i][t/Y_i]} = T$ implies $Ir_{i \to \tau}(u, t) = T$,
then $\| \lfloor \psi \rfloor W \|^{H^M, g[s/S_i][u/W_i]} = T$

$\Leftrightarrow$ For all $u \in D_i$ we have:
If $u \in V(\varphi)$ and
for all $t \in D_i$ we have $t \in V(\varphi)$ implies $u \succeq t$,
then $u \in V(\psi)$ (**see the justification ***)

$\Leftrightarrow$ $\text{opt}_{\succeq}(V(\varphi)) \subseteq V(\psi)$

$\Leftrightarrow$ $M, s \models \bigcirc(\psi/\varphi)$

Justification *: What we need to show is: $\| \lfloor \varphi \rfloor \|^{H^M, g[s/S_i]}$ is identified with $V(\varphi)$ (analogously ψ). By induction hypothesis, for all assignments g and states s, we have $\| \lfloor \varphi \rfloor S \|^{H^M, g[s/S_i]} = T$ if and only if $M, s \vDash \varphi$. Expanding the details of this equivalence we have: for all assignments g and states s

$$
\begin{aligned}
& s \in \| \lfloor \varphi \rfloor \|^{H^M, g[s/S_i]} \qquad \text{(functions to type } o \text{ are associated with sets)} \\
\Leftrightarrow\ & \| \lfloor \varphi \rfloor \|^{H^M, g[s/S_i]}(s) = T \\
\Leftrightarrow\ & \| \lfloor \varphi \rfloor \|^{H^M, g[s/S_i]} \| S \|^{H^M, g[s/S_i]} = T \\
\Leftrightarrow\ & \| \lfloor \varphi \rfloor S \|^{H^M, g[s/S_i]} = T \\
\Leftrightarrow\ & M, s \vDash \varphi \\
\Leftrightarrow\ & s \in V(\varphi)
\end{aligned}
$$

Hence, $s \in \| \lfloor \varphi \rfloor \|^{H^M, g[s/S_i]}$ if and only if $s \in V(\varphi)$.
By extensionality we thus know that $\| \lfloor \varphi \rfloor \|^{H^M, g[s/S_i]}$ is identified with $V(\varphi)$. Moreover, since H^M obeys the Denotatpflicht we know that $V(\varphi) \in D_\tau$.

Dyadic deontic logic F and G in HOL

The main challenge for a faithfulness semantical embedding of system **F** and **G** in higher-order logic is translating limitedness assumption into HOL.

- limitedness: if $V(\phi) \neq \emptyset$; then $\text{opt}_{\succeq}(V(\phi)) \neq \emptyset$

By replacing limitedness assumption with the stronger assumption (LIM) we could easily show the faithfulness of same shallow semantical embedding of system **E** that we provided in Section 6.2 for both systems **F** and **G**. But it is open that system **F** and **G** are sound and complete with respect of this new constraint for the preference models. In following with this conjecture we prove faithfulness of the embedding.

- LIM: if $X \subset P(W) \neq \emptyset$; then $\text{opt}_{\succeq}(X) \neq \emptyset$

Lemma A.1. *Let $H^M = \langle \{D_\alpha\}_{\alpha \in T}, I \rangle$ be a Henkin model for a preference model $M = \langle W, \succeq, V \rangle$ in which $\succeq$ is limited and transitive. We have $H^M \models^{HOL} TRA \wedge LMA$, where :*

(LIM) $\forall X_\tau (\exists W_i (X_\tau W_i) \rightarrow \exists Z_i (\lambda V_i (X_\tau V_i \wedge (\forall Y_i (X_\tau Y_i \rightarrow r_{i \rightarrow \tau} V_i Y_i)))) Z_i)$

(TRA) $\forall X_i Y_i Z_i ((r_{i \rightarrow \tau} X_i Y_i \wedge r_{i \rightarrow \tau} Y_i Z_i) \rightarrow r_{i \rightarrow \tau} X_i Z_i)$

Proof. (LIM)

Given arbitary assignment g, and arbitary $\bar{X} \in D_\tau$ such that
$\|\exists W(X\,W)\|^{H^M, g[\bar{X}/X_\tau]} = T$
$\Leftrightarrow$ There exists $w \in D_i$ such that $\|XW\|^{H^M, g[\bar{X}/X_\tau][w/W_i]} = T$
$\Leftrightarrow$ $\emptyset \neq \|X\|^{H^M, g[\bar{X}/X_\tau]} = \bar{X} \subseteq P(W)$
$\Leftrightarrow$ $\text{opt}_{\succeq}(\bar{X}) \neq \emptyset$ (by limitedness of $\succeq$)
$\{u \in \|X\|^{H^M, g[\bar{X}/X_\tau]} |$ For all t ($t \in \|X\|^{H^M, g[\bar{X}/X_\tau]}$ implies $Ir_{i \rightarrow \tau}(u, t) = T)\} \neq \emptyset$
$\Leftrightarrow$ There exists $u \in D_i$ such that
$\|\lambda V(X\,V \wedge (\forall Y(X\,Y \rightarrow r\,V\,Y)))Z\|^{H^M, g[\bar{X}/X_\tau][u/Z_i]} = T$
$\Leftrightarrow$ $\|\exists Z(\lambda V(X\,V \wedge (\forall Y(X\,Y \rightarrow r\,V\,Y))))Z\|^{H^M, g[\bar{X}/X_\tau]} = T$

Hence by definition of $\|.\|$, for all assignments g, for all $\bar{X} \in D_\tau$ we have
$\|\exists W(X\,W) \rightarrow \exists Z(\lambda V(X\,V \wedge (\forall Y(X\,Y \rightarrow r\,V\,Y))))Z\|^{H^M, g[\bar{X}/X_\tau]} = T$
$\Leftrightarrow$ For all assignments g we have
$\|\forall X(\exists W(X\,W) \rightarrow \exists Z(\lambda V(X\,V \wedge (\forall Y(X\,Y \rightarrow r\,V\,Y))))Z)\|^{H^M, g} = T$
$\Leftrightarrow$ $H^M \models^{\text{HOL}} LIM$

(TRA)

Given arbitary assignment g, and arbitary $s, t, u \in D_i$ such that
$\|r\,X\,Y \wedge r\,Y\,Z\|^{H^M, g[s/X_i][t/Y_i][u/Z_i]} = T$
$\Leftrightarrow$ $Ir_{i \rightarrow \tau}(s, t) = T$ and $Ir_{i \rightarrow \tau}(t, u) = T$

$\Leftrightarrow$ $Ir_{i\to\tau}(s,u)=T$ (by transitivity of $\succeq$)

$\Leftrightarrow$ $\|r\,X\,Z\|^{H^M,g[s/X_i][t/Y_i][u/Z_i]}=T$

Hence by definition of $\|.\|$, for all assignments g, for all $s,t,u\in D_i$ we have $\|(r\,X\,Y\wedge r\,Y\,Z)\to r\,X\,Z\|^{H^M,g[s/X_i][t/Y_i][u/Z_i]}=T$

$\Leftrightarrow$ For all assignments g we have $\|\forall XYZ(r\,X\,Y\wedge r\,Y\,Z)\to r\,X\,Z\|^{H^M,g}=T$

$\Leftrightarrow$ $H^M\models^{\mathrm{HOL}}TRA$

□

Theorem A.2 (Soundness and Completeness of the Embedding)**.**

$$\models^{\mathbf{F}}\varphi \textit{ if and only if } \{LIM\}\models^{HOL} vld\,\lfloor\varphi\rfloor$$

$$\models^{\mathbf{G}}\varphi \textit{ if and only if } \{LIM,\ TRA\}\models^{HOL} vld\,\lfloor\varphi\rfloor$$

Proof. We can prove Lemma 6.3 and 6.4 for system **F** and **G** and similarly prove soundness and completeness for system **F** and **G**. □

A.2 Appendix for Chapter 7

Proof for Lemma 7.3

We present detailed arguments for most cases.

AV:

For all $s\in D_i$: $Iav_{i\to\tau}(s)\neq\emptyset$ (by Lemma 1 (av))

$\Leftrightarrow$ For all $s\in D_i$, there exists $u\in D_i$ such that $Iav_{i\to\tau}(s,u)=T$

$\Leftrightarrow$ For all assignments g, for all $s\in D_i$, there exists $u\in D_i$ such that $\|av\,W\,V\|^{H^M,g[s/W_i][u/V_i]}=T$

$\Leftrightarrow$ For all g, all $s\in D_i$ we have $\|\exists V(av\,W\,V)\|^{H^M,g[s/W_i]}=T$

$\Leftrightarrow$ For all g we have $\|\forall W \exists V (av\, W\, V)\|^{H^M,g} = T$

$\Leftrightarrow$ $H^M \models^{\text{HOL}} AV$

PV1:

Given an arbitary assignment g, and arbitary $s, u \in D_i$ such that $\|av\, W\, V\|^{H^M,g[s/W_i][u/V_i]} = T$

$\Leftrightarrow$ $Iav_{i \to \tau}(s, u) = T$

$\Rightarrow$ $Ipv_{i \to \tau}(s, u) = T$ $\quad$ ($Iav_{i \to \tau}(s) \subseteq Ipv_{i \to \tau}(s)$, by Lemma 1 (pv1))

$\Leftrightarrow$ $\|pv\, W\, V\|^{H^M,g[s/W_i][u/V_i]} = T$

Hence by definition of $\|.\|$, for all g, for all $s, u \in D_i$ we have: $\|av\, W\, V\|^{H^M,g[s/W_i][u/V_i]} = T$ implies $\|pv\, W\, V\|^{H^M,g[s/W_i][u/V_i]} = T$

$\Leftrightarrow$ For all g, all $s, u \in D_i$ we have $\|av\, W\, V \to pv\, W\, V\|^{H^M,g[s/W_i][u/V_i]} = T$

$\Leftrightarrow$ For all g, all $s \in D_i$ we have $\|\forall V\ (av\, W\, V \to pv\, W\, V)\|^{H^M,g[s/W_i]} = T$

$\Leftrightarrow$ For all g we have $\|\forall W \forall V\ (av\, W\, V \to pv\, W\, V))\|^{H^M,g} = T$

$\Leftrightarrow$ $H^M \models^{\text{HOL}} PV1$

PV2:

This case is analogous to AV.

OB1:

For all $\bar{X} \in D_\tau$: $\emptyset \notin Iob_{\tau \to \tau \to o}(\bar{X})$ $\quad$ (by Lemma 1 (ob1))

$\Leftrightarrow$ For all g, all $\bar{X} \in D_\tau$ we have $\|\neg ob\, X\, (\lambda X. \bot)\|^{H^M,g[\bar{X}/X_\tau]} = T$

$\Leftrightarrow$ For all g we have $\|\forall X\, \neg(ob\, X\, (\lambda X_\tau \bot))\|^{H^M,g[\bar{X}/X_\tau]} = T$

$\Leftrightarrow$ $H^M \models^{\text{HOL}} OB1$

OB2:

Given an arbitary assignment g, and arbitary $\bar{X}, \bar{Y}, \bar{Z} \in D_\tau$ such that $\|\forall W((Y\, W \wedge X\, W) \longleftrightarrow (Z\, W \wedge X\, W))\|^{H^M,g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau]} = T$

$\Leftrightarrow$ For all $s \in D_i$ we have $\|(Y\, W \wedge X\, W) \longleftrightarrow (Z\, W \wedge X\, W)\|^{H^M,g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau][s/W_i]} = T$

$\Leftrightarrow$ For all $s \in D_i$ we have

$\|Y\,W \wedge X\,W\|^{H^M,g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau][s/W_i]} = T$ iff
$\|Z\,W \wedge X\,W\|^{H^M,g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau][s/W_i]} = T$

$\Leftrightarrow$ For all $s \in D_i$ we have $s \in \bar{Y} \cap \bar{X}$ iff $s \in \bar{Z} \cap \bar{X}$

$\Leftrightarrow$ $\bar{Y} \cap \bar{X} = \bar{Z} \cap \bar{X}$

$\Rightarrow$ $Iob_{\tau\to\tau\to o}(\bar{X}, \bar{Y}) = T$ iff $Iob_{\tau\to\tau\to o}(\bar{X}, \bar{Z}) = T$ (by Lemma 1 (ob2))

$\Leftrightarrow$ $\|ob\,X\,Y)\|^{H^M,g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau]} = T$ iff
$\|ob\,X\,Z\|^{H^M,g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau]} = T$

$\Leftrightarrow$ $\|ob\,X\,Y \longleftrightarrow ob\,X\,Z\|^{H^M,g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau]} = T$

Hence, by definition of $\|.\|$, for all g, for all $\bar{X}, \bar{Y}, \bar{Z} \in D_\tau$ we have:
$\|(\forall W(\,((Y\,W \wedge X\,W) \longleftrightarrow (Z\,W \wedge X\,W))$
$\to (ob\,X\,Y \longleftrightarrow ob\,X\,Z))\|^{H^M,g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau]} = T$

$\Leftrightarrow$ For all g we have
$\|\forall XYZ(\forall W(\,((Y\,W \wedge X\,W) \longleftrightarrow (Z\,W \wedge X\,W))$
$\to (ob\,X\,Y \longleftrightarrow ob\,X\,Z))\|^{H^M,g} = T$

$\Leftrightarrow$ $H^M \models^{\text{HOL}} OB2$

OB3:

Given an arbitary assignment g, and arbitary $\bar{\beta} \in D_{\tau\to o}, \bar{X} \in D_\tau$ such that
$\|\forall Z(\beta\,Z \to ob\,X\,Z)\|^{H^M,g[\bar{\beta}/\beta_{\tau\to o}][\bar{X}/X_\tau]} = T$ and
$\|\exists Z(\beta\,Z)\|^{H^M,g[\bar{\beta}/\beta_{\tau\to o}]} = T$ and
$\|\exists Y(((\lambda W \forall Z(\beta\,Z \to Z\,W))\,Y) \wedge X\,Y)\|^{H^M,g[\bar{\beta}/\beta_{\tau\to o}][\bar{X}/X_\tau]} = T$

$\Leftrightarrow$ For all $\bar{Z} \in D_\tau$ we have
$\|\beta\,Z\|^{H^M,g[\bar{\beta}/\beta_{\tau\to o}][\bar{X}/X_\tau][\bar{Z}/Z_\tau]} = T$ implies
$\|ob\,X\,Z\|^{H^M,g[\bar{\beta}/\beta_{\tau\to o}][\bar{X}/X_\tau][\bar{Z}/Z_\tau]} = T$ and
there exists $\bar{Z} \in D_\tau$ such that $\|\beta\,Z\|^{H^M,g[\bar{\beta}/\beta_{\tau\to o}][\bar{Z}/Z_\tau]} = T$ and
there exists $s \in D_i$ such that
$\|(\lambda W \forall Z(\beta\,Z \to Z\,W))\,Y \wedge X\,Y\|^{H^M,g[\bar{\beta}/\beta_{\tau\to o}][\bar{X}/X_\tau][s/Y_i]} = T$

$\Leftrightarrow$ For all $\bar{Z} \in D_\tau$ we have $\bar{Z} \in \beta$ implies $\bar{Z} \in Iob_{\tau\to\tau\to o}(\bar{X})$ and
there exists $\bar{Z} \in D_\tau$ such that $\bar{Z} \in \bar{\beta}$ and
there exists $s \in D_i$ such that $s \in \cap\bar{\beta}$ and $s \in \bar{X}$ **(see *)**

$\Leftrightarrow$ $\bar{\beta} \subseteq Iob_{\tau\to\tau\to o}(\bar{X})$ and $\bar{\beta} \neq \emptyset$ and $(\cap\bar{\beta}) \cap \bar{X} \neq \emptyset$

$\Rightarrow$ $Iob_{\tau\to\tau\to o}(\bar{X}, (\cap\bar{\beta})) = T$ (by Lemma 1 (ob3))

$\Leftrightarrow$ $\|ob\,X\,(\lambda W \forall Z(\beta\,Z \to Z\,W))\|^{H^M,g[\bar{\beta}/\beta_{\tau\to o}][\bar{X}/X_\tau]} = T$

Hence by definition of $\|.\|$, for all g, all $\bar{\beta} \in D_{\tau\to o}$, all $\bar{X} \in D_\tau$ we have:
$\|((\forall Z(\beta\, Z \to \mathit{ob}\, X\, Z)) \wedge (\exists Z(\beta\, Z)))$
$\to ((\exists Y(((\lambda W \forall Z(\beta\, Z \to Z\, W))Y) \wedge X\, Y))$
$\to \mathit{ob}\, X\, (\lambda W \forall Z(\beta\, Z \to Z\, W)))\|^{H^M, g[\bar{\beta}/\beta_{\tau\to o}][\bar{X}/X_\tau]} = T$

$\Leftrightarrow$ For all g, we have
$\|\forall\beta\forall X(((\forall Z(\beta\, Z \to \mathit{ob}\, X\, Z)) \wedge (\exists Z(\beta\, Z)))$
$\to ((\exists Y(((\lambda W \forall Z(\beta\, Z \to Z\, W))Y) \wedge X\, Y))$
$\to \mathit{ob}\, X\, (\lambda W \forall Z(\beta\, Z \to Z\, W))))\|^{H^M, g} = T$

$\Leftrightarrow$ $H^M \models^{\mathrm{HOL}} OB3$

Justification *: By definition of $\|.\|$, $\|\lambda W_i \forall Z_\tau(\beta_{\tau\to o} Z_\tau \to Z_\tau W_i)\|^{H^M, g[\bar{\beta}/\beta_{\tau\to o}][\bar{X}/X_\tau][s/Y_i]}$ is denoting the function f from D_i to D_o such that for all $d \in D_i$, $f(d) = \|\forall Z_\tau(\beta_{\tau\to o} Z_\tau \to Z_\tau W_i)\|^{H^M, g[\bar{\beta}/\beta_{\tau\to o}][\bar{X}/X_\tau][s/Y_i][d/W_i]}$. By definition of $\|.\|$, $\|\forall Z_\tau(\beta_{\tau\to o} Z_\tau \to Z_\tau W_i)\|^{H^M, g[\bar{\beta}/\beta_{\tau\to o}][\bar{X}/X_\tau][s/Y_i][d/W_i]} = T$ iff for all $\bar{Z} \in \bar{\beta}$ we have $d \in \bar{Z}$. Thus, f is the characteristic function of the set $\cap\bar{\beta}$. By the Denotatpflicht, which is obeyed in H^M, we know that $f(= \cap\bar{\beta}) \in D_\tau$.

OB4:

Given an arbitary assignment g, and arbitary $\bar{X}, \bar{Y}, \bar{Z} \in D_\tau$ such that
$\|\forall W(Y\, W \to X\, W) \wedge \mathit{ob}\, X\, Y \wedge$
$\forall W(X\, W \to Z\, W)\|^{H^M, g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau]} = T$

$\Leftrightarrow$ $\|\forall W(Y\, W \to X\, W)\|^{H^M, g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau]} = T$ and
$\|\mathit{ob}\, X\, Y\|^{H^M, g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau]} = T$ and
$\|\forall W(X\, W \to Z\, W)\|^{H^M, g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau]} = T$

$\Leftrightarrow$ For all $s \in D_i$ we have
($s \in \bar{Y}$ implies $s \in \bar{X}$) and $\bar{Y} \in Iob_{\tau\to\tau\to o}(\bar{X})$ and ($s \in \bar{X}$ implies $s \in \bar{Z}$)

$\Leftrightarrow$ $\bar{Y} \subseteq \bar{X}$ and $\bar{Y} \in Iob_{\tau\to\tau\to o}(\bar{X})$ and $\bar{X} \subseteq \bar{Z}$

$\Rightarrow$ $(\bar{Z} \setminus \bar{X}) \cup \bar{Y} \in Iob_{\tau\to\tau\to o}(\bar{Z})$ (by Lemma 1 (ob4))

$\Leftrightarrow$ $\|\mathit{ob}\, Z\, (\lambda W((Z\, W \wedge \neg X\, W) \vee Y\, W))\|^{H^M, g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau]} = T$ (**see ****)

Hence by definition of $\|.\|$ for all g, all $\bar{X}, \bar{Y}, \bar{Z} \in D_\tau$ we have
$\|(\forall W(Y\, W \to X\, W) \wedge \mathit{ob}\, X\, Y \wedge \forall W(X\, W \to Z\, W))$

$$
\begin{array}{ll}
 & \rightarrow ob\, Z\, (\lambda W((Z\, W \wedge \neg X\, W) \vee Y\, W))\|^{H^M, g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau]} = T \\
\Leftrightarrow & \text{For all } g \text{ we have} \\
 & \|\forall XYZ((\forall W(Y\, W \rightarrow X\, W) \wedge ob\, X\, Y \wedge \forall W(X\, W \rightarrow Z\, W)) \\
 & \rightarrow ob\, Z\, (\lambda W((Z\, W \wedge \neg X\, W) \vee Y\, W)))\|^{H^M, g} = T \\
\Leftrightarrow & H^M \models^{\mathrm{HOL}} OB4
\end{array}
$$

Justification **: Similar to justification *, we can convince ourselves that $\|\lambda W((Z\, W \wedge \neg X\, W) \vee Y\, W)\|^{H^M, g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau][\bar{Z}/Z_\tau]}$ is denoting the characteristic function f of the set $(\bar{Z} \setminus \bar{X}) \cup \bar{Y}$. By the Denotatpflicht, which is obeyed in H^M, we know that $f(= (\bar{Z} \setminus \bar{X}) \cup \bar{Y}) \in D_\tau$.

OB5:

This case is analogous to OB4.

□

Proof for Lemma 7.4

In the proof we implicitly employ curring and uncurring, and we associate sets with their characteristic functions. Throughout the proof whenever possible we omit types in order to avoid making the notation too cumbersome. The proof of Lemma 2.2 is by induction on the structure of δ. We start with the case where δ is p^j. We have

$$
\begin{array}{ll}
 & \|\lfloor p^j \rfloor S\|^{H^M, g[s/S_i]} = T \\
\Leftrightarrow & \|p^j_\tau S\|^{H^M, g[s/S_i]} = T \\
\Leftrightarrow & I p^j_\tau(s) = T \\
\Leftrightarrow & s \in V(p^j) \quad \text{(by definition of } H^M\text{)} \\
\Leftrightarrow & M, s \vDash p^j
\end{array}
$$

In the inductive cases we make use of the following **induction hypothesis**: *For sentences δ' structurally smaller than δ we have: For all assignments g and states s, $\|\lfloor \delta' \rfloor S\|^{H^M, g[s/S_i]} = T$ if and only if $M, s \vDash \delta'$.*

We consider each inductive case in turn:
(a) $\delta = \varphi \vee \psi$. In this case:

$$\begin{aligned}
& \|\lfloor \varphi \vee \psi \rfloor S\|^{H^M, g[s/S_i]} = T \\
\Leftrightarrow\ & \|(\lfloor \varphi \rfloor \vee_{\tau\to\tau\to\tau} \lfloor \psi \rfloor)S\|^{H^M, g[s/S_i]} = T \\
\Leftrightarrow\ & \|(\lfloor \varphi \rfloor S) \vee (\lfloor \psi \rfloor S)\|^{H^M, g[s/S_i]} = T \quad ((\lfloor \varphi \rfloor \vee_{\tau\to\tau\to\tau} \lfloor \psi \rfloor)S =_{\beta\eta} (\lfloor \varphi \rfloor S) \vee (\lfloor \psi \rfloor S)) \\
\Leftrightarrow\ & \|\lfloor \varphi \rfloor S\|^{H^M, g[s/S_i]} = T \text{ or } \|\lfloor \psi \rfloor S\|^{H^M, g[s/S_i]} = T \\
\Leftrightarrow\ & M, s \vDash \varphi \text{ or } M, s \vDash \psi \quad \text{(by induction hypothesis)} \\
\Leftrightarrow\ & M, s \vDash \varphi \vee \psi
\end{aligned}$$

(b) $\delta = \neg\varphi$. In this case:

$$\begin{aligned}
& \|\lfloor \neg\varphi \rfloor S\|^{H^M, g[s/S_i]} = T \\
\Leftrightarrow\ & \|(\neg_{\tau\to\tau} \lfloor \varphi \rfloor)S\|^{H^M, g[s/S_i]} = T \\
\Leftrightarrow\ & \|\neg(\lfloor \varphi \rfloor)S)\|^{H^M, g[s/S_i]} = T \quad ((\neg_{\tau\to\tau} \lfloor \varphi \rfloor)S =_{\beta\eta} \neg(\lfloor \varphi \rfloor S)) \\
\Leftrightarrow\ & \|\lfloor \varphi \rfloor S\|^{H^M, g[s/S_i]} = F \\
\Leftrightarrow\ & M, s \nvDash \varphi \quad \text{(by induction hypothesis)} \\
\Leftrightarrow\ & M, s \vDash \neg\varphi
\end{aligned}$$

(c) $\delta = \Box\varphi$. We have the following chain of equivalences:

$$\begin{aligned}
& \|\lfloor \Box\varphi \rfloor S\|^{H^M, g[s/S_i]} = T \\
\Leftrightarrow\ & \|(\lambda X \forall Y (\lfloor \varphi \rfloor Y))S\|^{H^M, g[s/S_i]} = T \\
\Leftrightarrow\ & \|\forall Y (\lfloor \varphi \rfloor Y)\|^{H^M, g[s/S_i]} = T \\
\Leftrightarrow\ & \text{For all } a \in D_i \text{ we have } \|\lfloor \varphi \rfloor Y\|^{H^M, g[s/S_i][a/Y_i]} = T \\
\Leftrightarrow\ & \text{For all } a \in D_i \text{ we have } \|\lfloor \varphi \rfloor Y\|^{H^M, g[a/Y_i]} = T \quad (S \notin free(\lfloor \varphi \rfloor) = \emptyset) \\
\Leftrightarrow\ & \text{For all } a \in D_i \text{ we have } M, a \models \varphi \quad \text{(by induction hypothesis)} \\
\Leftrightarrow\ & M, s \models \Box\varphi
\end{aligned}$$

(d) $\delta = \Box_a\varphi$:

$$\begin{aligned}
& \|\lfloor \Box_a\varphi \rfloor S\|^{H^M, g[s/S_i]} = T \\
\Leftrightarrow\ & \|\,(\lambda X \forall Y (\neg av\, X\, Y \vee \lfloor \varphi \rfloor Y))\, S\|^{H^M, g[s/S_i]} = T \\
\Leftrightarrow\ & \text{For all } a \in D_i \text{ we have } \|\neg av\, S\, Y \vee \lfloor \varphi \rfloor Y\|^{H^M, g[s/S_i][a/Y_i]} = T \\
\Leftrightarrow\ & \text{For all } a \in D_i \text{ we have } \|av\, S\, Y\|^{H^M, g[s/S_i][a/Y_i]} = F \text{ or} \\
& \|\lfloor \varphi \rfloor Y\|^{H^M, g[s/S_i][a/Y_i]} = T \\
\Leftrightarrow\ & \text{For all } a \in D_i \text{ we have } Iav_{i\to\tau}(s, a) = F \text{ or} \\
& \|\lfloor \varphi \rfloor Y\|^{H^M, g[a/Y_i]} = T \qquad (S \notin free(\lfloor \varphi \rfloor))
\end{aligned}$$

$\Leftrightarrow$ For all $a \in S$ we have $a \notin av(s)$ or
$M, a \models \varphi$ (by induction hypothesis)
$\Leftrightarrow$ $M, s \models \Box_a \varphi$

(e) $\delta = \Box_p \varphi$.

The argument is analogous to $\delta = \Box_a \varphi$.

(f) $\delta = \bigcirc(\psi/\varphi)$:

$\|\lfloor\bigcirc(\psi/\varphi)\rfloor S\|^{H^M,g[s/S_i]} = T$
$\Leftrightarrow$ $\|(\lambda X(ob\lfloor\psi\rfloor\lfloor\varphi\rfloor))S\|^{H^M,g[s/S_i]} = T$
$\Leftrightarrow$ $\|ob\lfloor\psi\rfloor\lfloor\varphi\rfloor\|^{H^M,g[s/S_i]} = T$
$\Leftrightarrow$ $Iob_{\tau\to\tau\to o}(\|\lfloor\psi\rfloor\|^{H^M,g[s/S_i]})(\|\lfloor\varphi\rfloor\|^{H^M,g[s/S_i]}) = T$
$\Leftrightarrow$ $\|\lfloor\varphi\rfloor\|^{H^M,g[s/S_i]} \in Iob_{\tau\to\tau\to o}(\|\lfloor\psi\rfloor\|^{H^M,g[s/S_i]})$
$\Leftrightarrow$ $V(\varphi) \in Iob_{\tau\to\tau\to o}(V(\psi))$ (**see *****)
$\Leftrightarrow$ $V(\varphi) \in ob(V(\psi))$
$\Leftrightarrow$ $M, s \models \bigcirc(\psi/\varphi)$

Justification *:** We need to show that $\|\lfloor\varphi\rfloor\|^{H^M,g[s/S_i]}$ is identified with $V(\varphi) = \{s \in S \mid M, s \models \varphi\}$ (analogous for ψ). By induction hypothesis, for all assignment g and world s, we have $\|\lfloor\varphi\rfloor S\|^{H^M,g[s/S_i]} = T$ if and only if $M, s \vDash \varphi$. We expand the details of this equivalence. For all assignment g and all worlds $s \in D_i$ we have

$s \in \|\lfloor\varphi\rfloor\|^{H^M,g[s/S_i]}$ (charact. functions are associated with sets)
$\Leftrightarrow$ $\|\lfloor\varphi\rfloor\|^{H^M,g[s/S_i]}(s) = T$
$\Leftrightarrow$ $\|\lfloor\varphi\rfloor\|^{H^M,g[s/S_i]}(\|S\|^{H,g[s/S_i]}) = T$
$\Leftrightarrow$ $\|\lfloor\varphi\rfloor S\|^{H^M,g[s/S_i]} = T$
$\Leftrightarrow$ $M, s \vDash \varphi$ (induction hypothesis)
$\Leftrightarrow$ $s \in V(\varphi)$

Hence, $s \in \|\lfloor\varphi\rfloor\|^{H^M,g[s/S_i]}$ if and only if $s \in V(\varphi)$. By extensionality we thus know that $\|\lfloor\varphi\rfloor\|^{H^M,g[s/S_i]} = V(\varphi)$. Moreover, since H^M obeys the Denotatpflicht we know that $V(\varphi) \in D_\tau$.

(g) $\delta = \bigcirc_a \varphi$:

$\| \lfloor \bigcirc_a \varphi \rfloor S \|^{H^M, g[s/S_i]} = T$

$\Leftrightarrow \quad \| (\lambda X (ob\,(av\,X) \lfloor \varphi \rfloor \wedge \exists Y (av\,X\,Y \wedge \neg(\lfloor \varphi \rfloor Y)))S \|^{H^M, g[s/S_i]} = T$

$\Leftrightarrow \quad \| ob\,(av\,S) \lfloor \varphi \rfloor \wedge \exists Y (av\,S\,Y \wedge \neg(\lfloor \varphi \rfloor Y)) \|^{H^M, g[s/S_i]} = T$

$\Leftrightarrow \quad \| ob\,(av\,S) \lfloor \varphi \rfloor \|^{H^M, g[s/S_i]} = T$ and

$\| \exists Y (av\,S\,Y \wedge \neg(\lfloor \varphi \rfloor Y)) \|^{H^M, g[s/S_i]} = T$

$\Leftrightarrow \quad \| ob\,(av\,S) \lfloor \varphi \rfloor \|^{H^M, g[s/S_i]} = T$ and

there exists $a \in D_i$ such that $\| av\,S\,Y \wedge \neg(\lfloor \varphi \rfloor Y) \|^{H^M, g[s/S_i][a/Y_i]} = T$

$\Leftrightarrow \quad I ob_{\tau \to \tau \to o}(\| av\,S \|^{H^M, g[s/S_i]})(\| \lfloor \varphi \rfloor \|^{H^M, g[s/S_i]}) = T$ and

there exists $a \in D_i$ such that

$\| av\,X\,Y \|^{H^M, g[s/S_i][a/Y_i]} = T$ and $\| \lfloor \varphi \rfloor Y \|^{H^M, g[s/S_i][a/Y_i]} = F$

$\Leftrightarrow \quad \| \lfloor \varphi \rfloor \|^{H^M, g[s/S_i]} \in I ob_{\tau \to \tau \to o}(\| av\,S \|^{H^M, g[s/S_i]})$ and

there exists $a \in D_i$ such that

$\| av\,X\,Y \|^{H^M, g[s/S_i][a/Y_i]} = T$ and $\| \lfloor \varphi \rfloor Y \|^{H^M, g[s/S_i][a/Y_i]} = F$

$\Leftrightarrow \quad V(\varphi) \in I ob_{\tau \to \tau \to o}(\| av\,S \|^{H^M, g[s/S_i]})$ and **(similar to ***)**

there exists $a \in D_i$ such that

$\| av\,X\,Y \|^{H^M, g[a/Y_i]} = T$ and $\| \lfloor \varphi \rfloor Y \|^{H^M, g[a/Y_i]} = F$

$\Leftrightarrow \quad V(\varphi) \in I ob_{\tau \to \tau \to o}(av(s))$ and **(similar to ***)**

there exists $a \in D_i$ such that

$\| av\,X\,Y \|^{H^M, g[a/Y_i]} = T$ and $\| \lfloor \varphi \rfloor Y \|^{H^M, g[a/Y_i]} = F$ $\quad (S \notin free(\lfloor \varphi \rfloor))$

$\Leftrightarrow \quad V(\varphi) \in ob(av(s))$ and

there exists $a \in S$ such that

$a \in av(s)$ and $M, a \not\models \varphi$ (by induction hypothesis)

$\Leftrightarrow \quad V(\varphi) \in ob(av(s))$ and

there exists $a \in S$ such that $a \in av(s)$ and $a \notin V(\varphi)$

$\Leftrightarrow \quad V(\varphi) \in ob(av(s))$ and

there exists $a \in S$ such that $a \in av(s) \cap V(\neg\varphi)$

$\Leftrightarrow \quad V(\varphi) \in ob(av(s))$ and $av(s) \cap V(\neg\varphi) \neq \emptyset$

$\Leftrightarrow \quad M, s \models \bigcirc_a \varphi$

(h) $\delta = \bigcirc_p \varphi$:

The argument is analogous to $\delta = \bigcirc_a \varphi$. $\square$

Bibliography

[1] Alchourrón, C. and Bulygin, E. (1971). *Normative systems.* Springer-Verlag; Wien New York.

[2] Alchourrón, C. E. (1991). Conflicts of norms and the revision of normative systems. *Law and Philosophy*, 10(4):413–425.

[3] Ambrossio, D. A. (2017). *Non-monotonic logics for access control: Delegation revocation and distributed policies.* PhD thesis, University of Luxembourg.

[4] Anderson, M. and Anderson, S. L. (2011). *Machine ethics.* Cambridge University Press.

[5] Andrews, P. (1971). Resolution in type theory. *Journal of Symbolic Logic*, 36(3):414–432.

[6] Andrews, P. (1972a). General models and extensionality. *Journal of Symbolic Logic*, 37(2):395–397.

[7] Andrews, P. (1972b). General models, descriptions, and choice in type theory. *Journal of Symbolic Logic*, 37(2):385–394.

[8] Andrews, P. (2014). Church's type theory. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy.* Metaphysics Research Lab, Stanford University, spring 2014 edition.

[9] Åqvist, L. (1984). Deontic logic. In *Handbook of Philosophical Logic*, pages 605–714. Springer.

[10] Åqvist, L. (1987). *Introduction to deontic logic and the theory of normative systems.* Biblopolis; Napoli.

[11] Åqvist, L. (2000). Three characterizability problems in deontic logic. *Nordic Journal of Philosophical Logic*, 5(2):65–82.

[12] Åqvist, L. (2002). Deontic logic. In *Handbook of Philosophical Logic*, pages 147–264. Springer.

[13] Asimov, I. (1930). *I, Robot.* Digit Books.

[14] Aucher, G., Boella, G., and van der Torre, L. (2011). A dynamic logic for privacy compliance. *Artificial Intelligence and Law*, 19(2-3):187–231.

[15] Bachmair, L. and Ganzinger, H. (2001). Resolution theorem proving. In *Handbook of Automated Reasoning*, pages 19–99. Elsevier.

[16] Baniasadi, Z., Parent, X., Max, C., and Cramer, M. (2018). A model for regulating of ethical preferences in machine ethics. In *International Conference on Human-Computer Interaction*, pages 481–506. Springer.

[17] Belanyek, A., Grossi, D., and van der Hoek, W. (2017). A note on nesting in dyadic deontic logic. CoRR abs/1710.03481.

[18] Belzer, M. (1986). A logic of deliberation. In Kehler, T., editor, *Proceedings of the 5th National Conference on Artificial Intelligence. Philadelphia, PA, USA, August 11-15, 1986. Volume 1: Science*, pages 38–43. Morgan Kaufmann.

[19] Benzmüller, C. (1999). *Equality and extensionality in automated higher-order theorem proving.* PhD thesis, Saarland University.

[20] Benzmüller, C. (2009). Automating access control logics in simple type theory with LEO-II. In *IFIP International Information Security Conference*, pages 387–398. Springer.

[21] Benzmüller, C. (2011). Combining and automating classical and non-classical logics in classical higher-order logics. *Annals of Mathematics and Artificial Intelligence*, 62(1-2):103–128.

[22] Benzmüller, C. (2013). Automating quantified conditional logics in HOL. In Rossi, F., editor, *23rd International Joint Conference on Artificial Intelligence (IJCAI-13)*, pages 746–753, Beijing, China. AAAI Press.

[23] Benzmüller, C. (2017). Cut-elimination for quantified conditional logic. *Journal of Philosophical Logic*, 46(3):333–353.

[24] Benzmüller, C. (2019). Universal (meta-) logical reasoning: Recent successes. *Science of Computer Programming*, 172:48–62.

[25] Benzmüller, C. and Andrews, P. (2019). Church's type theory. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*, pages pp. 1–62 (in pdf version). Metaphysics Research Lab, Stanford University, summer 2019 edition.

[26] Benzmüller, C., Brown, C., and Kohlhase, M. (2004). Higher-order semantics and extensionality. *Journal of Symbolic Logic*, 69(4):1027–1088.

[27] Benzmüller, C., Farjami, A., Fuenmayor, D., Meder, P., Parent, X., Steen, A., van der Torre, L., and Zahoransky, V. (2020a). LogiKEy workbench: Deontic logics, logic combinations and expressive ethical and legal reasoning (Isabelle/HOL dataset). *Data in Brief*, (106409):1–15.

[28] Benzmüller, C., Farjami, A., Meder, P., and Parent, X. (2019a). I/O logic in HOL. *Journal of Applied Logics – IfCoLoG Journal of Logics and their Applications (Special Issue on Reasoning for Legal AI)*, 6(5):715–732.

[29] Benzmüller, C., Farjami, A., and Parent, X. (2018a). A dyadic deontic logic in HOL. In Broersen, J., Condoravdi, C., Nair, S., and Pigozzi, G., editors, *Deontic Logic and Normative Systems — 14th International Conference, DEON 2018, Utrecht, The Netherlands, 3-6 July, 2018*, volume 9706, pages 33–50. College Publications.

[30] Benzmüller, C., Farjami, A., and Parent, X. (2018b). Faithful semantical embedding of a dyadic deontic logic in HOL. CoRR abs/1802.08454.

[31] Benzmüller, C., Farjami, A., and Parent, X. (2019b). Åqvist's dyadic deontic logic E in HOL. *Journal of Applied Logics – IfCoLoG Journal of Logics and their Applications (Special Issue on Reasoning for Legal AI)*, 6(5):733–755.

[32] Benzmüller, C., Gabbay, D., Genovese, V., and Rispoli, D. (2012). Embedding and automating conditional logics in classical higher-order logic. *Annals of Mathematics and Artificial Intelligence*, 66(1-4):257–271.

[33] Benzmüller, C. and Kohlhase, M. (1998). System description: Leo—a higher-order theorem prover. In *International Conference on Automated Deduction*, pages 139–143. Springer.

[34] Benzmüller, C. and Miller, D. (2014). Automation of higher-order logic. In Gabbay, D., Siekmann, J., and Woods, J., editors, *Handbook of the History of Logic, Volume 9 — Computational Logic*, pages 215–254. North Holland, Elsevier.

[35] Benzmüller, C. and Paleo, B. W. (2015). Higher-order modal logics: Automation and applications. In *Reasoning Web International Summer School*, pages 32–74. Springer.

[36] Benzmüller, C., Parent, X., and van der Torre, L. (2018c). A deontic logic reasoning infrastructure. In *Conference on Computability in Europe*, pages 60–69. Springer.

[37] Benzmüller, C., Parent, X., and van der Torre, L. (2020b). Designing normative theories for ethical and legal reasoning: LogiKEy framework, methodology, and tool support. *Artificial Intelligence*, 237:103348.

[38] Benzmüller, C. and Paulson, L. (2008). Exploring properties of normal multimodal logics in simple type theory with LEO-II. In Benzmüller, C., Brown, C., Siekmann, J., and Statman, R., editors, *Reasoning in Simple Type Theory — Festschrift in Honor of Peter B. Andrews on His 70th Birthday*, Studies in Logic, Mathematical Logic and Foundations, pages 386–406. College Publications. (Superseded by 2013 article in Logica Universalis).

[39] Benzmüller, C. and Paulson, L. (2010). Multimodal and intuitionistic logics in simple type theory. *Logic Journal of the IGPL*, 18(6):881–892.

[40] Benzmüller, C. and Paulson, L. (2013). Quantified multimodal logics in simple type theory. *Logica Universalis (Special Issue on Multimodal Logics)*, 7(1):7–20.

[41] Benzmüller, C., Sultana, N., Paulson, L. C., and Theiß, F. (2015). The higher-order prover LEO-II. *Journal of Automated Reasoning*, 55(4):389–404.

[42] Benzmüller, C. and Woltzenlogel Paleo, B. (2014). Automating Gödel's ontological proof of God's existence with higher-order automated theorem provers. In Schaub, T., Friedrich, G., and O'Sullivan, B., editors, *21st European Conference on Artificial Intelligence*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 93 – 98. IOS Press.

[43] Blackburn, P., van Benthem, J. F., and Wolter, F. (2006). *Handbook of Modal Logic*. Elsevier.

[44] Blanchette, J. and Nipkow, T. (2010). Nitpick: A counterexample generator for higher-order logic based on a relational model finder. In *International Conference on Interactive Theorem Proving*, number 6172 in Lecture Notes in Computer Science, pages 131–146. Springer.

[45] Blanchette, J. C., Böhme, S., and Paulson, L. C. (2013). Extending sledgehammer with SMT solvers. *Journal of Automated Reasoning*, 51(1):109–128.

[46] Blok, W. J. and Pigozzi, D. (1989). *Algebraizable logics*, volume 77. American Mathematical Society.

[47] Bochman, A. (2005). *Explanatory nonmonotonic reasoning*. World scientific.

[48] Bochman, A. (2013). *A logical theory of nonmonotonic inference and belief change*. Springer Science & Business Media.

[49] Boella, G. and Lesmo, L. (2002). A game theoretic approach to norms and agents. *Cognitive Science Quarterly*, 2(3-4):492–512.

[50] Boella, G. and van der Torre, L. (2003). Attributing mental attitudes to normative systems. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, volume 3, pages 942–943.

[51] Boella, G. and van der Torre, L. (2005). Constitutive norms in the design of normative multiagent systems. In *International Workshop on Computational Logic in Multi-Agent Systems*, pages 303–319. Springer.

[52] Boella, G. and van der Torre, L. (2006a). A logical architecture of a normative system. In *International Workshop on Deontic Logic and Artificial Normative Systems*, pages 24–35. Springer.

[53] Boella, G. and van der Torre, L. (2006b). Security policies for sharing knowledge in virtual communities. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, 36(3):439–450.

[54] Boella, G. and van der Torre, L. (2008). Institutions with a hierarchy of authorities in distributed dynamic environments. *Artificial Intelligence and Law*, 16(1):53–71.

[55] Boella, G., van der Torre, L., and Verhagen, H. (2006). Introduction to normative multiagent systems. *Computation and Mathematical Organizational Theory, Special issue on Normative Multiagent Systems*, 12(2-3):71–79.

[56] Boella, G. and van der Torre, L. W. (2004). Regulative and constitutive norms in normative multiagent systems. In *Procs. of 9th International Conference on the Principles of Knowledge Representation and Reasoning (KR 2004)*, volume 4, pages 255–265. AAAI Press.

[57] Bonevac, D. (1998). Against conditional obligation. *Noûs*, 32(1):37–53.

[58] Bratman, M. (1987). *Intention, plans, and practical reason*, volume 10. Harvard University Press Cambridge, MA.

[59] Bringsjord, S., Arkoudas, K., and Bello, P. (2006). Toward a general logicist methodology for engineering ethically correct robots. *IEEE Intelligent Systems*, 21(4):38–44.

[60] Broersen, J., Dastani, M., Hulstijn, J., Huang, Z., and van der, L. (2001). The BOID architecture: conflicts between beliefs, obligations, intentions and desires. In *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 9–16. ACM.

[61] Broersen, J., Dastani, M., Hulstijn, J., and van der Torre, L. (2002). Goal generation in the BOID architecture. *Cognitive Science Quarterly*, 2(3-4):428–447.

[62] Broersen, J. and van der Torre, L. (2011). Ten problems of deontic logic and normative reasoning in computer science. In *Lectures on Logic and Computation*, pages 55–88. Springer.

[63] Brown, C. E. (2004). *Set comprehension in Church's type theory.* PhD thesis, Department of Mathematical Sciences, Carnegie Mellon University. See also Chad E. Brown, *Automated Reasoning in Higher-Order Logic*, College Publications, 2007.

[64] Burgess, J. P. et al. (1981). Quick completeness proofs for some logics of conditionals. *Notre Dame Journal of Formal Logic*, 22(1):76–84.

[65] Cariani, F. (To appear). Deontic logic and natural language. In Gabbay, D., Horty, J., Parent, X., van der Meyden, R., and van der Torre, L., editors, *Handbook of Deontic Logic*, volume 2. College Publications.

[66] Carmo, J. and Jones, A. (2002). Deontic logic and contrary-to-duties. In Gabbay, D. M. and Guenthner, F., editors, *Handbook of Philosophical Logic: Volume 8*, pages 265–343. Springer Netherlands, Dordrecht.

[67] Carmo, J. and Jones, A. (2013). Completeness and decidability results for a logic of contrary-to-duty conditionals. *Journal of Logic and Computation*, 23(3):585–626.

[68] Carnielli, W., Coniglio, M. E., and van der Torre, L. (2009). *Input/output consequence relations: reasoning with intensional contexts*. Unpublished report.

[69] Castro, P. F. (2009). *Deontic action logics for specification and analysis of fault-tolerance*. PhD thesis, MacMaster University.

[70] Chellas, B. F. (1975). Basic conditional logic. *Journal of Philosophical Logic*, pages 133–153.

[71] Chellas, B. F. (1980). *Modal logic*. Cambridge university press.

[72] Chisholm, R. M. (1963). Contrary-to-duty imperatives and deontic logic. *Analysis*, 24:33–36.

[73] Chopra, A., van der Torre, L., Verhagen, H., and Villata, S. (2018). *Handbook of normative multiagent systems*. College Publications.

[74] Church, A. (1932). A set of postulates for the foundation of logic. *Annals of Mathematics*, 33(3):346–366.

[75] Church, A. (1940). A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5(2):56–68.

[76] Ciabattoni, A., Gulisano, F., and Lellmann, B. (2018). Resolving conflicting obligations in Mīmāṃsā: a sequent-based approach. In Broersen, J., Condoravdi, C., Nair, S., and Pigozzi, G., editors, *Deontic Logic and Normative Systems — 14th International Conference, DEON 2018, Utrecht, The Netherlands, 3-6 July, 2018*, pages 91–109.

[77] Ciuciura, J. (2013). Non-adjunctive discursive logic. *Bulletin of the Section of Logic*, 42(3/4):169–181.

[78] Costa, H. A. (2005). Non-adjunctive inference and classical modalities. *Journal of Philosophical Logic*, 34(5-6):581–605.

[79] Cruanes, S. and Blanchette, J. C. (2016). Extending nunchaku to dependent type theory. In Blanchette, J. C. and Kaliszyk, C., editors, *Proceedings First International Workshop on Hammers for Type Theories, HaTT@IJCAR 2016, Coimbra, Portugal, July 1, 2016*, volume 210 of *EPTCS*, pages 3–12.

[80] Danielsson, S. (1968). *Preference and obligation, studies in the logic of ethics*. PhD thesis, Filosofiska Färeningen.

[81] David, R. (1967). *The right and the good.* Oxford Clarendon Press.

[82] Davis, M. (2001). The early history of automated deduction: Dedicated to the memory of Hao Wang. In *Handbook of Automated Reasoning*, pages 3–15. Elsevier.

[83] Davis, M. and Putnam, H. (1960). A computing procedure for quantification theory. *Journal of the ACM (JACM)*, 7(3):201–215.

[84] Farjami, A., Meder, P., Parent, X., and Benzmüller, C. (2018). *I/O logic in HOL.* Preseanted in MIREL 2018, Workshop on MIning and REasoning with Legal texts.

[85] Feldman, F. (1986). *Doing the best we can: An essay in informal deontic logic*, volume 35 of *Philosophical Studies Series* . Dordrecht, Boston: D. Reidel Publishing Company.

[86] Føllesdal, D. and Hilpinen, R. (1970). Deontic logic: An introduction. In *Deontic Logic: Introductory and Systematic Readings*, pages 1–35. Springer.

[87] Font, J. M. and Jansana, R. (2017). *A general algebraic semantics for sentential logics.* Cambridge University Press.

[88] Forrester, J. W. (1984). Gentle murder, or the adverbial Samaritan. *The Journal of Philosophy*, 81(4):193–197.

[89] Frege, G. (1879). *Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens.* Halle.

[90] From, A. H. (2018). Epistemic logic. *Archive of Formal Proofs.*

[91] Fuenmayor, D. and Benzmüller, C. (2018). Formalisation and evaluation of Alan Gewirth's proof for the principle of generic consistency in Isabelle/HOL. *Archive of Formal Proofs.*

[92] Fuhrmann, A. (2017). Deontic modals: Why abandon the default approach. *Erkenntnis*, 82(6):1351–1365.

[93] Gabbay, D., Horty, J., Parent, X., van der Meyden, R., and van der Torre, L., editors (2013). *Handbook of Deontic Logic and Normative Systems.* College Publications.

[94] Gabbay, D., Parent, X., and van der Torre, L. (2019). A geometrical view of I/O logic. CoRR abs/1911.12837.

[95] Gelfond, M. and Lifschitz, V. (1988). The stable model semantics for logic programming. In Kowalski, R., Bowen, and Kenneth, editors, *Proceedings of International Logic Programming Conference and Symposium*, pages 1070–1080. MIT Press.

[96] Genesereth, M. R. and Nilsson, N. J. (2012). *Logical foundations of artificial intelligence*. Morgan Kaufmann.

[97] Gerdes, J. C. and Thornton, S. M. (2015). Implementable ethics for autonomous vehicles. In *Autonomes Fahren*, pages 87–102. Springer.

[98] Gibson, J. J. and Crooks, L. E. (1938). A theoretical field-analysis of automobile-driving. *The American Journal of Psychology*, 51(3):453–471.

[99] Goble, L. (1996). Ought'and extensionality. *Noûs*, 30(3):330–355.

[100] Goble, L. (2003). Preference semantics for deontic logic part I—simple models. *Logique et Analyse*, pages 383–418.

[101] Goble, L. (2004). Preference semantics for deontic logic part II–multiplex models. *Logique et Analyse*, pages 335–363.

[102] Goble, L. (2013). Prima facie norms, normative conflicts, and dilemmas. In Gabbay, D., Horty, J., Parent, X., van der Meyden, R., and van der Torre, L., editors, *Handbook of Deontic Logic*, volume 1, pages 499–544. College Publications.

[103] Gödel, K. (1931). Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte der Mathematischen Physik*, 38:173–198.

[104] Goffman, E. (1967). *Interaction ritual*. Penguin; Harmondsworth.

[105] Gonçalves, R. and Alferes, J. J. (2012). An embedding of input-output logic in deontic logic programs. In *International Conference on Deontic Logic in Computer Science*, pages 61–75. Springer.

[106] Gordon, M. J. and Melham, T. F. (1993). *Introduction to HOL: A theorem proving environment for higher order logic*. Cambridge University Press.

[107] Hansen, J. (2008). *Imperatives and deontic Logic: On the semantic foundations of deontic logic.* PhD thesis, University of Leipzig.

[108] Hansen, J. (2014). Reasoning about permission and obligation. In *David Makinson on Classical Methods for Non-classical Problems*, pages 287–333. Springer.

[109] Hansson, B. (1969). An analysis of some deontic logics. *Nous*, pages 373–398.

[110] Henkin, L. (1950). Completeness in the theory of types. *Journal of Symbolic Logic*, 15(2):81–91.

[111] Hintikka, J. (1957). *Quantifiers in deontic logic.* Societas Scientiarum Fennica.

[112] Horty, J. (2014). Deontic modals: Why abandon the classical semantics? *Pacific Philosophical Quarterly*, 95(4):424–460.

[113] Horty, J. F. (1997). Nonmonotonic foundations for deontic logic. In *Defeasible Deontic Logic*, pages 17–44. Springer.

[114] Horty, J. F. (2012). *Reasons as defaults.* Oxford University Press.

[115] Jansana, R. (2016). Algebraic propositional logic. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy.* Metaphysics Research Lab, Stanford University, winter 2016 edition.

[116] Jaśkowski, S. (1969). Propositional calculus for contradictory deductive systems (communicated at the meeting of march 19, 1948). *Studia Logica*, 24:143–160.

[117] Jones, A. J. (1990). Deontic logic and legal knowledge representation. *Ratio Juris*, 3(2):237–244.

[118] Jones, A. J. and Parent, X. (2007). A convention-based approach to agent communication languages. *Group Decision and Negotiation*, 16(2):101–141.

[119] Jones, A. J. and Sergot, M. (1992). Deontic logic in the representation of law: Towards a methodology. *Artificial Intelligence and Law*, 1(1):45–64.

[120] Jones, A. J. and Sergot, M. (1996). A formal characterisation of institutionalised power. *Logic Journal of the IGPL*, 4(3):427–443.

[121] Jörgensen, J. (1937). Imperatives and logic. *Erkenntnis*, 7(1):288–296.

[122] Kanger, S. (1970). New foundations for ethical theory. In *Deontic Logic: Introductory and Systematic Readings*, pages 36–58. Springer.

[123] Kanger, S. (1972). Law and logic. *Theoria*, 38(105-132).

[124] Kanger, S. and Kanger, H. (1966). Rights and parliamentarism. *Theoria*, 32(2):85–115.

[125] Kirchner, D., Benzmüller, C., and Zalta, E. N. (2020). Mechanizing principia logico-metaphysica in functional type theory. *The Review of Symbolic Logic*, 13(1):206–218.

[126] Kolodny, N. and MacFarlane, J. (2010). Ifs and oughts. *The Journal of philosophy*, 107(3):115–143.

[127] Kratzer, A. (1977). What 'must'and 'can'must and can mean. *Linguistics and Philosophy*, 1(3):337–355.

[128] Kratzer, A. (1981). The notional category of modality. *Words, Worlds, and Contexts: New Approaches in Word Semantics*, 6:38.

[129] Kratzer, A. (2012). *Modals and conditionals: New and revised perspectives*, volume 36. Oxford University Press.

[130] Kratzer, A., Pires de Oliveira, R., and Pessotto, A. L. (2014). Talking about modality: an interview with Angelika Kratzer. *ReVEL, especial*, (8).

[131] Lewis, D. (1973). *Counterfactuals.* Blackwell, Oxford.

[132] Lewis, D. (1974). Semantic analyses for dyadic deontic logic. In *Logical Theory and Aemantic Analysis*, pages 1–14. Springer.

[133] Lewis, D. (1981). Ordering semantics and premise semantics for counterfactuals. *Journal of Philosophical Logic*, 10(2):217–234.

[134] Libal, T. and Steen, A. (2020). NAI: Towards transparent and usable semi-automated legal analysis. *Verantwortungsbewusste Digitalisierung, Tagungsband des 23. Internationalen Rechtsinformatik Symposions IRIS 2020*, pages 265–272.

[135] Lindahl, L. (2012). *Position and change: a study in law and logic*, volume 112 of *Synthese Library.* Dordrecht: D. Reidel Publishing Co.

[136] Lindahl, L. and Odelstad, J. (2013). The theory of joining-systems. In Gabbay, D., Horty, J., Parent, X., van der Meyden, R., and van der Torre, L., editors, *Handbook of Deontic Logic*, volume 1, pages 545–634. College Publications.

[137] Mackenzie, D. (1995). The automation of proof: A historical and sociological exploration. *IEEE Annals of the History of Computing*, 17(3):7–29.

[138] Makinson, D. (1999). On a fundamental problem of deontic logic. *Norms, Logics and Information Systems. New Studies on Deontic Logic and Computer Science*, pages 29–54.

[139] Makinson, D. (2005). *Bridges from classical to nonmonotonic logic.* King's College.

[140] Makinson, D. and van der Torre, L. (2000). Input/output logics. *Journal of Philosophical Logic*, 29(4):383–408.

[141] Makinson, D. and van der Torre, L. (2001). Constraints for input/output logics. *Journal of Philosophical Logic*, 30(2):155–185.

[142] Makinson, D. and van der Torre, L. (2003). Permission from an input/output perspective. *Journal of Philosophical Logic*, 32(4):391–416.

[143] McCarty, L. T. (1976). Reflections on taxman: An experiment in artificial intelligence and legal reasoning. *Harvard Law Review*, 90:837.

[144] McCarty, L. T. (1983). Permissions and obligations. In *International Joint Conferences on Artificial Intelligence, American Association for Artificial Intelligence (IJCAI)*, volume 83, pages 287–294. Citeseer.

[145] McNamara, P. (1996). Doing well enough: Toward a logic for common-sense morality. *Studia Logica*, 57(1):167–192.

[146] McNamara, P. (2011). Supererogation, inside and out: Toward an adequate scheme for common sense morality. In Timmons, M., editor, *Oxford Studies in Normative Ethics, Volume I*, pages 202–235. Oxford University Press.

[147] McNamara, P. (2014). Deontic logic. In Zalta, E., editor, *The Stanford Encyclopedia of Philosophy.* Metaphysics Research Lab, Stanford University, winter 2014 edition.

[148] Meyer, J.-J. C. and van der Hoek., W., editors (1995). *Epistemic Logic for AI and Computer Science (Cambridge Tracts in Theoretical Computer Science).* Cambridge University Press.

[149] Mott, P. L. (1973). On Chisholm's paradox. *Journal of Philosophical Logic*, 2(2):197–211.

[150] Nagel, T. (1979). *Mortal Questions.* Cambridge University Press.

[151] Nair, S. (2014). Consequences of reasoning with conflicting obligations. *Mind*, 123(491):753–790.

[152] Navarro, P. E. and Rodríguez, J. L. (2014). *Deontic logic and legal systems.* Cambridge University Press.

[153] Negri, S. and Olivetti, N. (2015). A sequent calculus for preferential conditional logic based on neighbourhood semantics. In *International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, pages 115–134. Springer.

[154] Nelkin, D. K. (2019). Moral luck. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy.* Metaphysics Research Lab, Stanford University, summer 2019 edition.

[155] Niles, I. (1997). Rescuing the counterfactual solution to chisholm's paradox. *Philosophia*, 25(1-4):351–371.

[156] Nipkow, T., Paulson, L., and Wenzel, M. (2002). *Isabelle/HOL — A proof assistant for higher-order logic*, volume 2283 of *Lecture Notes in Computer Science.* Springer.

[157] Nute, D. (1997). *Defeasible deontic logic*, volume 263 of *Synthese Library.* Dordrecht: Kluwer Academic.

[158] Parent, X. (2008). On the strong completeness of Åqvist's dyadic deontic logic G. In *International Conference on Deontic Logic in Computer Science*, pages 189–202. Springer.

[159] Parent, X. (2010). A complete axiom set for Hansson's deontic logic DSDL2. *Logic Journal of the IGPL*, 18(3):422–429.

[160] Parent, X. (2014). Maximality vs. optimality in dyadic deontic logic. *Journal of Philosophical Logic*, 43(6):1101–1128.

[161] Parent, X. (2015). Completeness of Åqvist's systems E and F. *The Review of Symbolic Logic*, 8(1):164–177.

[162] Parent, X. (2017). A modal translation of an intuitionistic I/O operation. In *7th Workshop on Intuitionistic Modal Logic and Applications (IMLA 2017), organized by V. de Paiva and S. Artemov at the University of Toulouse (France)*, pages 17–28.

[163] Parent, X., Gabbay, D., and van der Torre, L. (2014). Intuitionistic basis for input/output logic. In *David Makinson on Classical Methods for Non-Classical Problems*, pages 263–286. Springer.

[164] Parent, X. and van der Torre, L. (2013). Input/output logic. In Gabbay, D., Horty, J., Parent, X., van der Meyden, R., and van der Torre, L., editors, *Handbook of Deontic Logic*, volume 1, pages 499–544. College Publications.

[165] Parent, X. and van der Torre, L. (2014). Sing and dance! In *International Conference on Deontic Logic in Computer Science*, pages 149–165. Springer.

[166] Parent, X. and van der Torre, L. (2017a). Detachment in normative systems: Examples, inference patterns, properties. *The IfCoLog Journal of Logics and their Applications, Special Issue "Logic for Normative Multi-Agent Systems" (Gest editors: G. Pigozzi and L. van der Torre)*, 4(9):2295–3039.

[167] Parent, X. and van der Torre, L. (2017b). The pragmatic oddity in norm-based deontic logics. In *Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law*, pages 169–178.

[168] Parent, X. and van der Torre, L. (2018a). *Introduction to deontic logic and normative systems.* College Publications.

[169] Parent, X. and van der Torre, L. (2018b). I/O logics with a consistency check. In Broersen, J., Condoravdi, C., Nair, S., and Pigozzi, G., editors, *Deontic Logic and Normative Systems — 14th International Conference, DEON 2018, Utrecht, The Netherlands, 3-6 July, 2018*, pages 285–299. College Publications.

[170] Pigozzi, G. and van der Torre, L. (2017). Multiagent deontic logic and its challenges from a normative systems perspective. *The IfCoLog Journal of Logics*

and their Applications, Special Issue "Logic for Normative Multi-Agent Systems" (Gest editors: G. Pigozzi and L. van der Torre), pages 2929–2993.

[171] Pollock, J. L. (1981). A refined theory of counterfactuals. *Journal of Philosophical Logic*, pages 239–266.

[172] Pörn, I. (1970). *The logic of power.* Blackwell, Oxford.

[173] Pörn, I. (1971). *Elements of social analysis.* Number 10. Filosofiska föreningen och Filosofiska institutionen vid Uppsala universitet.

[174] Prakken, H. and Sergot, M. (1996). Contrary-to-duty obligations. *Studia Logica*, 57(1):91–115.

[175] Prior, A. N. (1958). Escapism: The logical basis of ethics. In Melden, A. I., editor, *Essays in moral philosophy*, pages 135–146. University of Washington Press.

[176] Rahman, S., Granström, J. G., and Farjami, A. (2019). Legal reasoning and some logic after: All the lessons of the elders. In Gabbay, D., Magnani, L., Park, W., and Pietarinen, A.-V., editors, *Natural Arguments: A Tribute to John Woods.* College Publications.

[177] Rahman, S., Iqbal, M., and Soufi, Y. (2020). *Inferences by parallel reasoning in Islamic jurisprudence: Al-Shīrāzī's insights into the dialectical constitution of meaning and knowledge*, volume 19. Springer Nature.

[178] Rescher, N. (1958). An axiom system for deontic logic. *Philosophical Studies: An International Journal for Philosophy in the Analytic Tradition*, 9(1/2):24–30.

[179] Robaldo, L., Bartolini, C., Palmirani, M., Rossi, A., Martoni, M., and Lenzini, G. (2019). Formalizing GDPR provisions in reified I/O logic: the DAPRECO knowledge base. *Journal of Logic, Language and Information*, 29:401–449.

[180] Robinson, J. A. (1965). A machine-oriented logic based on the resolution principle. *Journal of the ACM (JACM)*, 12(1):23–41.

[181] Rönnedal, D. (2019). Contrary-to-duty paradoxes and counterfactual deontic logic. *Philosophia*, 47(4):1247–1282.

[182] Ross, A. (1944). Imperatives and logic. *Philosophy of Science*, 11(1):30–46.

[183] Royakkers, L. (1998). *Extending deontic logic for the formalisation of legal rules*, volume 36 of *Law and Philosophy Library.* Springer.

[184] Russell, B. (1908). Mathematical logic as based on the theory of types. *American Journal of Mathematics*, 30(3):222–262.

[185] Russell, S. J. and Norvig, P. (2016). *Artificial intelligence: a modern approach*. Pearson.

[186] Sadegh-Zadeh, K. (2011). *Handbook of Analytic Philosophy of Medicine*. Philosophy and Medicine. Springer.

[187] Searle, J. R., Willis, S., et al. (1995). *The construction of social reality*. Simon and Schuster.

[188] Spohn, W. (1975). An analysis of Hansson's dyadic deontic logic. *Journal of Philosophical Logic*, 4(2):237–252.

[189] Stalnaker, R. C. (1968). A theory of conditionals. In *Ifs*, pages 41–55. Springer.

[190] Stalnaker, R. C. and Thomason, R. H. (1970). A semantic analysis of conditional logic 1. *Theoria*, 36(1):23–42.

[191] Steen, A. (2020). Extensional paramodulation for higher-order logic and its effective implementation Leo-III. *KI-Künstliche Intelligenz*, 34(1):105–108.

[192] Stolpe, A. (2008a). Normative consequence: The problem of keeping it whilst giving it up. In *International Conference on Deontic Logic in Computer Science*, pages 174–188. Springer.

[193] Stolpe, A. (2008b). *Norms and norm-system dynamics*. PhD thesis, Department of Philosophy, University of Bergen, Norway.

[194] Stolpe, A. (2010a). Relevance, derogation and permission. In *International Conference on Deontic Logic in Computer Science*, pages 98–115. Springer.

[195] Stolpe, A. (2010b). A theory of permission based on the notion of derogation. *Journal of Applied Logic*, 8(1):97–113.

[196] Stolpe, A. (2015). A concept approach to input/output logic. *Journal of Applied Logic*, 13(3):239–258.

[197] Straßer, C. (2013). *Adaptive logics for defeasible reasoning: Applications in argumentation, normative reasoning and default reasoning*. Springer Publishing Company, Incorporated.

[198] Straßer, C., Beirlaen, M., and van de Putte, F. (2016). Adaptive logic characterizations of input/output logic. *Studia Logica*, 104(5):869–916.

[199] Sun, X. (2016). *Logic and games of norms: a computational perspective*. PhD thesis, University of Luxembourg, Luxembourg.

[200] Sun, X. (2018). Proof theory, semantics and algebra for normative systems. *Journal of logic and computation*, 28(8):1757–1779.

[201] Sun, X. and van der Torre, L. (2014). Combining constitutive and regulative norms in input/output logic. In *International Conference on Deontic Logic in Computer Science*, pages 241–257. Springer.

[202] Susskind, R. E. (2017). *Tomorrow's lawyers: An introduction to your future*. Oxford University Press.

[203] Tan, Y.-H. and van der Torre, L. (1996). How to combine ordering and minimizing in a deontic logic based on preferences. In *Deontic Logic, Agency and Normative Systems*, pages 216–232. Springer.

[204] Thomason, R. H. (1981). Deontic logic as founded on tense logic. In *New Studies in Deontic Logic*, pages 165–176. Springer.

[205] Tomberlin, J. E. (1981). Contrary-to-duty imperatives and conditional obligation. *Noûs*, pages 357–375.

[206] Tomberlin, J. E. (1989a). Deontic paradox and conditional obligation. *Philosophy and Phenomenological Research*, 50(1):107–114.

[207] Tomberlin, J. E. (1989b). Obligation, conditionals, and the logic of conditional obligation. *Philosophical Studies: An International Journal for Philosophy in the Analytic Tradition*, 55(1):81–92.

[208] Tosatto, S. C., Boella, G., van der Torre, L., and Villata, S. (2012). Abstract normative systems: Semantics and proof theory. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*, pages 358–368.

[209] van den Hoven, J. and Lokhorst, G.-J. (2002). Deontic logic and computer-supported computer ethics. *Metaphilosophy*, 33(3):376–386.

[210] van der Torre, L. (1997). *Reasoning about obligations: defeasibility in preference-based deontic logic.* PhD thesis, Erasmus University of Rotterdam.

[211] van der Torre, L. (2003). Contextual deontic logic: Normative agents, violations and independence. *Annals of mathematics and artificial intelligence*, 37(1-2):33–63.

[212] van der Torre, L. (2010). Violation games: a new foundation for deontic logic. *Journal of Applied Non-Classical Logics*, 20(4):457–477.

[213] van der Torre, L. and Tan, Y.-H. (1999). Contrary-to-duty reasoning with preference-based dyadic obligations. *Annals of Mathematics and Artificial Intelligence*, 27(1-4):49–78.

[214] van der Torre, L. and Tan, Y.-H. (2000). Two-phase deontic logic. *Logique et Analyse*, pages 411–456.

[215] van Elst, L., Dignum, V., and Abecker, A. (2004). Towards agent-mediated knowledge management. In *Agent-mediated Knowledge Management*, pages 1–30. Springer.

[216] van Fraassen, B. C. (1972). The logic of conditional obligation. *Journal of Philosophical Logic*, 1:417–438.

[217] van Fraassen, B. C. (1973). Values and the heart's command. *The Journal of Philosophy*, 70(1):5–19.

[218] von Fintel, K. (2012). The best we can (expect to) get? challenges to the classic semantics for deontic modals. In *Central Meeting of the American Philosophical Association*, volume 17.

[219] von Fintel, K. and Heim, I. (2011). Intensional semantics. *Unpublished Lecture Notes*.

[220] von Wright, G. H. (1951). Deontic logic. *Mind*, 60(237):1–15.

[221] von Wright, G. H. (1963). *Norm and action: a logical enquiry.* Humanities.

[222] von Wright, G. H. (1968). *An essay in deontic logic and the general theory of action.* North-Holland Publishing Company.

[223] von Wright, G. H. (1970). A new system of deontic logic. In *Deontic Logic: Introductory and Systematic Readings*, pages 105–120. Springer.

[224] Wallach, W. and Allen, C. (2008). *Moral machines: Teaching robots right from wrong*. Oxford University Press.

[225] Whitehead, A. N. and Russell, B. (1912). *Principia mathematica*, volume 2. University Press.

[226] Zach, R. (2019). Hilbert's program. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, fall 2019 edition.

[227] Zalta, E. N. (2016). Principia logico-metaphysica. *Draft version, preprint available at https://mally. stanford. edu/principia. pdf.*