FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Formal Reasoning about Fuzzy Evaluation Games

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Computational Intelligence

eingereicht von

Stoiko Ivanov, BSc
Matrikelnummer 00225612

an der Fakultät für Informatik
der Technischen Universität Wien

Betreuung: Ao. Prof. DI. Dr. Christian G. Fermüller

Wien, 23. April 2018

Stoiko Ivanov	Christian G. Fermüller

Formal Reasoning about Fuzzy Evaluation Games

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Computational Intelligence

by

Stoiko Ivanov, BSc
Registration Number 00225612

to the Faculty of Informatics
at the TU Wien

Advisor: Ao. Prof. DI. Dr. Christian G. Fermüller

Vienna, 23rd April, 2018

Stoiko Ivanov

Christian G. Fermüller

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.ac.at

Erklärung zur Verfassung der Arbeit

Stoiko Ivanov, BSc
Franzensgasse 6/8, 1050 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 23. April 2018

Stoiko Ivanov

Acknowledgements

I am grateful to my parents, Dora and Dentscho, for the continuous and unconditional support during my studies. Thank you for giving me the opportunity to enjoy a rewarding and unencumbered student's life!

I thank Chris for supporting and encouraging me throughout the long duration of this thesis. I consider myself lucky for having such a positive and appreciative advisor!

To the brilliant people, who have accompanied me during the writing of this work, listened to my ideas, discussed them with me, and never cease to amaze me with their positive attitude — Thank you!

I like to dedicate this thesis to two scientists whose work has fundamentally inspired it, and who passed away before its completion: Petr Hájek and Jaakko Hintikka.

Kurzfassung

Logische Auswertungsspiele stellen eine alternative Beschreibung von Wahrheit zu Tarskisemantik dar. Wahrheit aus der Perspektive formaler Spieltheorie zu betrachten führte zu einigen interessanten Erweiterungen klassischer Logik, wie beispielsweise Independence-Friendly Logic von Hintikka, die Wahrheitswertermittlung zu einem Spiel unvollständiger Information macht. Auswertungsspiele nach Hintikka sind zu einem Standardwerkzeug zur Analyse neuer Logiken geworden.

Mathematische Fuzzy Logik ist ein System, um vage Aussagen zu modellieren. Ihr Ursprung liegt in den Werken von Łukasiewicz und Gödel. In der zweiten Hälfte des zwanzigsten Jahrhunderts wurde der Begriff Fuzzy Logic von Zadeh für einen Formalismus in der Steuerungsautomation geprägt. Diese zwei Ansätze wurden durch Hájek zusammengeführt und sind seither ein lebendiges Forschungsgebiet.

Es werden Beschreibungen von Auswertungsspielen, sowohl für klassische als auch für prominente Fuzzy Logiken, mit Definition aus formaler Spieltheorie verglichen. Es zeigen sich Diskrepanzen zwischen den Auswertungsspielen und den formalen Definitionen von Spielen in extensiver Form vollständiger Information. Durch die Darstellung der Regel für Negation als regulären Zug einEr SpielerIn werden die Definitionen unifiziert. Die präzisen Definitionen für zwei Spiele: je eines für klassische und eines für Zadehs Fuzzy Logik, mit Äquivalenzbeweisen zur Standardsemantik, sind ein zentrales Resultat dieser Diplomarbeit.

Die Analyse spieltheoretischer Konzepte mit Hilfe formaler Logik, insbesondere Modallogik, stellt ein verhältnismäßig neues Gebiet dar. Sie hat sehr ausdrucksstarke Formalismen, wie beispielsweise Game Logic hervorgebracht, mit welcher die Effektivität von SpielerInnen untersucht werden kann. Die hier vorgenommene Untersuchung der Unschärfen führt zu einer prägnanten Weiterführung der Formalisierung der Spiele: Eine Axiomatisierung der Spielbäume mit Modallogik wird erstellt, welche die Bäume als Kripke-Strukturen auffasst. Ausführliche Beweise, die belegen, dass die Axiome die Bäume tatsächlich beschreiben, werden erarbeitet. Eine gründliche Analyse von Prädikatenmodallogik und modaler Korrespondenztheorie dient als Basis, um möglichst viele Eigenschaften syntaktisch zu beschreiben.

Die vorliegende Axiomatisierung stellt eine mögliche Grundlage dar, um die Klasse der logischen Auswertungsspiele in ausdrucksstärkeren Formalismen zu untersuchen.

Abstract

Logical evaluation games are an alternative characterization of truth to standard Tarskian semantics. Analyzing truth with the apparatus of formal game theory has led to some interesting generalizations of classical logic, like Independence-friendly logic by Hintikka, based on making evaluation a game of imperfect information. Hintikka-style evaluation games have become standard means of analyzing new logics.

Mathematical fuzzy logic provides a formalism for reasoning about vague statements. Its roots can be traced back to works by Łukasiewicz and Gödel. In the second half of the twentieth century Zadeh coined the term fuzzy logic for a formalism used in automation. The approaches were unified in Hájek's framework and have been an active research topic since.

A close alignment of evaluation games, for classical and prominent fuzzy logics, with formal game theory reveals a gap between the presentation of logical evaluation games and formal definitions of extensive games of perfect information. This work joins the two notions by providing an explicit game rule of negation as an in-game move and aligning the definitions. The two resulting games, one for classical logic and one for Zadeh's fuzzy logic, along with correspondence proofs to standard semantics are a central result of this thesis.

The direction of analyzing game theoretic concepts with formal logic, especially modal logic is a comparatively new field and has yielded highly expressive formalisms, like Game Logic, for reasoning about players' powers for certain game situations. Examining the minor imprecisions in evaluation game definitions in literature led us to the idea of formalizing the games a distinctive step further: we construct modal axiomatizations of the game trees, by viewing them as Kripke structures and carry out extensive formal proofs showing that those axioms describe the games. A thorough analysis of first-order modal logic and modal correspondence theory is carried out in order to capture as many aspects as possible syntactically.

The provided axiomatization may serve as a base for concretely analyzing the class of logical evaluation games in richer formal systems.

Contents

CHAPTER 1

Introduction

1.1 Motivation

The connection between logic and games can be traced back to Aristotle [Hod13]. It was rediscovered and led to a very fruitful exchange between the fields of game theory and formal logic since the middle of the twentieth century, starting with the work of Lorenzen in 1955 [Kei11]. Initially interest was focused on analyzing logic notions with game theory:

> Many logical notions can be cast very naturally as two-player games. Examples are argumentation between a defender and critic of a claim (Lorenzen games), model comparison between people disputing an analogy (Ehrenfeucht-Fraïssé games), and perhaps most basically of all, semantical evaluation of assertions made with respect to some given situation.... [vB03]

The shift of perspective can lead to a better intuitive understanding of results in logic. For instance, the fact that propositional modal logic is the fragment of first-order logic, that is invariant for bisimulation [BvBW07] is more evident when the difference is expressed as what object a player picks when playing an evaluation game — an accessible world in the modal logic case, and an object of an arbitrary domain for first-order logic [vB03].

The other direction of looking at formal games with the formalisms of logic is a comparatively recent development and presents us with a wide field for studying new results. Especially the link with modal logic has led to exciting insights by describing Nash equilibria and players' powers with multi-modal logics [HMvdHW03], or using epistemic logics to characterize games of imperfect information [BvBW07].

1.2 Problem Statement and Aim of the Work

Originally this thesis set out to analyze several logical evaluation games[1] with the framework of game logic as introduced by van Benthem, Parikh and Pauly [vB02, Par85, PP03].

During the initial investigation, we encountered a few minor imprecisions, in the presentations of the evaluation games, that needed addressing and a resolution. This led to our current exploration of joining the original presentations of the games with the formal definitions of basic game theory. Additionally we took the formalization one distinctive step further, by axiomatizing the game rules for our version of the evaluation games in modal logic.

We provide mathematical proofs for the correspondence theorems between the extensive games introduced and the standard notion of truth. Additionally we formally show that our modal axiomatizations describe the evaluation game trees.

1.3 Methodological Approach

This thesis has a strong theoretical focus and is rooted in the fields of mathematical fuzzy logic, game theory and modal logic as a formalization tool.

The nature of the work as a refinement of the definitions of logical evaluation games compels us to pay particular attention to the formal presentation of notions and their accuracy and adequateness.

A challenge with this approach lies in not losing track of, or artificially hiding the intuitive ideas we base the work on. We address the challenge by providing instructive examples with ample explanations on the one hand, and visual representations for clarification on the other hand.

Addressing the issue of gender neutral language, we fully agree with Osborne's view laid out in the preface of *A Course in Game Theory* [OR94], that no language is neutral, and would prefer to use 'she' as generic pronoun. In accordance with regulatory requirements, however, the thesis uses 'they' as generic pronoun, apart from the descriptions used to talk about two player games. Player 1 will be addressed as 'she' and player 2 as 'he'.

1.4 Outline

The thesis is structured into four main chapters and aims to be self-contained for readers with an undergraduate background in mathematical logic.

In chapter 2 we introduce the two fields this worked is aligned between: logic and game theory. We start by presenting all formal logics used in a structured manner, by first presenting the logic on a propositional level, and augmenting to first-order level, where needed. Apart from providing the necessary information, this composition serves to unify the notation found in seminal works and textbooks, that we use as a starting point.

[1] Hintikka's original game for classical logic, a version extended to fuzzy logic in the broad sense and ultimately an evaluation game for Łukasiewicz logic

The second part of the chapter contains the necessary background in game theory to describe the main focus of our treatment: logical evaluation games. A summary of central works on logical evaluation games completes the chapter.

Chapter 3 describes and formally defines two logical evaluation games: $\mathcal{H}$-game for classical propositional logic and $\mathcal{H}$-mv-game for weak Łukasiewicz logic. The definitions are in accordance, with the basic game-theoretical notions introduced in chapter 2, and additionally provide a novel intuition behind the game rule for negation. The equivalence of the games to standard Tarski style semantics is shown for both games. Additionally the formalization is accompanied by graphical depictions to aid intuition.

The game trees introduced are taken as base for Kripke frames, and a modal logic axiomatization of the games in chapter 4. The correspondence between abstract logical axiomatization and formal game trees is proved, while reflecting on the implicit assumptions present in the formal game trees.

Finally chapter 5 summarizes the central results of this thesis, and presents input for further research in this area.

CHAPTER 2

State of the Art

This work connects the presentation of logical evaluation games in the literature to the principal definitions of basic game theory. Additionally it models the game trees with modal logic axioms.

This chapter serves two main purposes: On the one hand it gives a thorough introduction to the logics used throughout this work and defines central notions of our topic along with their formal notation. On the other hand we provide an overview of previous work in the field of logical evaluation games.

We use logic on two distinct levels in what follows. First the evaluation games have logical formulas as their objects of discourse. Second we use first-order modal logic to model the game trees of these games. Logics are introduced in their propositional version and extended to their first-order counterpart where needed.

The second part of this chapter introduces game theory, with a focus on extensive games of perfect information, the class that our logical evaluation games belong to.

Finally the chapter provides an overview of landmark publications in the field of logical evaluation games.

2.1 Classical Logic

2.1.1 Propositional Classical Logic

One common aspect in most definitions of logic, is that it deals with language [Sha13, BvBW07], be it informal or formal. We start with the syntax, the language, of propositional logic. The central objects of logic are *formulas* that, in the propositional case, consist of *propositional variables* ($p, q, p_1, p_2, \ldots$), of two *propositional constants* ($\top$, $\bot$), and of *logical connectives* $\rightarrow, \neg, \leftrightarrow, \vee, \wedge, \ \&$, representing implication, negation, equivalence, disjunction and two forms of conjunction respectively[1]. Additionally we use *auxiliary symbols* like parentheses.

[1]In some non-classical logics, like linear logic or fuzzy logic, there is more than one variant of conjunction and disjunction, which is reflected by the exclusion of (some of) the structural rules in their calculi. One of the connectives

Definition 2.1.1 (syntax of propositional logic)**.** The set of all *propositional formulas* $\mathcal{Prop}$ is inductively defined as follows:

- Propositional variables $p, q, p_1, p_2, \ldots$ and constants $\top, \bot$ are propositional formulas.
- Let φ and ψ be propositional formulas, then $(\varphi \rightarrow \psi)$, $(\neg\varphi)$, $(\varphi \leftrightarrow \psi)$, $(\varphi \vee \psi)$, $(\varphi \wedge \psi)$, $(\varphi \ \&\ \psi)$ are propositional formulas.
- These are all propositional formulas.

We may omit the parentheses, if it is possible to do so unambiguously, according to the usual conventions regarding priority and associativity of the connectives[2]

If a formula is a propositional variable or constant, we call it an *atomic formula* , or propositional *atom*. A formula is *compound*, only if it contains at least one connective. The set of atomic propositional formulas is denoted by $\mathcal{Atom}$, the set of arbitrary propositional formulas by $\mathcal{Prop}$. We denote atomic formulas by $a, a_1, a_2, \ldots$, if we want them to possibly include the constants $\top, \bot$.

Some proofs in this work use induction on the number of connectives occurring in a formula. We call this measure the *complexity* or *depth* of the formula. The term *depth* is best understood in a visual context — if we draw the formula as a tree its complexity equals the depth of the tree.

Definition 2.1.2 (complexity of a formula)**.** Given a formula φ we inductively define its *complexity* $comp(\varphi)$ as follows:

- If $\varphi \in \mathcal{Atom}$: $comp(\varphi) = 0$
- $\varphi = \neg\psi$: $comp(\varphi) = 1 + comp(\psi)$
- $\varphi = \psi_1 \circ \psi_2$ for $\circ \in \{\wedge, \ \&\ , \vee, \rightarrow, \leftrightarrow\}$: $comp(\varphi) = 1 + \max(comp(\psi_1), comp(\psi_2))$

Two terms for syntactic concepts, seldom used explicitly for propositional logic, are signature and language. We introduce them formally, to highlight what they correspond to for the syntactically more complex logics we discuss later.

Definition 2.1.3 (signature and language)**.** The set of syntactically correct formulas is called the *language*.

A particular set of propositional variables $p, q, p_1, q_1, \ldots$ fixed in a specific context is called a *signature*. Given a *signature*, we refer to all correct formulas, using only atoms from that signature as the *language over the signature*.

For propositional logic, without any restriction on the propositional variables, this is $\mathcal{Prop}$.

is idempotent, while the other one is in general not, thus for instance the classical equivalence $\varphi\&\varphi \leftrightarrow \varphi$ does not hold in most fuzzy logics; see section 3.1 of [CHN11] — in classical logic they coincide.

[2]Negation has precedence over conjunction and disjunction, which have precedence over implication, which has precedence over equivalence. Conjunction and disjunction are left-associative, implication and equivalence are right-associative.

The propositional variables represent statements, that can identified with a truth value, like for example "Vienna is the capital of Austria", or "Two is the only even prime number". In classical logic we assign the two absolute truth values *true* and *false* to propositions. The truth values are often identified with 1 and 0 respectively. Propositional logic analyzes the composition of such atomic propositions through logical connectives; it has a focus on *tautologies*, *satisfiable formulas* and *contradictions* — meaning statements which are always true, sometimes true or never true, depending on the truth values of the contained atoms. A *valuation* assigns truth values to formulas. The function assigning truth values to atomic formulas is called an *atomic valuation*. We make two choices for our atomic valuation:

- We include the truth constants $\top, \bot$ in it, with $\top$ and $\bot$ always evaluating to true and false respectively.
- We make a notational distinction between the atomic valuation ν_{CL}, and the one for compound formulas ν^*_{CL}.

This deviates from most definitions of the topic where the propositional constants are introduced as 0-ary connectives.

Definition 2.1.4 (classical valuation)**.** An atomic classical valuation ν_{CL} is a mapping from $\mathcal{Atom}$ to $\{0, 1\}$, s.t.

- $\nu_{CL}(\bot) = 0$
- $\nu_{CL}(\top) = 1$
- $\nu_{CL}(p) \in \{0, 1\}$, for propositional variables p

We call p *true* (under ν_{CL}) if $\nu_{CL}(p) = 1$ and *false* if $\nu_{CL}(p) = 0$.

For giving arbitrary formulas their value we extended ν_{CL} to a valuation for compound formulas ν^*_{CL}, in the following way:

- $\nu^*_{CL}(\varphi) = \nu_{CL}(\varphi)$, iff $\varphi \in \mathcal{Atom}$
- $\nu^*_{CL}(\varphi \wedge \psi) = 1$, iff $\nu^*_{CL}(\varphi) = 1$ and $\nu^*_{CL}(\psi) = 1$
- $\nu^*_{CL}(\varphi \vee \psi) = 1$, iff $\nu^*_{CL}(\varphi) = 1$ or $\nu^*_{CL}(\psi) = 1$
- $\nu^*_{CL}(\neg\varphi) = 1$, iff $\nu^*_{CL}(\varphi) = 0$
- $\nu^*_{CL}(\varphi \to \psi) = 1$, iff $\nu^*_{CL}(\varphi) = 0$ or $\nu^*_{CL}(\psi) = 1$
- $\nu^*_{CL}(\varphi \leftrightarrow \psi) = 1$, iff $\nu^*_{CL}(\varphi) = \nu^*_{CL}(\psi)$

In general atomic (ν_{CL}) and compound (ν^*_{CL}) valuations are not defined separately. In this work however it is instructive to make this distinction, since logical evaluation games are an alternative characterization of the truth functions for the connectives. Both refer to a ν_{CL} for evaluating a formula. We explicitly distinguish between ν_{CL} and ν^*_{CL} when dealing with the evaluation games, but take some liberties in the used notation where the distinction is not fundamental. The compound valuation is replaced by the evaluation games. This is the reason for including $\top$ and $\bot$ in the atomic valuation.

Observe that in the definition of ν^*_{CL}, the right hand sides only refer to the evaluations of the subformulas — the truth value of a compound formula depends only on the truth value of its subformulas. This principle is referred to as *truth-functionality*, since truth-functional connectives are in fact *functions*.

This is best visualized by the use of *truth tables* for defining the semantics of classical connectives. Given that their domain is $\{0, 1\}$, truth tables are an exhaustive definition of a function. An example is the truth table for implication in Table 2.1.

φ	ψ	$\varphi \to \psi$
0	0	1
0	1	1
1	0	0
1	1	1

Table 2.1: Truth table for implication

The propositional language with seven connectives above, is tailored towards expressing various semantic relations between propositions succinctly. When reasoning formally we want to be able to express concepts like conjunctions, disjunctions, and, probably above all, implications in a direct way. This makes the language redundant: all possible functions on the two classical truth-values may be expressed with a subset of the connectives defined above[3]. Each subset which can express all functions for a given logic is called *functionally complete* for this logic. For example the *fragment* of the propositional language only using $\neg, \wedge, \vee$, which we work with in our games is functionally complete for classical propositional logic. Just observe that $\varphi \to \psi$ and $\neg\varphi \vee \psi$, are logically equivalent, and that both constants can be expressed by a tautology and a contradiction.

Actually only one of $\wedge$ or $\vee$ is needed in addition to $\neg$, when taking into account De Morgan's laws. Using both in the evaluation game, serves as an example of both players making moves, akin to the way they choose objects from the domain when considering the quantifier rules in the first-order case, which is analogous to expressing existential quantification as (possibly infinitely) iterated disjunction, and universal quantification as iterated conjunction.

2.1.2 First-Order Classical Logic

Propositional logic lacks structure [Háj98]. In many situations it is an inadequately coarse tool for modeling. We want to talk about properties and relations of objects of our world, without introducing a propositional variable for each such statement. The notion of a predicate — a function from the domain of discourse to a truth value — is the generalization, that leads us to first-order logic. Properties of objects are unary predicates, relations between k objects are k-ary

[3]There are connectives, like nand and nor, which by themselves are functionally complete

predicates. Our example from above: “Two is the only even prime number”, could be modeled with two unary predicates: P and Q representing the properties “is an even number” and “is prime” respectively. The statement $\exists x(P(x) \wedge Q(x))$ then states: “There is an even number which is also prime”. In order to express the statement from above, we additionally need equality for saying that there is only one such number: $\forall x \forall y((P(x) \wedge Q(x)) \wedge (P(y) \wedge Q(y)) \rightarrow x = y)$.

What follows is a basic presentation of first-order logic with equality, as can be found in most text books on the subject. This particular version is primarily inspired by Leitsch's textbook on the resolution calculus [Lei97], the book on first-order modal logic by Fitting and Mendelsohn [FM98], and, to a lesser degree, by the Handbook of Modal Logic [BvBW07].

We assume that we are given countably infinite sets of variables VS, function symbols FS and predicate symbols PS. $FS = \bigcup_{k=0}^{\infty} FS_k$ and $PS = \bigcup_{k=0}^{\infty} PS_k$. Each FS_k and PS_k are sets of k-ary function and predicate symbols respectively. We call 0-ary function symbols *constants* and 0-ary predicate symbols *propositions*. We denote variables by $x, y, z, x_1, \ldots$, k-ary function symbols by $f^k, g^k, f_1^k, \ldots$ and k-ary predicate symbols by $P^k, Q^k, P_1^k, \ldots$. We omit the arities of function or predicate symbols, if they are clear from the context.

In the first-order case the *signature* (Definition 2.1.3) refers to a set of atomic predicates, including their arity. In addition it also determines the used function symbols.

On the syntactic level objects from the domain are represented by *terms*:

Definition 2.1.5 (term)**.** The set of all *terms* T is defined inductively by:

- $VS \in T$
- If $t_1, \ldots, t_n \in T$ and $f \in FS_n$ then $f(t_1, \ldots, t_n) \in T$

A *term* $t \in T$ is called *ground* iff it contains no variables.

In order to facilitate comparison of the various logics we present in this work we introduce their syntax with the following recursive notation:

Definition 2.1.6 (syntax of first-order logic)**.**

$$\varphi := \quad P(t_1, \ldots, t_n) \mid t = s \mid \top \mid \bot \mid \neg\varphi \mid (\varphi \circ \psi) \mid \forall x \varphi \mid \exists x \varphi$$

with $\circ \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$, $P \in PS_n$, $t, s, t_1, \ldots, t_n \in T$, and $x \in VS$

Variables are classified, according to whether they are *bound* by a quantifier, or whether they are *free* — in the formula $\forall x P(x, y)$, x is bound and y is free:

Definition 2.1.7 (free and bound occurrences of variables)**.** The set of free variables in a formula φ is denoted by $FV(\varphi)$ and defined as follows:

- φ is atomic, i.e. of the form $P(t_1, \ldots, t_n)$: $FV(\varphi)$ contains all variables in $(t_1, \ldots, t_n)$.
- $\neg\varphi$: $FV(\neg\varphi)$=$FV(\varphi)$.
- $\varphi \circ \psi$, for $\circ \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$: $FV(\varphi \circ \psi)$= $FV(\varphi) \cup FV(\psi)$.

- $\forall x\varphi$: $FV(\forall x\varphi)$= $FV(\varphi)\backslash\{x\}$.
- $\exists x\varphi$: $FV(\exists x\varphi)$= $FV(\varphi)\backslash\{x\}$.

A formula without free variables is called *sentence* or *closed formula*. If we call a formula, with free variables a sentence, we mean the universal closure of that formula, i.e. for the formula φ with $FV(\varphi) = \{x_1, \ldots, x_n\}$ we mean the formula $\forall x_1, \ldots, \forall x_n \varphi$.

On the semantic side we need to assign truth-values to atomic formulas like $P(x)$ Thus we need to know what P stands for and what x could represent. The structure, which links syntactic predicate and function symbols, as well as providing objects to be represented by variables in a given context is called a *model*:

Definition 2.1.8 (first-order model)**.** A *first-order model* is a pair $M = \langle D, V \rangle$, where:

1. D is a non-empty set, called the *domain*.
2. V, called the *interpretation*[4], is a mapping defined on $FS \cup PS$ s.t.:
   - For $f \in FS_k$, $V(f)$ is a k-ary function over D — $V(f) : D^k \to D$.
   - For $P \in PS_k$, $V(P)$ is a k-ary predicate or relation over D — $V(P) : D^k \to \{0, 1\}$.

In general[5] both D and V are needed to analyze whether a first-order formula is true or false w.r.t. to them. Take for example the formula φ as $\neg\exists x P(x)$: assume we set $D = \mathbb{N}$, and let $V(P)$ be "is less then 0". In this model φ evaluates to true. However if we change D to represent all integers, the formula is false.

There are several ways to approach equality in first-order logic. Usually it is treated as a special predicate symbol. In most calculi rules are added for the reflexivity of equality $x = x$ and a version of the *substitution principle* as a schema: $x = y \to (\varphi(x) \leftrightarrow \varphi(y))$ (φ standing for an arbitrary formula). From these two principles the symmetry and transitivity of equality can be deduced, however we can not force the $=$ symbol to be the equality relation. We get an equivalence relation where each equivalence class contains objects which cannot be distinguished by the predicates in the given signature. For example two predicates identifying their arguments as plants and animals, say $P(x)$ and $A(x)$ cannot distinguish between different plants — if we define equality by the rules above a tulip would be equal to a tree. Another detail is that the axiomatization of equality is not finite, since the substitution principle is expressed by an axiom schema. Equality is finitely axiomatizable in second-order logic by the single formula $x = y \leftrightarrow \forall X(X(x) \leftrightarrow X(y))$. To bypass the specifics of equality in first-order logic, its definition is made on the semantic side. A *normal model* is a model in which $=$ is interpreted as true equality, the equivalence relation, where every class consists of a single element. All models in used in this work are normal models, unless stated otherwise.

In order to arrive at a definition of truth for first-order logic, it is first necessary to give a semantic meaning to variables and terms. Clearly we want them to designate objects from the *domain*:

[4]The generalization of a basic valuation ν_{CL} on the propositional level (Definition 2.1.4)

[5]of course there are formulas which are true or false in any model, e.g. $A(x) \vee \neg A(x)$ and $A(x) \wedge \neg A(x)$ respectively.

Definition 2.1.9 (variable valuation)**.** Given a first-order model $M = \langle D, V \rangle$ a variable valuation I is a mapping that assigns to each $x \in VS$ an element of the domain: $I : VS \to D$.

Definition 2.1.10 (variable variant)**.** Let I be a *variable valuation*. By $I[x \mapsto d]$ we mean the variable valuation, which is identical to I, save possibly for the value of x, which is d:

$$I[x \mapsto d](y) = \begin{cases} d \text{ if } y = x \\ I(y) \text{ if } y \neq x \end{cases}$$

When comparing two given variable valuations I and J, which differ only on x we call J an *x-variant* of I (and vice versa) and write $I \sim_x J$.

Since terms contain both function symbols (including constants), and variables, both the *interpretation* of the model as well as the *variable valuation* partake in semantically interpreting them:

Definition 2.1.11 (term valuation)**.** Given a *model* $M = \langle D, V \rangle$ and a *variable valuation* I, we define the *term valuation function* $TV : T \to D$ recursively as follows:

1. If $x \in VS$, then $TV(x) = I(x)$
2. If $f(t_1, \ldots, t_n) \in T$, then $TV(f(t_1, \ldots, t_n)) = V(f)(TV(t_1), \ldots, TV(t_n))$

The term valuation of *constants* $c \in FS_0$, as a special case of function symbols, is written as $TV(c) = V(c)$.

While it is possible to express the valuation for first-order logic as a function from formulas to truth values, like ν^*_{CL} for propositional logic, it is customary to define first-order satisfiability in a model by a relation $\models$, from models and variable valuations to formulas. If a formula φ is true in a model M with a variable valuation I, the relation holds — we write $M, I \models \varphi$:

Definition 2.1.12 (first-order satisfiability)**.** Let φ, ψ be first-order formulas, $t_1, \ldots, t_n \in T$, $P \in PS_n$, $M = \langle D, V \rangle$ a *first-order model*, I a *variable valuation* and TV the *term valuation function*. We define the *first-order satisfiability relation* $\models$ as:

1. $M, I \models P(t_1, \ldots, t_n)$, iff $V(P)(TV(t_1), \ldots, TV(t_n)) = 1$
2. $M, I \models s = t$, iff $TV(s) = TV(t)$
3. $M, I \models \forall x \varphi$, iff for all $d' \in D$ $M, I[x \mapsto d'] \models \varphi$
4. $M, I \models \exists x \varphi$, iff there exists a $d' \in D$, s.t. $M, I[x \mapsto d'] \models \varphi$
5. $M, I \models \top$
6. $M, I \not\models \bot$ [6]

[6] We write the negation of $\models$ as $\not\models$

7. $M, I \models \neg\varphi$, iff $M, I \not\models \varphi$
8. $M, I \models \varphi \wedge \psi$, iff $M, I \models \varphi$ and $M, I \models \psi$
9. $M, I \models \varphi \vee \psi$, iff $M, I \models \varphi$ or $M, I \models \psi$
10. $M, I \models \varphi \rightarrow \psi$, iff $M, I \not\models \varphi$ or $M, I \models \psi$
11. $M, I \models \varphi \leftrightarrow \psi$, iff $M, I \models \varphi \rightarrow \psi$ and $M, I \models \psi \rightarrow \varphi$[7]

Note that the variable valuation plays only a role for evaluating *free variables*. For variables, that are quantified over, $\models$ refers to *variants* of the given variable valuation: for existential quantification, a variant which makes the formula true has to exist, for universal quantification all possible variants have to make the formula true. For a *sentence* this means that we never refer to the particular domain elements which are mapped to the variables. In case we do not refer to a particular valuation we may omit I in the satisfiability relation and write $M \models \forall x P(x)$.

2.2 Fuzzy Logics

Where first-order classical logic provides us with an apparatus to reason about various domains of discourse, fuzzy logic can be used as a tool to model and reason about vague or imprecise statements, which frequently occur in natural language. The last sentence is already a good example of this: what does it mean when we say that imprecise statements occur frequently? — in a classical setting we would only have the choice of saying that the statement is absolutely true or false, however, it stands to reason, that the frequency of “imprecise” statements depends very much on context (who states those statements, in which context are they stated). If we want to assign a singular truth value to such a global sentence, we would say that it is neither absolutely true nor absolutely false – it is true to a certain degree. we have already identified the absolute truth values true and false with 0 and 1, making values from the real unit-interval $[0, 1]$ an obvious choice for intermediate truths.

We restrict our attention to a treatment of propositional fuzzy logics in this work for the extension to a first-order setting see *Metamathematics of Fuzzy Logic* [Háj98].

It is instructive to take a look at the differences between “mainstream” fuzzy logic as we use it here and probability theory [8]:

The statement: “This water is poisoned to a degree of 0.5” is substantially different from the statement: “This water is poisoned with a probability of 0.5”. In the former case a person drinking the water would get stomachache and would not feel to well for a few days, in the latter case the equivalent of a coin-toss would decide whether the person wakes up the next day[9].

One of the defining requirements for *mainstream* fuzzy logic is that the connectives behave truth-functionally. The compound statement: “Barbara is working in a very responsible position

[7] Compare this to ν^*_{CL} from Definition 2.1.4

[8] There has been quite some research on the field of probability and logic and their possible combinations – see [HGE13] for example, however this is beyond the scope of this thesis

[9] This is the distinction between a *degree of truth* in the fuzzy setting, and a *degree of belief* in the probability setting.

and Barbara earns a very good salary" should only depend on the degree of truth assigned to each conjunct.

This is another distinction from probability theory, where the probability of two events occurring simultaneously, i.e. their conjunction occurring, is, in general, not solely dependent on their individual probabilities, but also on the dependent probabilities of one event with respect to the other.

Using the real unit-interval also presents us with the natural linear and dense order of its members, making the truth degrees comparable.

Given that some statements are *crisp* and have an absolute truth value, and that mathematical fuzzy logic is an extension to classical logic, we want the semantics to behave like in classical logic on the extreme values 0 and 1.

The field of fuzzy logic can be roughly divided into two subfields. On the one hand there is "fuzzy logic in the *broad* sense" [Zad94, Háj11, CFN17, Háj98], which is also referred to as fuzzy set theory and fuzzy control. Originally introduced by Zadeh in 1965, addressing the problems arising when modeling "...complex input–output relations in an environment of imprecision and uncertainty." [Zad94], it has found a large range of applications from the control of washing machines to analysis of natural language and the field of soft computing.

Fuzzy logic in the broad sense is studied in the fields of engineering, applied computer engineering and applied sciences. A thorough introduction to the subject from this perspective can be found in the textbook by Nguyen and Walker [NW05].

Arguably the logic identified as *fuzzy logic* in literature focused on fuzzy set theory, e.g. in chapter four of [NW05], or in work predating [Háj98][10], can be identified as the logic containing only the weak conjunction and disjunction[11] as well as negation of Łukasiewicz logic.

Originally introduced by Zadeh in the seminal article on *Fuzzy Sets* in 1965 [Zad65] as fuzzy logic, the obtained system coincides with the equational logic of Kleene Algebras [AGM09].

For these two historic roots we call the logic synonymously KZ-logic (Kleene-Zadeh logic) or $Ł^w$ (for weak Łukasiewicz logic), following more recent work in the field, e.g. [FM15, FR12, Fer14].

On the other hand there is the field of mathematical fuzzy logic as laid out in Hájek's defining monograph *Metamathematics of Fuzzy Logic* [Háj98]. Nowadays it has become a well established discipline within formal logic. It is frequently identified with Zadeh's notion of "fuzzy logic in the *narrow* sense" [Háj98, CHN11, CFN17]. As such it addresses the questions asked in the field of formal logic: "...syntax, model theoretic semantics, proof systems, completeness, etc." [CFN17]. The term *t-norm based fuzzy logics* is used synonymously, since a triangular norm or t-norm serves as the truth function for strong conjunction.

2.2.1 KZ-Logic or $Ł^w$

One of the main results of this work provides a formal modeling of evaluation games for KZ-logic. We picked this particular logic from the field of fuzzy logics, because it has a few very pleasant properties when it comes to formalizing its semantics as a Hintikka style evaluation

[10] For example a paper by Novak [Nov87]

[11] also referred to as lattice connectives or min conjunction and max disjunction

game: its conjunction and disjunction being idempotent, its negation involutive, we need not consider keeping track of multisets of all occurring subformulas, as is needed for Giles' game for full Łukasiewicz logic [FM09, Gil77].

The game obtained needs only to consider the (sub)formula currently analyzed, the only change from the classical version, is that the outcome of the game has to be represented by a real payoff value, instead of a win or loss for the players.

Definition 2.2.1 (KZ syntax)**.** The syntax of KZ-logic is the syntax of classical propositional logic (Definition 2.1.1), restricted to the three connectives $\neg, \wedge, \vee$:

$$\varphi := \quad a \mid \neg\varphi \mid (\varphi \wedge \psi) \mid (\varphi \vee \psi)$$

with $a \in \mathcal{Atom}$.

The truth functions in KZ-logic for conjunctions, disjunction and negation as minimum, maximum and $1 - x$ respectively, are suitable candidates, when considering the constraints on connectives for fuzzy-logic (truth-functionality and classical behavior on absolute values).

As for classical propositional logic, we introduce *atomic* and *compound* valuations for KZ-logic.

Definition 2.2.2 (KZ valuation)**.** An atomic valuation for KZ-logic ν_{KZ} is a mapping from $\mathcal{Atom}$ to the real unit-interval $[0, 1]$: $\nu_{KZ} : \mathcal{Atom} \to [0, 1]$, with $\nu_{KZ}(\top) = 1$ and $\nu_{KZ}(\bot) = 0$. We say a evaluates to r, if $\nu_{KZ}(a) = r$. The valuation for compound formulas ν^*_{KZ} is defined inductively:

- $\nu^*_{KZ}(\varphi) = \nu_{KZ}(\varphi)$, iff $\varphi \in \mathcal{Atom}$
- $\nu^*_{KZ}(\varphi \wedge \psi) = \min(\nu^*_{KZ}(\varphi), \nu^*_{KZ}(\psi))$
- $\nu^*_{KZ}(\varphi \vee \psi) = \max(\nu^*_{KZ}(\varphi), \nu^*_{KZ}(\psi))$
- $\nu^*_{KZ}(\neg\varphi) = 1 - \nu^*_{KZ}(\varphi)$[12]

The simple structure of KZ-logic comes at the price of its restricted expressiveness: the lack of implication as the syntactic connective capturing semantic consequence is a downside to our intention of using logic for reasoning. The definition of implication via negation and disjunction ($\varphi \to \psi \Leftrightarrow \neg\phi \vee \psi$) does not translate well to KZ-logic, because $\varphi \to \varphi$ is not a tautology anymore.

The equivalence between the set of all valid classical formulas and those formulas, which have a *value* of 0.5 in all KZ-valuations was shown quite elegantly by Goldstern in the two papers [Gol97, Gol13]. This suggests that valid formulas in KZ need to make use of truth constants on a very essential level [Fer14].

[12]Note, that, if we restrict our attention only to the extreme values of the interval 0 and 1, this definition coincides with the one for classical valuations (Definition 2.1.4)

2.2.2 T-Norm Based Fuzzy Logics

A very fruitful approach, both for addressing with the shortcomings in KZ-logic, and unifying historical treatments of many-valued logics — most prominently Łukasiewicz's three valued logic introduced in [Łuk20], and the system developed by Gödel in [Göd32] for analyzing intuitionistic logic — results from the use of a triangular norm[13] as the truth function for strong conjunction.

The resulting system is called Basic Logic (BL) and by adding one single axiom to its set of axioms (Definition 2.2.6) systems for the prominent fuzzy logics are obtained. See Figure 2.1 for a visual description of the logics, w.r.t. to their expressive strength.

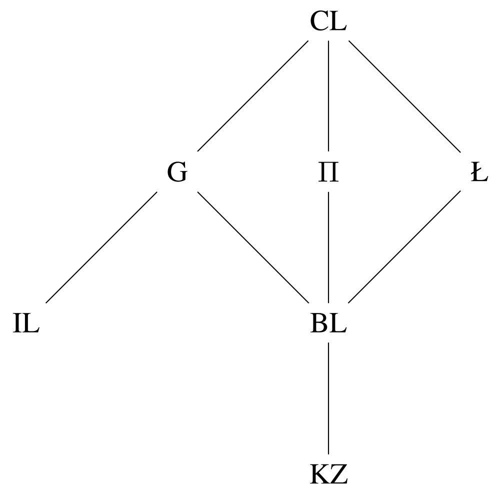


Figure 2.1: The mathematical fuzzy logics discussed in this thesis, in relation to intuitionistic logic (IL) and classical logic (CL), ordered from bottom to top according to their expressive strength. (inspired by [CHN11])

2.2.3 Basic Logic

We follow Hájek's monograph in initially presenting Basic Logic (BL) as the system resulting from taking a continuous t-norm as function for strong conjunction, followed by showing the three prominent logics (Łukasiewicz logic, Gödel logic and product logic), corresponding to the three primitive t-norms respectively.

Definition 2.2.3 (t-norm)**.** A binary function $* : [0,1]^2 \to [0,1]$ is called a *t-norm* if it fulfills the following conditions for $x, y, z \in [0,1]$:

1. $x * y = y * x$
2. $(x * y) * z = x * (y * z)$

[13] A thorough examination of t-norms can be found in [KMP00]

3. $y \leq z \Rightarrow (x * y) \leq (x * z)$

4. $x * 1 = x$

Thus $*$ is a commutative monoid with 1 as its unit element. Furthermore $*$ is monotone in both arguments.

This definition fulfills our requirements for the connectives — it is *truth-functional*, has the real unit-interval as its range and does behave like classical conjunction on 0 and 1.

Additionally Hajek and others [Háj98, CHN11] have named two further conditions or *design choices*, for truth function of conjunction: First a large truth degree of $\varphi \;\&\; \psi$ should imply a large truth degree of both conjuncts. Second the function should be continuous[14].

There exist uncountably many operations on $[0, 1]$, which fulfill the conditions of Definition 2.2.3 [KMP00]. Due to the Mostert-Shields theorem all t-norms can be represented as *ordinal sums* of three fundamental t-norms [CHN11]:

Definition 2.2.4 (Fundamental continuous t-norms)**.** The following are referred to as fundamental t-norms (see Figure 2.2 for a graphical presentation):

- The *Łukasiewicz t-norm*: $x *_{Ł} y = \max(x + y - 1, 0)$ (Figure 2.2a)
- The *minimal* or *Gödel t-norm*: $x *_G y = \min(x, y)$ (Figure 2.2b)
- The product on the reals: $x *_{\Pi} y = x \cdot y$ (Figure 2.2c)

Each continuous t-norm $*$ induces a unique binary operation $\Rightarrow_*$ on $[0, 1]$, s.t. for $x, y, z \in [0, 1]$ we have $z * x \leq y$ iff $z \leq x \Rightarrow_* y$, called its *residuum*, which is a suitable candidate for the truth function for implication.

Definition 2.2.5 (Residua of the fundamental t-norms)**.** The residua of the fundamental t-norms are (see Figure 2.3): for $x \leq y$, $x \Rightarrow_* y = 1$, else

- *Łukasiewicz residuum*: $x \Rightarrow_{*_Ł} y = 1 - x + y$ (Figure 2.3a)
- *Gödel residuum*: $x \Rightarrow_{*_G} y = y$ (Figure 2.3b)
- product residuum: $x \Rightarrow_{*_\Pi} y = x/y$ (Figure 2.3c)

The operations of minimum and maximum are definable by a t-norm and its residuum, which makes the structure: $\langle [0, 1], \wedge, \vee, \leq, 0, 1 \rangle$ a linear and commutative *lattice* [15].

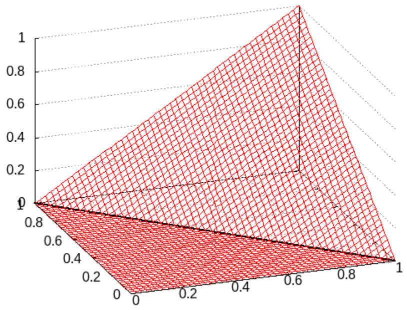
This fact is reflected in the possibility to define $\wedge, \vee, \neg$, from $\&, \rightarrow, \bot$[16]:

- $\varphi \wedge \psi \Leftrightarrow \varphi \;\&\; (\varphi \rightarrow \psi)$
- $\varphi \vee \psi \Leftrightarrow ((\varphi \rightarrow \psi) \rightarrow \psi) \wedge ((\psi \rightarrow \varphi) \rightarrow \varphi)$

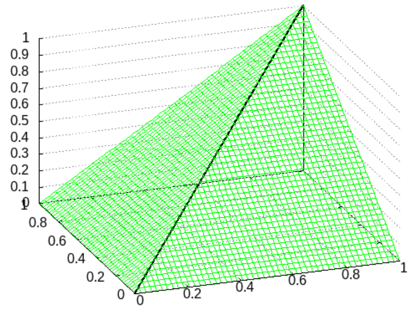
[14] As put in [CHN11]: "The condition of continuity formalizes the intuitive idea that an infinitesimal change of the truth value of a conjunct should not radically change the truth value of the conjunction."

[15] Which explains why the connectives of KZ-logic are called as lattice connectives.

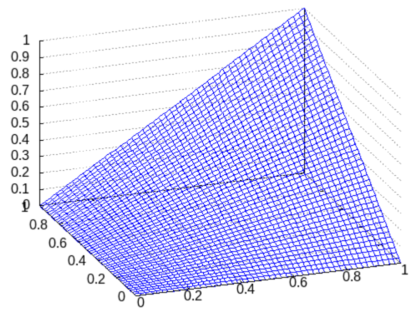
[16] This makes the set $\&, \rightarrow, \bot$ *functionally complete*



(a) The Łukasiewicz t-norm



(b) The Gödel t-norm



(c) The Product t-norm

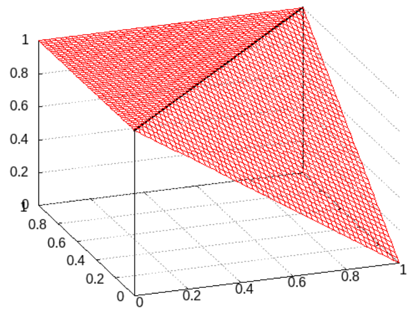
Figure 2.2: The three fundamental t-norms

- $\neg\varphi \Leftrightarrow \varphi \rightarrow \bot$
- $\top \Leftrightarrow \neg\bot$

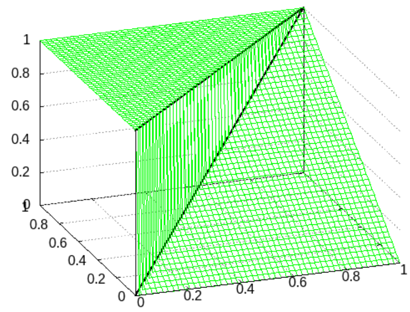
Instead of defining a logic either in terms of its standard semantics, as we did for classical propositional logic in Definition 2.1.4, or through its consequence relation, as was done for classical first-order logic in Definition 2.1.12 it is also possible to identify a logic with a syntactic proof-system, consisting of sets of *axioms* and *deduction rules*. Arguably one of the central questions for a logic is whether there is a correspondence, between syntactic *provability* and semantic *truth*[17].

We choose this syntactic approach for basic fuzzy logic, since it facilitates its comparison to the prominent fuzzy logics. They are obtained by adding one further axiom to the set for BL, that reflects their respective increased expressiveness compared to BL and their mutual difference in a visual way.

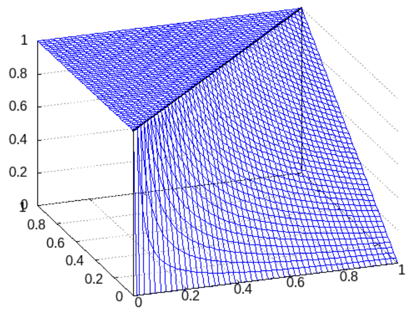
[17] Actually there are the two questions of *soundness* (provability implying truth) and *completeness* (truth implying provability).



(a) The residuum of the Łukasiewicz t-norm



(b) The residuum of the Gödel t-norm



(c) The residuum of the Product t-norm

Figure 2.3: Residua of the three fundamental t-norms

Definition 2.2.6 (Axioms of BL)**.** The following is a minimal set of axioms for Basic Logic

(BL-1) $(\varphi \to \psi) \to ((\psi \to \chi) \to (\varphi \to \chi))$

(BL-2) $\varphi \mathbin{\&} (\varphi \to \psi) \to \psi \mathbin{\&} (\psi \to \varphi)$

(BL-3) $(\varphi \mathbin{\&} \psi \to \chi) \to (\varphi \to (\psi \to \chi))$

(BL-4) $(\varphi \to (\psi \to \chi)) \to (\varphi \mathbin{\&} \psi \to \chi)$

(BL-5) $((\varphi \to \psi) \to \chi) \to (((\psi \to \varphi) \to \chi) \to \chi)$

(BL-6) $\bot \to \varphi$

Taking *modus ponens* — from φ and $\varphi \to \psi$ infer ψ — as the sole deduction rule yields a complete proof system for Basic Logic.

2.2.4 Łukasiewicz Logic

Originally invented by Łukasiewicz in 1920 as a three valued logic, for analyzing modal concepts — the third truth value represented something being possibly true — Łukasiewicz logic as we present it here is arguably the most prominent and best-studied mathematical fuzzy logic. In literature, e.g. an article by Novak [Nov87], it is often identified with *fuzzy logic* or *many valued logic*

Logically it offers a few nice properties, that are lacking in other fuzzy logics: negation defined through implication is involutive — thus the double negation law holds. Furthermore the residuum of the Łukasiewicz t-norm is continuous. Propositional Łukasiewicz logic is obtained from Basic Logic by the following axiom, reflecting the involutive negation:

(Ł) $\neg\neg\varphi \to \varphi$

2.2.5 Gödel Logic

Gödel or Gödel-Dummet logic historically stems from Gödel's examination of Intutionism in [Göd32]. Dummet proved the completeness for infinitely valued Gödel-Dummet logic of a propositional set of Axioms in his paper from 1959 [Dum59]. Using *minimum* as a t-norm makes the two conjunctions coincide. The axiom needed for Gödel logic reflects that strong conjunction is idempotent:

(G) $\varphi \to (\varphi \mathbin{\&} \varphi)$

2.2.6 Product Logic

Product logic has only been the focus of scientific investigations comparatively recently. Its first axiomatization was obtained in 1996 by Hájek, Godo and Esteva in [HGE96]. Their starting point was using the product, being the third fundamental t-norm, as a candidate for conjunction. Adding the following axiom to Basic Logic yields a complete axiomatization for propositional product logic:

(Π) $\neg\varphi \vee ((\varphi \to \varphi \mathbin{\&} \psi) \to \psi)$

2.3 Modal Logics

Having its roots in philosophy, where it still is an active topic of research, modal logic has found rich interaction with the fields of mathematics, computer science, game theory and linguistics [BvBW07]. A modal qualifies the truth of a logical statement. Statements can be "necessarily true" and "possibly true" in addition to just being true. Historically the notion of modality entered philosophical discussion for addressing problems with material implication[18] by Lewis in the end of the nineteenth century. On the symbolic level modalities appear as non-truth-functional operators: for example we write $\Box\varphi$ for "φ is necessary". Possibility is dual to

[18] The equivalence of $\varphi \to \psi \Leftrightarrow \neg\varphi \vee \psi$

necessity — saying that something is not necessarily not true, amounts to it being possibly true[19] — it is denoted as $\Diamond\varphi$.

Apart from necessity, that is also referred to as *deontic* modality, modal logic is a suited tool for many other areas as well: time, knowledge, computation, obligation and action, to name a few.

From a computer-science perspective the area of *computer aided verification* has found a rich and expressive formalism in computation tree logic (*CTL*), and linear temporal logic (*LTL*), used to reason about properties, for example the absence of dead-locks,of programs.

A concept, tightly linked to modal logic, are *possible world semantics*, also known as *Kripke semantics* , in reference to Kripke, who presented them in 1963 in his paper [Kri63]. The central idea is that the meaning of φ being necessarily true, is translated to φ being true in *all possible worlds*, and φ being possibly true, meaning it being true in *at least one possible world*. Each possible worlds has a valuation (see Definition 2.1.4), that may be different in different worlds. The most natural way of representing Kripke structures is a directed graph, with the nodes representing the possible worlds, and the edges describing the possible transitions between those worlds. A typical visual representation of a Kripke model with those propositional variables, that are true written below the worlds can be seen in Figure 2.4.

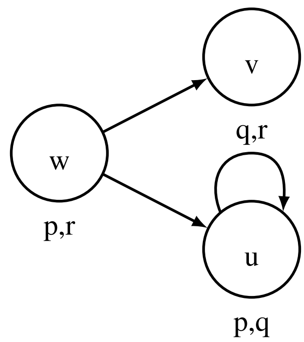


Figure 2.4: An example Kripke structure

In our work we use a multi-modal logic[20] to formally describe the game-trees of logical evaluation games. The worlds correspond to the nodes of the game tree, the accessibility relations represent the moves at each history and the *signature* captures the labels of the tree.

2.3.1 Propositional Modal Logics

Like for classical logic we introduce the formal definitions for modal logic, by starting with a propositional version, that is expanded to a first-order logic.

Definition 2.3.1 (signature of propositional modal logic)**.** In addition to propositional variables (see Definition 2.1.3) a *modal signature* contains a set of modalities: $\mathcal{Mod}$ as well.

[19] A similar duality as with the first-order quantifiers: $\exists x P(x) \Leftrightarrow \neg\forall x \neg P(x)$

[20] A logic with more than one modality

Definition 2.3.2 (syntax of propositional modal logic)**.**

$$\varphi := \quad a \mid \neg\varphi \mid (\varphi \wedge \psi) \mid (\varphi \vee \psi) \mid (\varphi \rightarrow \psi) \mid (\varphi \leftrightarrow \psi) \mid \langle m \rangle \varphi \mid [m]\varphi$$

with $a \in \mathcal{Atom}$ and $m \in \mathcal{Mod}$. If we talk about a logic with only one modality we sometimes denote the modalities as $\Box$ and $\Diamond$ instead of $[m], \langle m \rangle$.

Definition 2.3.3 (Kripke model)**.** A *Kripke model* is a triple $K = \langle W, \{R_m\}_{m \in \mathcal{Mod}}, V \rangle$, where

- W is a non-empty set of *worlds*.
- Each R_m is a binary relation on W: the *accessibility relation* for m.
- V is a mapping for each propositional variable p to a subset of worlds, where p is true, viz. $V(p) = X : X \subseteq W$.

The pair $\langle W, \{R_m\}_{m \in \mathcal{Mod}} \rangle$ of a *Kripke model*, i.e. the Kripke model without the valuation, is called a *Kripke frame* or just *frame*.

We refer to the *worlds* synonymously as *states* or *points*.

Truth in modal logics refers is evaluated with respect to a world, thus the *satisfiability relation* (see Definition 2.1.12) here refers to a evaluation in a particular world:

Definition 2.3.4 (modal satisfiability and validity)**.** Let $K = \langle W, \{R_m\}_{m \in \mathcal{Mod}}, V \rangle$ be a *Kripke model*, $w \in W$, $m \in \mathcal{Mod}$ and φ, ψ formulas. We define *modal satisfiability* as a relation from K and a world w to a formula:

- $K, w \models p$, iff $w \in V(p)$
- $K, w \models \langle m \rangle \varphi$, iff $\exists v \in W$ s.t. $(w, v) \in R_m$ and $K, v \models \varphi$
- $K, w \models [m]\varphi$, iff $\forall v \in W$ s.t. $(w, v) \in R_m$, we have $K, v \models \varphi$
- The clauses for the connectives remain, with the addition of w, as in Definition 2.1.12 items 5–11.

A formula φ is said to be *globally satisfied* in K, iff it is true for all $w \in W$ and *valid* if it is globally satisfied in all Kripke models.

Take the Kripke model from Figure 2.4 as an example. There the following satisfiabilities hold:

- $K, w \models p$
- $K, w \models \Diamond p$
- $K, w \not\models \Box p$
- $K, u \models \Box(q \wedge r)$
- $K, v \models \Box\bot$
- $K, w \models \neg q \wedge \Box q$

Propositional modal logic has an interesting connection to classical logic. While it can be seen as propositional classic logic with added modalities, another, quite fruitful, view is seeing it as a fragment of first-order logic, with the domain of quantification being the worlds. Each accessibility relation is translated to a binary predicate symbol, and each propositional variable to a unary predicate symbol. This is called the *standard translation* of modal logic. The modal formula $\Box p \to q$ then translates to $\forall y(R(x,y) \to P(x)) \to Q(x)$, with R representing the accessibility relation of the single modality and P and Q being the predicate symbols for p, q respectively. The formula has one free variable, and evaluates to true, with a world w assigned to x, iff the original modal formula is true in w. The standard translation provides us with properties like *Compactness* or the *Löwenheim-Skolem property* for propositional modal logic. Being a strict fragment of first-order logic is reflected in the facts that most propositional modal logics are decidable, and that they exhibit the *finite model property* — there is no formula having only infinite models. In classical first-order logic there are such formulas, like $\forall x(\neg R(x,x)) \wedge \forall x \forall y \forall z(R(x,y) \wedge R(y,z) \to R(x,z)) \wedge \forall x \exists y(Rx,y)$. Proofs of the above results and further details of the interaction of both logics can be found in the book *Modal Logic and Classical Logic* by van Benthem [vB83] and in the first chapter of the *Handbook of Modal logic* [BvBW07].

2.3.2 First-Order Modal Logics

While it might look like First-order modal logic is a direct combination of classical first-order logic and propositional modal logic, the topic raises many questions with respect to the interactions between quantifiers and modalities. Although their fine points go beyond the scope of this work, we have to take them into consideration and justify our particular choices regarding their resolution. The interested reader is referred to the very accessible treatment of the subject in *First-Order Modal Logic* by Fitting and Mendelsohn [FM98], which serves as a basis for our treatment as well. As phrased in the foreword to the book:

> Classically, first-order issues like constant and function symbols, equality, quantification, and definite descriptions, have straightforward formal treatments that have been standard items for long time. Modally, each of these items needs rethinking. First-order modal logic, most decidedly, is not just propositional modal logic plus classical quantifier machinery.

Take the statement "The number of planets is necessarily even" as a motivating example: we could translate it either into $\exists x \Box(P(x) \wedge Q(x))$ or into $\Box \exists x(P(x) \wedge Q(x))$, and evaluate it in a model where the *interpretation* of P is "is even" and the one for Q is "is the number of planets". The former is true, since the number of planets is 8, which is an even number. The latter on the other hand is arguably wrong, as people remembering the definition of planets in our solar system until 2006, back then including Pluto, do know a world where the number of planets is odd[21]. The two formalization of the statements correspond to what is known as the *de re* and the *de dicto* reading of the statements respectively.

Syntactically these problems are of no concern: first-order modal logic uses the same basic objects as first-order logic (predicate and function symbols and variables) in addition to modal operators.

[21] The same example was used in [FM98] with "the number of planets is necessarily odd".

As in the propositional case the notion of signature from first-order logic is augmented by the used modalities.

Definition 2.3.5 (signature of first-order modal logic)**.** The *signature* of a first-order modal logic consists of:

- a set of predicate symbols of a given arity
- a set of function symbols of a given arity
- a set of modalities

The concept of a *term* (see Definition 2.1.5) remains unchanged. Summing up we define the complete syntax as a combination of first-order logic and propositional modal logic.

Definition 2.3.6 (syntax of first-order modal logic)**.**

$$\varphi := \quad P(t_1, \ldots, t_n) \mid t = s \mid \neg\varphi \mid (\varphi \circ \psi) \mid \langle m \rangle\varphi \mid [m]\varphi \mid \forall x\varphi \mid \exists x\varphi$$

with $\circ \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$, $P \in PS_n$, $t, s, t_1, \ldots, t_n \in T$, $m \in \mathcal{Mod}$, and $x \in VS$.

Definition 2.3.7 (free and bound occurrences of variables)**.** Starting from Definition 2.1.7, we add the following clause stating that modalities do not change the binding of variables:

- $\circ\varphi$, for $\circ \in \{\langle m \rangle, [m]\}$: $FV(\circ\varphi)$= $FV(\varphi)$.

Semantically the combination of quantifiers and modalities has a few peculiarities. On the propositional level every world has its own valuation for propositional variables, which translates in first-order to every predicate symbol and function symbol being possibly interpreted differently in each world. The question remains, of whether the domain should be interpreted locally in each world, or whether there should be a global one for the whole model. Arguably the first version seems more natural, for instance when considering temporal modalities — neither does everything that will exist at some point in the future already exist, nor will everything, that exists now, exist in the future. This leads to situations where a bound variable occurring in scope of a modality quantifies over objects which do not exist when the formula is evaluated[22]. Thus neither $\Box(P(x) \vee \neg P(x))$ nor $\forall x P(x) \rightarrow P(x)$ are globally satisfiable in this case.

Formally the situation is addressed, either by having a single domain, which is used for quantification in all worlds, or additionally having a function, which assigns subsets of the domain[23], to each world. The approaches are known as *constant domain semantics* and *varying domain semantics* respectively. Both semantics can be translated into each other and have the same expressive power. Constant domain semantics are the special case of varying domain semantics,

[22] One approach for resolving this, is adding a third truth value and thus combining modal logics with a many-valued logic, like Łukasiewicz' original system; see page 231 of [FM98], another one is to make the quantified formula false in these cases.

[23] Alternatively this can be read as each world having its own domain, and referring to the domain of the model as the union of the world domains.

where the domain-function assigns the whole domain to every world. The other direction needs equality and adds predicates, which reflect the existence of objects in each world.

In constant domain semantics the two schemas $\forall x\Box\varphi \rightarrow \Box\forall x\varphi$ and $\Box\forall x\varphi \rightarrow \forall x\Box\varphi$ hold. They are known as *Barcan formula* and *Converse Barcan formula* respectively. The Barcan formula holds in decreasing frames, where the domain of a world is a superset of the domains for all reachable worlds. The Converse Barcan formula similarly holds in increasing frames. Before the introduction of Kripke semantics modal logics were mostly analyzed by axiom schemas, and many debated axiom schemas syntactically describe intuitive conditions on the class of Kripke models they are valid in. The Barcan formulas are an example for this and reflect that quantifiers and modalities are interchangeable.

For our purposes constant domain semantics are sufficient and provide a better behaving system. We quantify over propositional formulas and reals, and can safely assume that they do not cease to exist after game moves are performed.

Definition 2.3.8 (first-order constant domain Kripke model)**.** A *first-order constant domain Kripke model* is a tuple $K = \langle W, \{R_m\}_{m\in\mathcal{Mod}}, D, V\rangle$, where:

1. $\langle W, \{R_m\}_{m\in\mathcal{Mod}}\rangle$ is a Kripke frame
2. D is the *domain*.
3. V, the *interpretation* as in Definition 2.1.8, takes a world as an additional argument. It is thus a mapping defined on $W \times (PS \cup FS)$ s.t.:
   - $V(w, f) : D^k \rightarrow D$, for $f \in FS_k$.
   - $V(w, P) : D^k \rightarrow \{0, 1\}$, for $P \in PS_k$.

Where convenient we identify the *interpretation* of a k-place predicate symbol, at a world w, $V(w, P)$, with the characteristic function of a set of $k+1$-tuples, $V(P)$, where the first element is w and the remaining k are the domain-elements denoted by the given arguments — for $w \in W, d_i \in D$ we have:

$$(w, d_1, \ldots, d_k) \in V(P) \text{ , iff } V(w, P)(d_1, \ldots, d_k) = 1.$$

We sometimes do not mention the constant domain explicitly and refer to the model simply as *first-order Kripke model* or just *Kripke model*.

The definition of *variable valuations* (Definition 2.1.9) and *variable variants* (Definition 2.1.10) remain unchanged, since variable valuations are stable across worlds and are always well defined in constant domain semantics. The *term valuation* (Definition 2.1.11) is lifted analogously to the *interpretation* to include a particular world.

The interpretation of function and constant symbols being local to a world seems like a natural extension of the propositional case. A closer look is advisable to understand the interesting properties of first-order modal logic. The sentence: "Soon there will be computers, that are ten times faster than the fastest computer." serves as an example — If we formalize this sentence in a temporal modal logic, we would designate "the fastest computer" by a constant symbol c, and the phrase "Soon there will be" by $\Box$, we would write it as: $P(c) \to \Box\exists x(Q(c, x))$, with P representing the predicate "is the fastest computer", and Q representing "is ten times faster than". Suppose we evaluate the formula at a world w representing the current time. Then c is the currently fastest computer, but if interpreted in a world v, with $(w, v) \in R_\Box$, c would designate a different computer, the fastest computer in v, which would make $\exists x(Q(c, x))$ false. Yet the sentence we stated makes sense to us in a temporal setting. It is true, if we let c be read as "the fastest computer in w". The latter reading of c is referred to as a *rigid designator* in [FM98]. The problem in this example is due to a syntactic ambiguity of the modal language and is inherently linked to the non-truth-functionality of modalities. In literature the problem is addressed by introducing *predicate abstractions*, with a syntax borrowing from the λ-calculus, making it possible to distinguish the two formulas $\langle \lambda x.\Box P(x)\rangle$ and $\Box\langle \lambda x.P(x)\rangle$ [FM98, BvBW07]. They both coincide in our language as $\Box P(x)$. The formalism removes the syntactic ambiguity, in these cases.

The term *rigid designator* goes back to a series of lectures by Kripke, published as *Naming and Necessity* [Kri80], and describes *terms*, that designate the same objects in every world of a model. It also introduces the additional term *strong designator*, for a rigid designator, that refers to a existing object. An interpretation, assigning the same objects to terms in all worlds is called a *rigid interpretation*.

The different treatment of terms and variables lies in the difference of how they get interpreted — functions and constants get their interpretation in the context of a particular world, whereas variables get their value through the variable valuation, which is global for a model.

For our purposes all constants and terms are rigid. Having decided on constant domain semantics and rigid interpretations we define the modal first-order satisfiability relation.

Definition 2.3.9 (modal first-order satisfiability)**.** The *satisfiability relation* for first-order modal logic is the combination of satisfiability in propositional modal logic (Definition 2.3.4), and the satisfiability in first-order logic (Definition 2.1.12), when taking into account that predicates and terms are evaluated, with reference to a given world.

2.3.3 Correspondence Theory

As indicated above with the Barcan and Converse Barcan formulas, it is possible to express qualities of models, within the modal language.

Apart from being able to force the domain of a model to be the same for every world, many other conditions of the frame can be expressed even in propositional modal logic. The topic of this interaction is called (modal) correspondence theory and has been discussed in an identically named article by van Benthem [vB01].

Historically the concern was the expressibility of certain axioms, and whether their semantics are applicable in a certain modal context. The question, whether something that is necessary is necessarily necessary, for example, can be expressed as $\Box\varphi \to \Box\Box\varphi$, and is valid in all frames, where the accessibility relation is transitive. The axioms were discussed as additions to an early modal system introduced by Lewis, called S1 and were named S2, S3, S4 or S5 for example. It still is customary to name certain axioms according to the system, to which they correspond. $\Box\varphi \to \Box\Box\varphi$ is referred to as 4, since it defines increase in expressive strength of

S4 in contrast to the basic modal system K. A few well-known axioms along with their names, and the conditions they impose on frames, can be seen in Table 2.2 and in most introductions to modal logics [FM98, BvBW07, Bal17, Gar16].

The basic modal system K extends classical propositional logic with $\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$ as additional axiom, and modal necessitation as additional rule. Modal logics which extend it are called *normal modal logics*.

Name	Axiom	Frame condition ($u, v, w \in W$)	Condition name
T	$\Box\varphi \to \varphi$	$\forall u(R(u,u)$	reflexive
4	$\Box\varphi \to \Box\Box\varphi$	$\forall u\forall v\forall w(R(u,v) \wedge R(v,w) \to R(u,w))$	transitive
5	$\Diamond\varphi \to \Box\Diamond\varphi$	$\forall u\forall v\forall w(R(u,v) \wedge R(u,w) \to R(v,w))$	euclidean
B	$\varphi \to \Box\Diamond\varphi$	$\forall u\forall v(R(u,v) \to R(v,u))$	symmetric
	$\Box\bot$	$\forall u(\neg R(u,u))$	irreflexive points
D	$\Box\varphi \to \Diamond\varphi$	$\forall u\exists v(R(u,v))$	serial
	$\Diamond\varphi \to \Box\varphi$	$\forall u\forall v\forall w(R(u,v) \wedge R(u,w) \to v = w$	partial functionality
	$\Diamond\varphi \leftrightarrow \Box\varphi$	$\forall u(\exists v(R(u,v) \wedge \forall w(R(u,w) \to v = w)$	functionality

Table 2.2: Modal axioms with their historic names and the conditions they impose on frames

Not every condition imposed on a frame can be expressed by adding axioms to the basic system K, and there is no concise characterization of those conditions that can be defined through axioms. One way of obtaining results about the non-undefinability of a certain condition is observing that the definable frames need to be robust against the formation of the disjoint union of models within that class, or the generation of subframes. Thus for example it is not possible to qualify frames, which contain some irreflexive points by an axiom. Any subframe, without irreflexive point could not become a model for the formula. A more detailed treatment of modal correspondence theory is presented in more detail in [vB01].

From the perspective of the evaluation games, that we want to model, some of the desired properties can be expressed syntactically, while we need to state the other restrictions on a semantic level. The game-trees are irreflexive, however they are connected and thus the axiom in Table 2.2 cannot be used to express the desired property. A system used to reason about provability, called GL is classified by the axiom $\Box(\Box\varphi \to \varphi) \to \Box\varphi$. Its models are based on frames, which are transitive, finite and irreflexive [BvBW07, Gar16]. However our game trees are not transitive, therefore we cannot use instances of this axiom for our modalities to characterize the game trees.

One condition we need for our game trees is partial functionality, as expressed in Table 2.2, at every history, if a move is possible, we want it to lead to exactly one next state.

It is instructive to take a closer look at the difference between the axioms for *partial functionality* and *functionality* — for total functionality the direction $\Box\varphi \to \Diamond\varphi$ holds in addition to the axiom for partial functionality. In general this direction holds, after all, if a formula is true in all reachable worlds, it should be true in a particular one. However $\Box\bot \to \Diamond\bot$ is not true in the case, where there is no reachable world — In that case $\Box\bot$ is trivially true by Definition 2.3.4, whereas $\Diamond\bot$ is not. Therefore a formula of the form $\Box\bot$ for any particular modality holds exactly at those worlds where no world is accessible. Axiom D is the converse implication and it does indeed express that every world has a successor.

2.4 Game-Theoretic Preliminaries

Games, whether the ones we play for pleasure, like Monopoly, Tic-Tac-Toe, Backgammon, Poker or Chess, or their formal counterpart generally have players take turns making decisions. When we want to analyze the decisions made at every given turn a *decision tree* provides us with a quite natural representation: every turn is a node and the possible actions are edges connecting the turns. This model is known in game theory as the *extensive form* of a *game*. Take Backgammon as a first motivating example: two players throw two dice and move their pieces around afterwards. Given a placement of the pieces on the board, assume a player, has n possible ways to move their pieces around. The node in the decision tree then represents the current board layout and it has n outgoing edges with the n possible resulting game boards after making the move. At first sight it seems natural to consider the two players as the only ones making decisions. However the dice are missing in this approach. We have a few ways to model the dice: One would be to incorporate the outcome of the next throw into the resulting game board after a move. Then, after a given throw with n different possible moves, we would have $36 * n$ successor states[24]: one for each possible move and outcome of the throw. This modeling suggests that a player has an influence on the result of a dice-throw. Another approach in modeling Backgammon would be to think of the dice as a third player who decides, which outcome a dice-throw yields, after each move. This is referred to as a *game with chance moves*.

Poker on the other hand has a different element, that makes it a game worth playing and worth analyzing from a game-theoretic point of view. Chance plays a role only initially when the cards a dealt, afterwards the outcome of the game is only dependent on the strategies the players follow. The element which is probably the most interesting when playing poker, is the lack of information each player has. Would all cards lie open on the table, the winner would be certain from the very beginning, and bluffing would be impossible, or at least not a very wise choice. It is the expectation a player has about the hand of her opponents, which makes poker interesting. In game theory Poker is called a *game of imperfect information*.

For the games we occupy ourselves with in this work, those considerations need not be made: chance plays no role in evaluating the truth value of a formula under a given truth valuation. Furthermore the two players, know everything by knowing the formula and the given valuation. The games belong to the class of *extensive games of perfect information*.

2.4.1 Extensive Games of Perfect Information

We took two standard text-books as a starting point for our treatment of game theory: *A course in game theory* by Osborne and Rubinstein [OR94] — especially chapter 6 on extensive games with perfect information — and *Fun and games* by Binmore [Bin92], where many ideas regarding zerosum games, and Zermelo's algorithm were explained in a very comprehensible way. The following definitions are based mainly on those two, with some additions taken from the chapter

[24] symmetric throws aside

on *Modal logic for Games and Information* in the *Handbook of Modal Logic* [BvBW07]. The explicit tree representation was inspired by two papers on game logic by van Benthem [vB03, vB02].

Definition 2.4.1 (extensive game form of perfect information)**.** An *extensive game form of perfect information* is a triple $\langle Pl, H, Tu \rangle$, where:

- Pl is a set of players. We use variables i, j to denote players. By $-i$ we mean the set of all players except i: $-i = \{j | j \in Pl \ , \ j \neq i\}$.

- H is a set of sequences (finite or infinite), over a set A of *moves* or *actions*. $h \in H, a_i \in A : h = (a_1, \ldots)$. We write $h|k$ to denote the initial subsequence of length k ($h|k = (a_1, \ldots, a_k)$), and (h, a) to denote the sequence of length $k+1$, where $h \in H$ is of length k and $a \in A$. Likewise we denote the concatenation of two sequences h_1 and h_2 by (h_1, h_2).

  H satisfies the following conditions:

  1. the empty sequence $() \in H$
  2. if $(a_k)_{k=1,\ldots,K} \in H$ (where K may be infinite), and $L < K$ then $(a_k)_{k=1,\ldots,L} \in H$
  3. if $(a_k)_{k=1}^{\infty}$ satisfies that $(a_k)_{k=1,\ldots,L} \in H$ for every $L > 0$, then $(a_k)_{k=1}^{\infty} \in H$

  We call $h \in H$ a *history*, and its components $a \in A$ *actions*. A history h is *terminal*, if it is infinite, or if there is a k s.t. $h = h|k$. We denote the set of terminal histories with $Z \subseteq H$.

- Tu is a function $Tu : H \backslash Z \rightarrow Pl$, that assigns to each non-terminal history the player whose turn it is — the *turn function*. By *active player* at a given history h we mean the player whose turn it is: $Tu(h)$.

We say that a game form is *finite* if H is finite, we say it has *finite horizon* if the longest $h \in H$ is finite.

Note that having a *finite horizon* is a weaker concept than being *finite*. If a game has an infinite history h, it would contain the infinitely many initial subsequences of h. A game having infinitely many actions available at a finite history is has finite horizon, but is not finite. The game displayed in Figure 2.5 is an example. It has only one intermediate node, but infinitely many actions — one for each $d \in [0, 200]$.

The one thing lacking to obtain a extensive game from a game form is something modeling players' wishes and preferences:

Definition 2.4.2 (extensive game of perfect information)**.** An *extensive game of perfect information G* is a tuple $G = \langle Pl, H, Tu, \{\succsim_i\}_{i \in Pl} \rangle$, where:

- $\langle Pl, H, Tu \rangle$ is an *extensive game form of perfect information*

- each $\succsim_i$ is a *preference relation* on the *terminal histories* for player i: $\succsim_i \subseteq Z \times Z$. $h_1 \succsim_i h_2$ means that player i prefers terminal history h_1 over h_2.

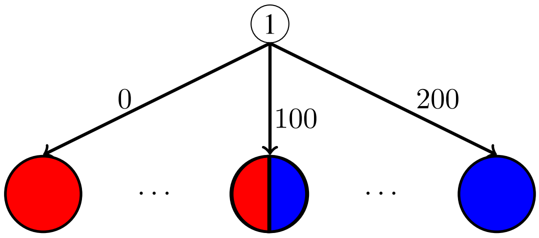


Figure 2.5: Cake Splitting game — infinite constant-sum game with finite horizon

We write $h_1 \sim_i h_2$, iff $h_1 \succsim_i h_2$ and $h_2 \succsim_i h_1$. If $h_1 \succsim_i h_2$ and not $h_1 \sim_i h_2$ we write $h_1 \succ_i h_2$. We may use *payoff functions* $u_i : Z \to \mathbb{R}$ assigning payoffs for players i to a given terminal history h. The payoff function defines the preference relation: $h_1 \succsim_i h_2 \Leftrightarrow u_i(h_1) \geq u_i(h_2)$, making $\succsim_i$ linear. When talking about the payoffs of all players we collect them into a *payoff vector* $(u_i(h))_{i \in Pl}$.

Introducing extensive game forms separately from extensive games, as usually done, emphasizes that the utilities enter the game only at the end of each run. For our logical evaluation games this provides the link with the given atomic valuation: for a given formula the set of histories is always the same, no matter which truth value an atom has. The valuation only plays a role at the terminal histories, saying if, or how much, a player has won. Each formula has only one game form, but, possibly infinitely, many games — one for each valuation.

Properties of the preference relations can be used to classify games. A *generic game* has a *strict* preference relation — no two histories are equally liked by any player. A typical example would be a coin toss — the game has two terminal histories, heads and tails, the player betting on heads prefers heads to tails, the other player prefers tails, and these preferences are strict.

Definition 2.4.3 (generic game)**.** We call a extensive game of perfect information (Definition 2.4.2) *generic*, iff no player is indifferent between any two outcomes of the game:

$$\forall i \in Pl,\ \forall h, h' \in Z\ :\ u_i(h) = u_i(h') \Rightarrow h = h'$$

Tic-Tac-Toe is an example of a non-generic game: a player that wins is completely indifferent, whether the three crosses are in a row, a column or a diagonal. Logical evaluation games are non-generic as well — a player defending a formula only cares about ending up in a leaf where the atomic formula evaluates to true, but does not care which leaf.

2.4.2 Game Trees

Extensive games are usually depicted through a rooted labeled tree, where each history corresponds to a node, or rather the branch of the tree from the root to this node. Note that there is a one-to-one correspondence between nodes and the path from the root to them in a tree, assuming the *shadow*[25] of the tree is acyclic. Each action is represented by an edge in the tree — just like every action transforms one history into another. Our work extensively uses these *game trees* for reasoning, hence we define them formally.

[25]The undirected graph resulting from a graph, where each edge goes in both directions

Definition 2.4.4 (graph)**.** A *graph* is a pair $\langle N, E \rangle$, of nodes N and edges E, were $E \subseteq N^2$.

We call $\langle N', E' \rangle$, with $N' \subseteq N$ and $E' \subseteq E$ a *subgraph* of $\langle N, E \rangle$.

Definition 2.4.5 (node-degree)**.** Given a *node* v in a graph $\langle N, E \rangle$, we define its *in-degree* $d^+(v)$ as the number of edges ending in v, and its *out-degree* $d^-(v)$ as the number of edges originating in it:

$$d^+(v) = |\{(u, v)|u \in N, (u, v) \in E\}|$$

$$d^-(v) = |\{(v, u)|u \in N, (u, v) \in E\}|$$

The sum of in-degree and out-degree is referred to as the *degree* $d(v)$ of a node v.

The *width* of a graph is defined as the maximum out-degree of all of its nodes:

$$\max_{v \in N}(d^+(v))$$

Definition 2.4.6 (labeled graph)**.** A *labeled graph* is a *graph* with, possibly multiple *labels* or *annotations* in its nodes, edges or both.

A *labeling function* l for a label t associates nodes or edges with labels from a set L: $l_t : N \to L$ or $l_t : E \to L$.

Definition 2.4.7 (path)**.** A *path* in a graph is a connection between two nodes via one or more edges. For $u, v \in N$ we write $u \rightsquigarrow v$, iff $(u, v) \in E^*$, where E^* denotes the transitive closure of E. If $u \rightsquigarrow v$ we say that u and v are *connected* or that there is a *path connecting* u and v.

Unless stating otherwise we exclude paths of length zero from u to itself.

By a path in a subgraph $\langle N', E' \rangle$, $u \rightsquigarrow_{E'} v$, we mean a path, where $\{u, v\} \in N'$ and all edges along the path are in E': $(u, v) \in E'^*$.

Definition 2.4.8 (tree)**.** A tree is an acyclic graph, i.e. one where the transitive closure of the relation E is acyclic, viz. $\nexists n \in N : n \rightsquigarrow n$.

It is called *rooted*, iff exactly one node has r has in-degree 0 (i.e. $\nexists n \in N : (n, r) \in E$). r is referred to as its *root*.

A rooted tree is called *connected*, iff there is a path, from the root to all nodes: $\forall n \in N : r \rightsquigarrow n$. Here we include the path with length 0 from r to itself.

Definition 2.4.9 (tree-node names)**.** Nodes in a tree are classified, according to their position in the tree: We call a node:

- A *leaf* or *terminal node* l, if it has out-degree 0 (i.e. $\nexists n \in N : (l, n) \in E$). The set of leafs for a graph $\langle N, E \rangle$ is denoted by N_0.
- A *non-terminal* or *intermediate node* n if it is not a leaf. $n \in N \backslash N_0$
- If $(u, v) \in E$, u is called the *parent-node* or simply *parent* of v, and v is a *child-node* or *child* of u.

Definition 2.4.10 (subtree)**.** A *subtree* is a *subgraph* of a tree. Given a rooted connected tree $\langle N, E \rangle$, we mean by the subtree of $\langle N, E \rangle$ *rooted at u*, the *rooted connected subtree* where u is the new root.

With the presented notation and concepts about graphs and trees we turn back to game theory.

Definition 2.4.11 (game tree)**.** A *game tree* is a *labeled rooted tree* $\langle N, E \rangle$ having at least the following *labels*:

- $l_{turn} : N \backslash N_0 \to Pl$ — the turn-labeling function, which indicates for each *intermediate node*, which players' turn it is.
- $l_{move} : E \to A$ — A label associating with each edge, the action it represents.
- $l_{payoff} : N_0 \to \mathbb{R}^{|Pl|}$ — A label associating with each terminal history the *payoff vector*: $l_{payoff}(h) = (u_i(h))_{i \in Pl}$

We do not restrict the labels to those three, because we may add other, specific, labels for each game we consider. For example the formula currently played.

2.4.3 Strategies and Equilibria

Definition 2.4.12 (strategy)**.** Let $A(h)$ denote the set of *actions* possible after an initial history h, $A(h) = \{a \mid a \in A, (h, a) \in H\}$, and let Tu^{-1} denote the pre-image of the turn function, assigning to each player i the set of histories h where $Tu(h) = i$.

We define a (pure) *strategy* for a player i as a function $\sigma_i : Tu^{-1}(i) \to A$, such that $\sigma_i(h) \in A(h)$. We use lower case Greek letters σ_i, τ_i to denote strategies. In the tree representation we identify with a strategy for player i in the game tree $\langle N, E \rangle$ a subtree $\langle N, E_{\sigma_i} \rangle$, with

$$(u, v) \in E_{\sigma_i} \Leftrightarrow \begin{cases} l_{turn}(u) \neq i \\ l_{turn}(u) = i, \quad \text{and } \sigma_i(u) = v \end{cases}$$

For a given strategy tree $\langle N, E_{\sigma_i} \rangle$ let N_{0,σ_i} denote the set of terminal nodes in $\langle N, E_{\sigma_i} \rangle$, which are still connected to the root i.e. $N_{0,\sigma_i} = \{v | v \in N_0, r \rightsquigarrow_{E\sigma_i} v\}$.

We denote the set of all strategies for player i in a game by Σ_i.

We say that player i *plays* action a at a given history h, if she is active at h and $\sigma_i(h) = a$.

The term *pure* strategy is used to distinguish strategies as defined here from *mixed* strategies. A pure strategy is a concrete plan for one run of the game. Mixed strategy, being a probability distribution over the set of pure strategies, express that a player i chooses to play a strategy τ_i with the associated probability. They are needed to capture optimal plans for players in games of imperfect information, and for games in strategic from. Games of perfect information, have the property of having an equilibrium over pure strategies.

It is instructive to contrast our definition for a strategy of player i in the tree representation, to the one only taking the edges from nodes n with $Tu(n) = i$, and excluding the moves of other players[26]. In our definition, all reachable terminal histories, if i plays σ_i, are still connected to the root. For an example see Figure 2.7. There is a bijection between the two presented choices for E_{σ_i}.

[26]This would be a more literal translation of the definition

Definition 2.4.13 (strategy profile)**.** A *strategy profile* is a tuple $\sigma = \langle \sigma_i \rangle_{i \in Pl}$ of strategies for each player. By $o(\sigma) \in Z$ we denote the *history*, that results if each $i \in Pl$ plays their strategy in σ. A *strategy profile* is a branch from the root to a leaf in the tree representation, which is $o(\sigma)$.

When we want to talk about the strategy of one specific player i we use the notation $\langle \sigma_i, \sigma_{-i} \rangle$ or $\langle \sigma_{-i}, \sigma_i \rangle$. The tree corresponding to a strategy profile $\langle \sigma_i \rangle_{i \in Pl}$ is simply $\langle N, \bigcap_{i \in Pl} E_{\sigma_i} \rangle$

A fundamental question of game theory is that of the solution to a game: given a game tree, preference relations for players and possible strategies, which strategy should players choose, which one serves their interests best? A greedy approach would be choosing a strategy, having the highest possible payoff in all its leafs. This is a too simple approach, because it does not take into consideration the other players wishes. If the best terminal history for player i yields a bad outcome for player j, and j can make a move, resulting in a better payoff for j, and the worst payoff for i, i should settle for an outcome with lower payoff, that j likes as well.

Outcomes of strategy profiles, where all players play optimally considering their opponents strategies as well are called *equilibria* in game theory:

Definition 2.4.14 (Nash equilibrium)**.** A strategy profile (Definition 2.4.13) $\sigma = \langle \sigma_i, \sigma_{-i} \rangle$ for a extensive game of perfect information (Definition 2.4.2) $G = \langle Pl, H, Tu, \{\succsim_i\}_{i \in Pl} \rangle$ is a *Nash equilibrium*, iff

$$\forall i \in Pl \; o(\langle \sigma_i, \sigma_{-i} \rangle) \succsim_i o(\langle \tau_i, \sigma_{-i} \rangle)$$

.

Nash introduced the notion of equilibria in his dissertation in 1950. They are usually applied to games in strategic form, where all decisions are made in the beginning, whereas games in extensive form have players decide each time they take a turn. This makes Nash equilibria problematic as solutions to extensive games.

The example game in Figure 2.6 has an implausible Nash equilibrium. The intermediate nodes show the active player A or B, the leafs contain the payoff vector, with A's payoff first. If A takes the action l, B has to choose between actions C and D. A strategy for player i contains an action for every node, where i is active. Here this is one action for each player. Strategy profiles for this game are represented by a pair — the first component is A's action, the second is B's. According to Definition 2.4.14, both $\langle l, D \rangle$ and $\langle r, C \rangle$ constitute Nash equilibria of the game, but $\langle r, C \rangle$ is not a state, where the game would end. By choosing l A gets the highest outcome possible, since B would take move D. However we have $o(\langle r, C \rangle) \succ_A o(\langle l, C \rangle)$, and $\langle r, C \rangle \sim_B \langle r, D \rangle$, making $\langle r, C \rangle$ a Nash equilibrium. The problem here is that C should not be considered a good strategy for B: if the game consisted only of the subtree where B moves, the action would be D and not C.

This leads to *subgames* and *subgame perfect equilibria*, as a better concept for analyzing games in extensive form.

A more intuitive perspective on subgame perfect equilibria can be found in *Fun and Games*: a subgame perfect equilibrium contains the best strategy for each player, even if their opponent makes a mistake and the game arrives at a point it would not have, if the players were following their strategies [Bin92]. A Nash equilibrium is indifferent to these situations, since it only focuses on the payoffs in the terminal node of the strategy profile. Tic-Tac-Toe is a simple example — if both players play optimally the game always ends in a draw. However if a player erroneously marks the wrong box, the opponent suddenly could win the game by changing strategies, whereas the original strategy would still result in a draw. — The player should change to an optimal strategy from the new game state's perspective.

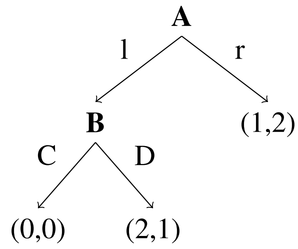


Figure 2.6: A game with implausible Nash equilibria — taken from [OR94]

Definition 2.4.15 (subgame)**.** The *subgame* of an *extensive game* $G = \langle Pl, H, Tu, \{\succsim_i\}_{i \in Pl}\rangle$ that follows history h is the extensive game $G(h) = \langle Pl, H|_h, Tu|_h, \{\succsim_i\}_{i \in Pl}|_h\rangle$, where

- $H|_h$ is the set of sequences possible after initial history h: $h' \in H|_h \Leftrightarrow (h, h') \in H$.
- $Tu|_h(h') = Tu((h, h'))$, for each $h' \in H|_h$
- $h' \succsim_i|_h h'' \Leftrightarrow (h, h') \succsim_i (h, h'')$, for $h', h'' \in H|_h$

Strategies and *strategy profiles* are likewise *restricted* to a subgame following h: $\sigma_i|_h(h') = \sigma_i(h, h')$, and $\sigma|_h = \langle \sigma_i|_h \rangle$.

A *subgame tree* following node n equates a subtree rooted at n (Definition 2.4.10) and is denoted by $\langle N|_n, E|_n\rangle$, subgame strategy trees by $\langle N|_n, E_{\sigma_i}|_n\rangle$.

Definition 2.4.16 (subgame perfect equilibrium)**.** A strategy profile σ (of a game G) is a *subgame perfect (Nash) equilibrium*, iff for every non-terminal history h, the restriction of σ to h is a Nash equilibrium of the subgame following h, viz.

$$\forall h \in H \backslash Z,\ \forall i \in Pl\ s.t.\ Tu(h) = i\ :\ o|_h(\langle \sigma_i|_h, \sigma_{-i}|_h\rangle) \succsim_i|_h o|_h(\langle \tau_i|_h, \sigma_{-i}|_h\rangle)$$

Compare this definition to the one for Nash equilibrium — a subgame perfect equilibrium is a specialization of Nash equilibrium: not only does the outcome have to be optimal for each player, it has to be optimal for each player at every possible state of the game, not only at the initial history. Consider our example of an implausible equilibrium above (Figure 2.6). Only after the initial history l, player B makes a choice between C and D. Playing rationally he would pick D, thus eliminating $\langle r, C\rangle$ as a candidate for a subgame perfect equilibrium and leaving us only with the desired $\langle l, D\rangle$ as equilibrium of the game.

The existence of a subgame perfect equilibrium, is not only present in this toy example, it is a general property of extensive games of perfect information. Every game has such an equilibrium, and its payoff is unique:

Theorem 2.4.1 (Kuhn)**.** Every *finite extensive game of perfect information* (see Definition 2.4.2) has a subgame perfect equilibrium. In each *generic game* (Definition 2.4.3) this equilibrium is unique.

The method used to prove Theorem 2.4.1 is known as *backwards induction*. We do not provide a formal proof, since similar techniques are used for showing the equivalence of our evaluation games to the standard definitions of truth. However a short sketch of the backwards induction provides valuable insights: The argument is an induction on the length len of the longest branch in the game tree. If $len = 0$ the whole game tree consists of a single terminal node, without any actions, and with the payoff vector for the game. Let $len = n + 1$, and let the root of the tree have i as active player. By the induction hypothesis we have the optimal payoff vectors, and corresponding strategies, for all subgames starting in a child-node of the root. Now simply let i choose the action/edge which leads to the subgame which yields the highest payoff for i. The players build up their strategy by starting at the end positions and choosing the action leading to the path with the best outcome for them. Hence the name *backwards* induction.

Backward induction provides us with a constructive procedure for actually finding the existing equilibrium. Furthermore it also is a natural method, as expressed in [OR94]:

> Part of the appeal of the notion of subgame perfect equilibrium derives from the fact that the algorithm describes what appears to be a natural way for players to analyze such a game so long as the horizon is relatively short.

The method illustrates why the uniqueness of the equilibrium can only be guaranteed for *generic games* (Definition 2.4.3). If two paths lead to the same payoff for a player, they are indifferent, which one they take. However the actual payoff value is unique, when using *payoff functions*: The preference relation is the usual order on $\mathbb{R}$: we have $x \leq y$ and $y \leq x \Rightarrow x = y$. As long as we are only interested in the outcome of the game and not which path in the game tree lead to it, we can safely assume that the outcome is unique.

2.4.4 Zerosum Games

The games we analyze in this work fall into the class of finite two-player zerosum games — Hintikka's evaluation game for classical propositional logic is a *win-lose game*, and the many-valued version we provide for KZ-logic is a *constant-sum game*.

Definition 2.4.17 (two-player zerosum game)**.** A two-player game for players $\{1, 2\}$, where we have $h_1 \succsim_1 h_2 \Leftrightarrow h_2 \succsim_2 h_1$, for any two terminal histories $h_1, h_2 \in Z$, is called a *strictly competitive game* or *zerosum game*.

If $\succsim_i$ are given in form of payoff-functions we have $u_1(h) + u_2(h) = 0$. In case the payoff is $u_i(h) \in \{-1, 1\}$, a player either *wins* or *loses* the game and we call it a *win-lose game*.

The class of *finite* two-player zerosum games has some pleasant properties. Theorem 2.4.1 provides us with the existence of an unique payoff, and through backwards induction we have a constructive procedure to obtain one of the branches in the game tree with this payoff.

The additional information that the two players interests are diametrically opposed, additionally simplifies the search for this result: We can focus on payoff of one of the two players. If 1's payoff $u_1(h)$, 2's payoff is simply $-u_1(h)$. Assuming perfectly rational players, who try to maximize their respective payoff, we can paraphrase the choices of player 2: Maximizing their payoff amounts to them trying to minimize 1's payoff.

The *payoff vector* of a terminal history in a zerosum game can therefore be represented by a single real number u — the *value* of this terminal history or run of the game.

The *subgame perfect equilibrium* defines the *value* of the game. It is obtained by considering those strategies, where player 1 tries to maximize u, and player 2 tries to minimize it. This strategy, where a player tries to *maximize* her own profit, while expecting that their opponent tries to *minimize* it is known in the literature as *min-max strategy* [FR12], or a *maxminimizer* [OR94, Bin92].

Our Definition 2.4.12 of a strategy tree provides benefits in understanding the idea behind *min-max strategies*, as exemplified by the two strategy trees drawn in Figure 2.7 and Figure 2.8. The two game trees, have each leaf representing the value of the zerosum game. They represent the two strategies A and B for player 1, corresponding to the single choice of a or b by green arrows. The question is which of the two actions should 1 choose, in order to get the best result? The highest payoff value would be 0.7, following actions (b, L), but this would be the lowest payoff for player 2 (-0.7), and after 1 chooses b it is player 2's choice which of R, M, L to take. Playing to maximize the payoff, 2 would choose M, since this yields a payoff of 1.0 for 2 and -1.0 for player 1. So, expecting player 2 to maximize their own profit, player 1 considers the other strategy (Figure 2.7): the maximal payoff for 1 is only 0.5, but player 2, having to choose between C and D, yielding -0.3 and -0.5 respectively, would choose C. So the rational choice for player 1 is to choose strategy A. Strategy B would leave 1 with -1.0 as payoff. Strategy A corresponds to the subgame perfect equilibrium (a,C) for the game, and thus fixes its value $v = 0.3$. Furthermore it explains the notion of a min-max strategy (Definition 2.4.18): it is the strategy having the highest minimal payoff value. In this example the minimal payoffs reachable in the two strategies are -1.0 and 0.3, and the maximal of those is 0.3.

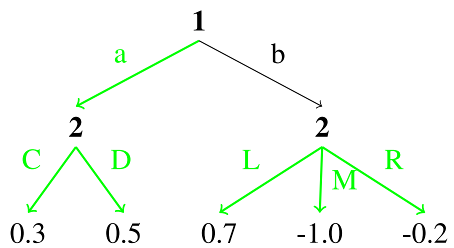


Figure 2.7: strategy A (green arrows) for player 1 for a two-player zerosum game of perfect information

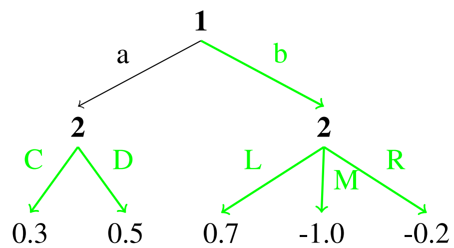


Figure 2.8: strategy B (green arrows) for player 1 for a two-player zerosum game of perfect information

Definition 2.4.18 (min-max strategy)**.** Given a two-player zerosum game tree $\langle N, E\rangle$, *min-max strategy* for player i is a strategy which satisfies the following condition:

$$\max_{\sigma_i \in \Sigma_i} \min_{n \in N_{0,\sigma_i}} (l_{payoff}(n))$$

The unique value $l_{payoff}(n)$ resulting by a player i playing her min-max strategy is called the *value* of the game v, or the *enforceable payoff* for player i.

A class of games, seeming like a generalization of zerosum games are *constant-sum games*, where the players' utilities add up to some constant c. They are treated in Binmore's textbook [Bin92]. Take two people dividing a cake fairly with the "you cut", "I choose" procedure[27] as an example. Assume the cake weighs 200 g, and that player 1 makes the cut. The procedure can be formalized as a game with infinitely many moves — we represent the whole cake by the real interval $[0, 200]$ and each move splits it into two sub-intervals ($[0, d]$ and $(d, 200]$). Player 2 has the choice of taking d g cake or $200 - d$ g cake. Assuming that both players want the greatest amount of cake for themselves the subgame perfect equilibrium is reached if player 1 chooses $d = 100$. The game is displayed in Figure 2.5. By using d as utility function, $u_1(h) \neq -u_2(h)$, which would be required. However This can easily be repaired by subtracting c from one player's utility function. In our example we have at half the terminal histories $u_1(h) = d$ and $u_2(h) = 200 - d$. Observing that $c = 200$ we get the new utility function for player 2 as $u_2(h) = 200 - d - 200 = -d$ and thus a zerosum game.

We mention constant-sum games, because the evaluation game for KZ-logic, we introduce, is most naturally seen as a constant-sum game with $c = 1$. The cake in those games is the truth.

As a special case of zerosum games *win-lose games* have been historically the focus of many early works on game theory. This is probably mostly due to the fact, that many actually existing games, like chess and Tic-Tac-Toe belong to this category. A player either wins, loses or draws at a specific run of those games, however the actual payoff is of no importance. A widely quoted result of those early treatments of win-lose games is accredited to Zermelo [OR94, BvBW07, vB03]:

Corollary 2.4.2 (Zermelo)**.** Every finite two-player *win-lose game* is determined, i.e. one of the players has a *winning strategy*.

This precise statement differs somewhat from the result actually established by Zermelo, namely that at a given game of chess both players have strategies which result at least in a draw. The context of the work where the version above (Corollary 2.4.2) was postulated, mainly deals with evaluation games for classical logic, which are games where no draw is possible — a formula is either true or it is false under a given valuation [BvBW07, vB03, vB02].

We introduce Zermelo's theorem as a corollary to Kuhn's theorem (Theorem 2.4.1), because it is the same result for the special case of win-lose games. The former proves the existence of a certain strategy profile with a specific outcome, the latter deals with a strategy for one of two

[27] This procedure generalized to n players is used in an article by Pauly and Parikh, where it is used to motivate game logic [PP03].

players. Both agree on the outcome: a winning strategy for player i is a strategy ensuring that all strategy profiles it is part of end in a state, where i wins.

The logical evaluation games for classical logic are with win-lose games, reflecting the law of excluded middle. We slightly change the labeling of the game trees (Definition 2.4.11), and provide a formal definition of a *winning strategy*.

Definition 2.4.19 (win-lose game tree)**.** A win-lose game tree is a *game tree* (Definition 2.4.11) for a two-player win-lose game, where l_{payoff} is replaced by two labeling functions for the two players respectively:

- $l_{win_1} : N_0 \rightarrow \{win, lose\}$
- $l_{win_2} : N_0 \rightarrow \{win, lose\}$

having the following properties:

- $l_{win_1}(n) = win \Leftrightarrow l_{win_2}(n) = lose,$
- $l_{win_1}(n) = lose \Leftrightarrow l_{win_2}(n) = win,$

Note that, this is only a matter of presentation, and does not entail any fundamental changes on the formal level: our win-lose games still can quite easily be transformed into standard zero-sum games: just consider that $l_{win_1}(n) = win \Leftrightarrow l_{payoff}(n) = 1$.

The concept of a winning strategy should be clear: in a win-lose game the min-max strategy, for each player, is optimal. We say that player 1 has a *winning strategy*, iff the strategy guarantees that, no matter what her opponent plays, she wins. In the tree representation this is a strategy tree where all leaf-nodes, which are still connected to the root, have $l_{win_1}(n) = win$.

Definition 2.4.20 (winning strategy)**.** A winning strategy tree $\langle N, E_{\sigma_i} \rangle$ (Definition 2.4.12) for player i is a strategy tree where all terminal nodes connected to the root are winning nodes for i: $\forall n \in N_{0,\sigma_i} : l_{win_i}(n) = win$

2.5 Logical Evaluation Games

In [Hin73b] Hintikka presented an alternative characterization, to the Tarskian notion of truth: what does it mean for a first-order formula to be true in a given model. Instead of semantically analyzing the formula to arrive at a truth value one can construct a two-person zerosum extensive game of perfect information for a given formula and a given model. While the merit of such an alternative presentation may not be obvious at first glance — after all there already is a precise definition of the notion of truth — there is much to be gained from taking this alternative view.

Originally Henkin used an informal game as a way to argue, that certain infinite formulas can be understood intuitively, although the standard definition only deals with formulas of finite length [Hod13, Hin73a].

Hintikka showed that the game can be formalized as a zero-sum game of perfect information, which opened the possibility to look at the semantics of classical logic through the rich and well-developed apparatus of formal game theory.

From this perspective questions, like: “What if it were a game of imperfect information”, pose themselves naturally. This particular question led to research in logics with branching quantifiers. For example a formula $\forall x \exists (y/x)(R(x,y)$, represents the statement, for all x there exists a y, not depending on x, such that $R(x,y)$. Although the step seems small the resulting *independence friendly logic* or *IF-logic* is stronger and closer to second-order logic with standard semantics than to first-order logic in terms of expressiveness [Hin82].

The law of excluded middle translates to determinism in game theoretic terms, a quality of games, which is highly non-trivial, raising the question of the outcome of dropping the requirement of determinism.

Another important gain of the game-theoretic approach lies in its intuitivity — it helps in understanding the concepts of quantification, when you see it as choosing objects, to either help your argument, or to make your opponents argument invalid. As described in [Hod13], there were at least two implementations of these games, which proved very effective as a means to teach first-order semantics. one developed in Stanford, and the second developed independently for a school of gifted children in Omsk.

Hintikka calls his game, a game of seeking and finding objects, in order to make a certain formula true or false. He argues that semantic games from this perspective are very well suited to give a non self-referential meaning to logical quantifiers and connectives. Tarskian truth conditions cannot provide a genuinely new understanding for both, in order to understand what it means for the statement $\varphi \wedge \psi$ to be true, they refer to the word “and” in the meta-language. That the verbs seek and find indeed capture the meaning of quantifiers is exemplified in [Hin73a] by restating the sentence: “All swans are white” as “No swan can be found that is not white”.

The game has the elegant peculiarity of completely characterizing a game state by the formula and an assignment of domain objects to variables in the first-order case.

We reproduce Hintikka’s presentation of his games, which he gave mainly through the rules for the two players, called me and Nature by him.

2.5.1 Hintikka’s Evaluation Game for Classical Logic

In 1968 Hintikka’s article *Language-Games for Quantifiers* [Hin73a][28] introduces the game as an extension to an idea of Henkin, who only analyzed formulas in prenex normal form[29]. There connectives are also explained via game rules:

> ...My aim is to end up with a true atomic sentence. The rules for quantifiers remain the same. Disjunction now marks my move: I have to choose a disjunct with reference to which the game is continued. Conjunction marks a move by my opponent: he chooses a conjunct with reference to which the game is continued. Negation $\sim F$ has the effect of changing the roles of the players, after which the game continues with reference to F.

The rules are explicitly listed in *Quantifiers, Language-Games and Transcendental Arguments* [Hin73c]. We adapt the symbols for the connectives and quantifiers to our notation, for

[28] Reprinted in [Hin73b]

[29] All quantifiers occur only in the beginning and are followed by a quantifier free part, referred to as the matrix.

example writing $\neg$ instead of $\sim$, and using φ and ψ to denote formulas, leaving the remaining formulation as in the original.

Definition 2.5.1 (Hintikka's rules for his original semantic games)**.** Given a domain D and a formula φ the rules of the game are as follows

G.$\exists$ If φ is of the form $\exists x\psi$, I choose a member of D, give it a name, say 'n' (if it did not have one before). The game is continued with respect to $\psi[x \mapsto n]$. Here of course $\psi[x \mapsto n]$ is the result of substituting n for x in ψ.

G.$\forall$ If φ is of the form $\forall x\psi$, Nature likewise chooses a member of D.

G.$\vee$ If φ is of the form $\psi_1 \vee \psi_2$, I choose ψ_1 or ψ_2, and the game is continued with respect to it.

G.$\wedge$ If φ is of the form $\psi_1 \wedge \psi_2$, Nature likewise chooses ψ_1 or ψ_2.

G.$\neg$ If φ is of the form $\neg\psi$ the game is continued with respect to ψ with the roles of the two players interchanged.

In a finite number of moves, an expression A of the form $P(n_1, n_2, \ldots, n_k)$ will be reached, where P is a k-adic predicate defined on D. since $n_1, n_2, \ldots, n_k$ are members of D, A is either true or false. If it is true, I have won and Nature has lost; otherwise Nature has won and I have lost.

A game in extensive form as a tree can be seen in Figure 2.9. The nodes describe which player is to choose the conjunct or disjunct at a given state, and the colors indicate how often the roles of the players were interchanged, due to a negation. In the blue nodes I would win, if the asserted formula is true, in the red nodes Nature would win.

Comparing the rules as given by Hintikka, to the formal requirements for a game in our sense, some issues can be identified. Especially the rule for negation does not fit into the standard definitions. Hintikka recognized this and addresses it, by saying that the negation rule, or the exchange of labels is not necessary, since any formula can be transformed into an equivalent one in negation normal form [Hin73c]. However he also points out that the rule for universal quantification and conjunction seems a bit unnatural, in having Nature actively choosing objects. This can be avoided by dropping the universal quantifier and defining it in terms of the existential quantifier: $\forall x\varphi \Leftrightarrow \neg\exists x\neg\varphi$, which in turn makes the role change essential for the game. The problem of Nature having to play an active role after a role change is not addressed directly.

2.5.2 Hintikka Games for Many-Valued Logics

Semantic characterizations of mathematical fuzzy logic through game theory have been examined in literature quite extensively, focusing primarily on Łukasiewicz logic as the most prominent fuzzy logic. [Fer14, FR12, MC09, FM09, Gil77].

One common aspect of all treatments is that they lose the elegant minimal game states of Hintikka's original game being defined just by the formula under consideration.

Transforming Hintikka's game for classical logic to a fuzzy setting for evaluating formulas in full Łukasiewicz logic cannot be achieved in a straight-forward manner, as was captured in an

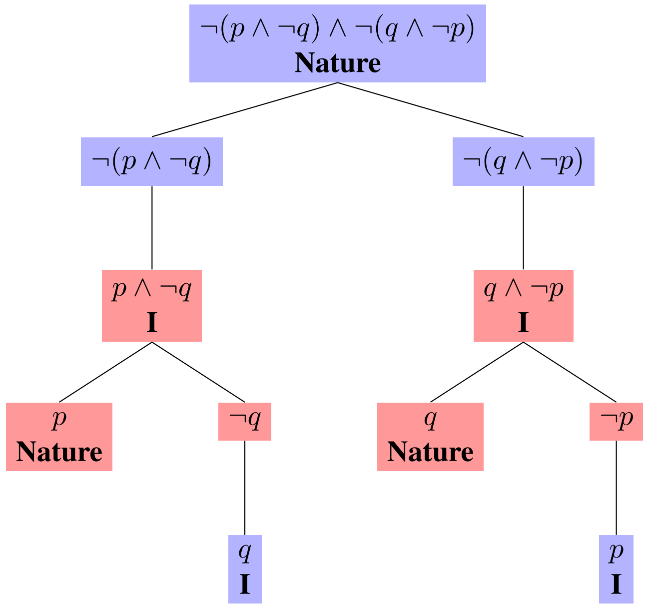


Figure 2.9: Hintikka evaluation game for $\neg(p \wedge \neg q) \wedge \neg(q \wedge \neg p)$ – the labels indicate which player chooses the subformula, the colors show which player wins if the atom is true: blue for my win, red for Nature's win

article by Fermüller [Fer14]. The problem is rooted in the semantics of the strong connectives. A rule for evaluating strong conjunction, disjunction or implication cannot only refer to one of its subformulas.

However, if we restrict ourselves to the weaker subset formed by KZ-logic the fundamentals of the game need only slight adaptation, as was shown in the article by Fermüller and Roschger [FR12]. The rules for the logical connectives remain unchanged. Only the rule for the atomic case needs reevaluation. In KZ-logic we are dealing with real-valued values for (atomic) formulas, so a direct game characterization of KZ formulas is best modeled with real-valued payoffs $r \in [0, 1]$

The most fitting notion from game theory for these games is the constant-sum game: the value r of a particular terminal node in the game tree then simply is the evaluation of the atom asserted in that node $r = \nu_{KZ}(a)$. This leads us to a formulation of a rule for the atomic case:

Definition 2.5.2 (game rules for atomic KZ formulas)**.** :

$\mathbf{R}_{\mathcal{A}tom_{KZ}}$ If φ is atomic, and evaluates to r in the given valuation, the game ends with a value of r for me.

2.5.3 Evaluation Games for Full Łukasiewicz Logic

We want to highlight two approaches for giving game-theoretic semantics to full Łukasiewicz logic. On the one hand we have the evaluation games introduced by Cintula and Majer [MC09], oriented at extending Hintikka's game-theoretic semantics to fuzzy logic. The games are initially still formulated as win-lose games, with the two players betting that the considered formula has a certain value r. If the atomic formula evaluates to a truth value $\geq r$ the current verifier wins, else the falsifier wins. Additionally there is a dedicated rule if r is 0, making it a win for the verifier. The rules for the lattice connectives remain classical — they do not change the value r. The strong connective rules and the quantifier rules do modify r, reflecting the need for the more complicated game states.

On the other hand there is the approach first introduced by Giles in 1977, motivating the use of non-classical logics in physics [Gil77]. This was linked to fuzzy proof theory by Fermüller and Metcalfe [FM09]. While referring to dialogue games by Lorenzen, used originally to describe validity in intuitionistic logic, the formulation actually characterizes an evaluation game, referring to a given valuation[FR12]. Giles' games consist of two separate phases — one for decomposing a formula, and one for betting on "elementary experiments" (corresponding to atomic formulas). Game states are represented by multisets for both players, reflecting the sub-structural aspect of Łukasiewicz logic.

CHAPTER 3

Formalizing Logic Games

This chapter introduces two Hintikka-style evaluation games: $\mathcal{H}$-game for classical propositional logic, and $\mathcal{H}$-mv-game for KZ-logic[1]. In our definition of the games we address certain ambiguities found in the original presentations of the games, when viewed from a basic game-theoretic perspective.

We align the games with Hintikka's original games, in choosing conjunction, disjunction and negation as set of connectives. In the many-valued case this corresponds to KZ-logic and enables us to keep the characterization of a game state simple, since it only needs to refer to the formula currently considered and to the player trying to verify it.

Our focus is restricted to propositional logic in accordance with our aim of describing the game trees by logic axioms, without the need of tracking the chosen domain elements for the quantifier rules. This simplifies the winning condition from Definition 2.5.1, to "In a finite number of moves, an expression $a \in \mathcal{Atom}$ will be reached. a is either true or false. If it is true, I have won and Nature has lost; otherwise Nature has won and I have lost."

The rules for the game stated by Hintikka intuitively describe a extensive game of perfect information. Given a formula and an atomic valuation it should be possible to construct a win-lose game tree according to Definition 2.4.19, and set the win labels, according to an atomic valuation.

However, the rule for negation $R_\neg$, does not state explicitly, whose turn it is and the concept of exchanging roles of two players does not fit game-theoretic terms either.

A more formal description of the concept of role-exchanging can be found in [vB03]: if the outermost connective is a negation the game tree is dualized — the win-lose markings and the turn indicators are reversed, the current formula is syntactically dualized, by exchanging $\wedge$ and $\vee$, and inversing the polarity of atomic formulas.

This characterization is more explicit in the working of negation as an action, but still has the disadvantage of dealing with it on a meta level — rewriting the game tree below a negated formula by dualizing cannot be a single move.

[1] The naming of $\mathcal{H}$-game and $\mathcal{H}$-mv-game was taken from [FR12] and [Fer14].

3.1 Adapting the Game Rules

We address the mismatch between precise game theoretic notions and the rules for logical evaluation games by shifting the view on the games. In addition to letting the two players *seek* and *find* certain objects — elements of the domain in the first-order case, the right conjunct or disjunct in the propositional case — we give the players a role. At each game state, a player is either an *asserter*, or an *attacker* of the current formula. We use *defender* as a synonym for asserter and *asserting, defending, attacking* in this sense. Players can change roles between states.

This makes negation as the outermost connective a simple role change. To make this an action of one player, we arbitrarily define this to be a move by the player asserting the negated formula. Appealing to intuition we say that the asserter *gives* the formula to their opponent. For this reason we choose to name the players you and me instead of Nature and me.

This concept is known from another class of logical games, the dialogue games introduced by Lorenzen and Lorenz, and used extensively by Giles in his game for Łukasiewicz logic [Gil77]. These games are used to show the validity of a formula, irrespective of a given valuation, a change we are not adopting.

Our game has three actions available in the respective states: {R,L,Neg}. R and L indicate that the active player chooses the right or left subformula of a disjunction or conjunction, and Neg is indicating a role change. The game tree in Figure 3.1 exemplifies the changes to Hintikka's original presentation; the levels introduced in Figure 2.9 are not necessary anymore, since negation is a regular move.

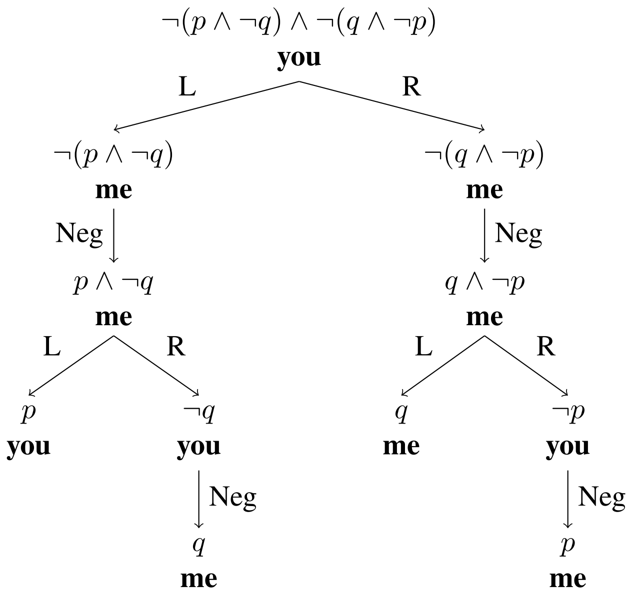


Figure 3.1: Evaluation game with negation as action for the Hintikka game in Figure 2.9.

3.2 $\mathcal{H}$-game for Classical Logic

The following presentation is inspired by [vB02] and [FM09]. We add some labeling functions to the ones for turns, moves and the outcome to our Definition 2.4.11 of game trees, representing the central components of evaluation games: the roles of the two players and the currently asserted formulas.

The role exchange implies that our definition needs to be formulated agnostic of a particular player and their *dual player* or opponent. We write d to represent a player $d \in \{you, me\}$, and d' for the dual player: if $d = you$, $d' = me$, and if $d = me$, $d' = you$.

We arrive at a formal definition of the game tree for a given formula and valuation:

Definition 3.2.1 ($\mathcal{H}$-game on formula φ over classical valuation ν_{CL})**.** Given an initial defender $d \in \{you, me\}$ the game tree $G_{\varphi,\nu_{CL},d} = \langle N, E\rangle$ is defined inductively over the *structure* of φ. We write r_φ for the root of the game tree and omit ν_{CL} and d if they are clear from the context.

We use five node labeling functions and one edge labeling function:

- $l_{form} : N \to \mathcal{Prop}$
- $l_{turn} : N \backslash N_0 \to \{you, me\}$
- $l_{assert} : N \to \{you, me\}$
- $l_{win_{me}} : N_0 \to \{win, lose\}$
- $l_{win_{you}} : N_0 \to \{win, lose\}$
- $l_{move} : E \to \{L, R, Neg\}$

1. φ is an $\mathcal{Atom}$ a: $G_{a,\nu_{CL},d}$ consists of a single *leaf* r_a with:
   - $l_{form}(r_a) = a$
   - $l_{assert}(r_a) = d$
   - $l_{win_d}(r_a) = \begin{cases} win & \text{iff } \nu_{CL}(a) = 1 \\ lose & \text{iff } \nu_{CL}(a) = 0 \end{cases}$
   - $l_{win_{d'}}(r_a) = \begin{cases} lose & \text{iff } \nu_{CL}(a) = 1 \\ win & \text{iff } \nu_{CL}(a) = 0 \end{cases}$
2. $\varphi = \psi_1 \vee \psi_2$: given game trees $G_{\psi_1,\nu_{CL},d}$ and $G_{\psi_2,\nu_{CL},d}$ the tree for $\varphi = \psi_1 \vee \psi_2$ is constructed from a new root r_φ with:
   - $l_{form}(r_\varphi) = \varphi = \psi_1 \vee \psi_2$
   - $l_{assert}(r_\varphi) = d$
   - $l_{turn}(r_\varphi) = d$

   and two edges to the two subtrees for ψ_1 and ψ_2 respectively:

- $l_{move}((r_\varphi, r_{\psi_1})) = L$
- $l_{move}((r_\varphi, r_{\psi_2})) = R$.

3. $\varphi = \psi_1 \wedge \psi_2$: given game trees $G_{\psi_1,\nu_{CL},d}$ and $G_{\psi_2,\nu_{CL},d}$ the tree for $\varphi = \psi_1 \wedge \psi_2$ is constructed from a new root r_φ with:

   - $l_{form}(r_\varphi) = \varphi = \psi_1 \wedge \psi_2$
   - $l_{assert}(r_\varphi) = d$
   - $l_{turn}(r_\varphi) = d'$

   and two edges to the two subtrees for ψ_1 and ψ_2 respectively:

   - $l_{move}((r_\varphi, r_{\psi_1})) = L$
   - $l_{move}((r_\varphi, r_{\psi_2})) = R$.

4. $\varphi = \neg\psi$: given $G_{\psi,\nu_{CL},d'}$, with r_ψ as its root, the tree for $\varphi = \neg\psi$ is constructed from a new root r_φ with:

   - $l_{form}(r_\varphi) = \varphi = \neg\psi$
   - $l_{assert}(r_\varphi) = d$
   - $l_{turn}(r_\varphi) = d$

   and an edge to the subtree for ψ:

   - $l_{move}((r_\varphi, r_\psi)) = Neg$

A game node is uniquely determined by the formula asserted in l_{form}. We refer to *subgames* (see Definition 2.4.15) by writing $\langle N|_\varphi, E|_\varphi\rangle$, for the subgame rooted at n, with $l_{form}(n) = \varphi$.

The game is a *win-lose game* according to Definition 2.4.19.

An example game tree $G_{\neg(p\wedge\neg q)\wedge\neg(q\wedge\neg p),\nu_{CL},me}$, with $\nu_{CL}(p) = 1$ and $\nu_{CL}(q) = 1$, can be seen in Figure 3.2[2]: the node labels are written inside the squares representing the nodes, and the move labels are written next to the edges. In the root node the outermost connective is a conjunction, making you the active player. Both child nodes have a negation at the root position, resulting in a swap of the assert label in their children respectively.

The valuation ν_{CL} plays a role only in the leaf nodes, where the win-labels are assigned based on the current defender and the value of the atom considered. The left-most terminal node in Figure 3.2 has you as assert label, and p as atom, which is true in ν_{CL}, thus making you the winner in this state, and me the loser.

Observing that $\neg(p \wedge \neg q) \wedge \neg(q \wedge \neg p)$ is equivalent to $p \leftrightarrow q$, the formula evaluates to true, iff p and q have the same truth value, like in the given ν_{CL}. By *Hintikka's theorem*, there has to be a winning strategy for me as initial asserter in $G_{\neg(p\wedge\neg q)\wedge\neg(q\wedge\neg p),\nu_{CL},me}$.

[2] The fully formalized game tree from Figure 3.1

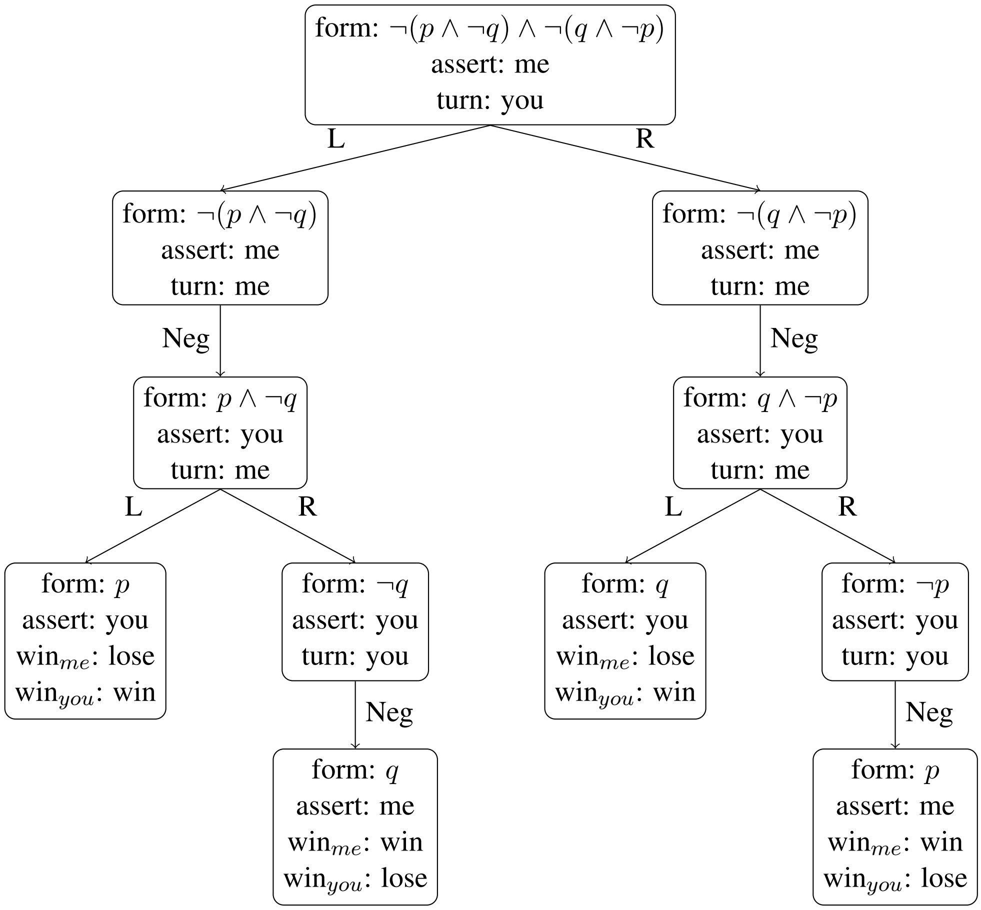


Figure 3.2: Hintikka evaluation game $G_{\neg(p\wedge\neg q)\wedge\neg(q\wedge\neg p),\nu_{CL},me}$ from Figure 2.9 as a decorated tree according to Definition 3.2.1 for $\nu_{CL}(p) = 1, \nu_{CL}(q) = 1$

We have illustrated one (and in this case the only) winning strategy for me in Figure 3.3, indicated by the green arrows in the game tree. The indicated subtree is indeed a strategy for me. The only states with an actual choice for me are the nodes $p \wedge \neg q$ and $q \wedge \neg p$, both having you as a defender. My interest is picking an action leading to a subgame, where the asserted formula is false — $\neg p$ and $\neg q$ respectively. In both subgames you have to give me the atom, because of the negation, leaving me in two terminal states, defending a true atom — a win for me. From an outside point of view, the strategy is a winning strategy, because all leaves, still connected to the root of the game tree are winning positions for me (see Definition 2.4.20).

The game just defined is sufficiently different from the original presentation, and we need to prove that our version still has the property, that it connects truth in a valuation for a formula φ with a winning strategy in the game:

Lemma 3.2.1. Iff $\nu^*_{CL}(\varphi) = 1$, then the initial defender d has a winning strategy $\langle N, E_{\sigma_d}\rangle$ (see Definition 2.4.20) for the $\mathcal{H}$-game on φ over ν_{CL} $G_{\varphi,\nu_{CL},d} = \langle N, E\rangle$.

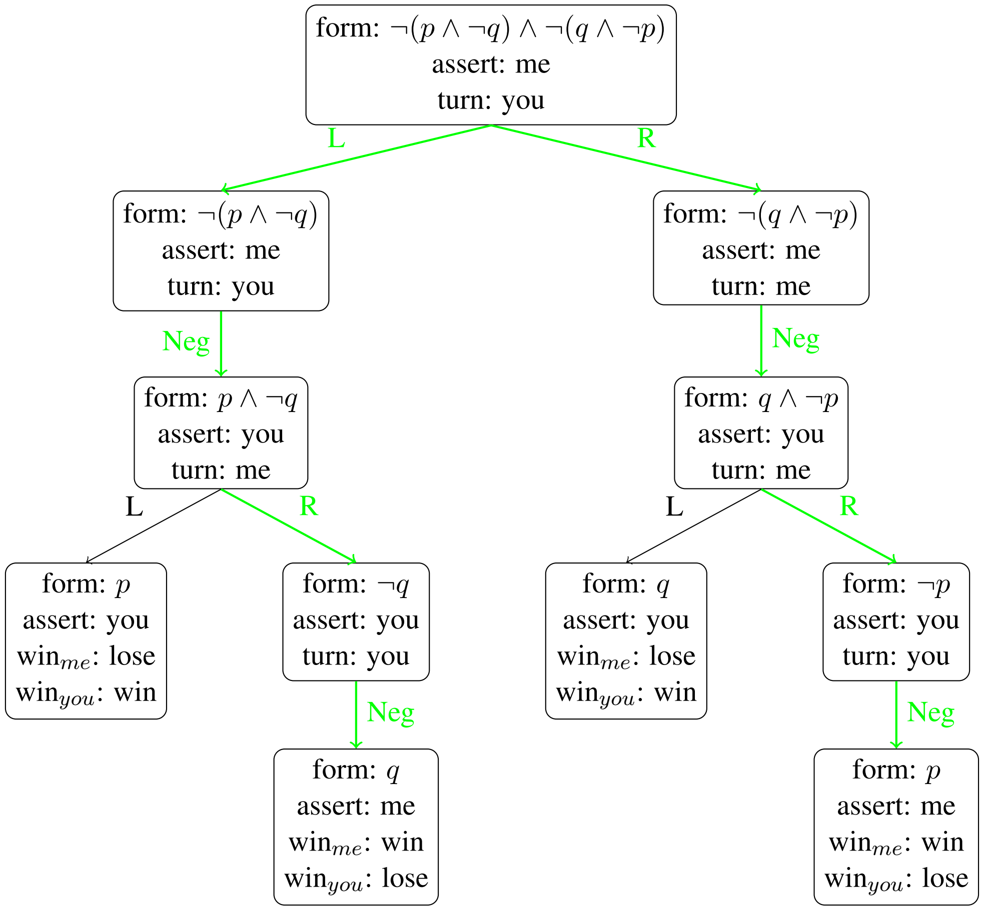


Figure 3.3: Winning strategy for me in the game from Figure 3.2, indicated by the green edges.

Proof. We show that d has a winning strategy, if $\nu^*_{CL}(\varphi) = 1$ and that d does not have one if $\nu^*_{CL}(\varphi) = 0$ simultaneously, by structural induction on the number of connectives in φ. Assume $d = me$, the case for $d = you$ is symmetric.

Base case: $\varphi = a \in \mathcal{Atom}$.

- $\nu_{CL}(a) = 1$: $G_{a,\nu_{CL},me}$ consists only of a leaf-node r_a with $l_{assert}(r_a) = me$ and $l_{win_{me}}(r_a) = win$, thus the only, trivial, winning strategy (Definition 2.4.12) consists of a single winning node for me.

- $\nu_{CL}(a) = 0$ and $l_{assert}(r_a) = me$. This implies $l_{win_{me}} = lose$, which is a losing strategy for me.

In both cases my formal strategy tree is $\langle \{r_a\}, \emptyset \rangle$.

For the inductive step assume that d has a winning strategy σ_d in all $\mathcal{H}$-games $G_{\varphi,\nu_{CL},d}$, if $\nu^*_{CL}(\varphi) = 1$ and that d does not have a winning strategy if $\nu^*_{CL}(\varphi) = 0$, where $comp(\varphi) = k$ (Definition 2.1.2). We have to distinguish 3 cases, each with two sub-cases for $comp(\varphi) = k+1$:

- $\varphi = \psi_1 \vee \psi_2$:
  - $\nu^*_{CL}(\psi_1 \vee \psi_2) = 1$: either $\nu^*_{CL}(\psi_1) = 1$, or $\nu^*_{CL}(\psi_2) = 1$, or both. $comp(\psi_1) \leq k$ and $comp(\psi_2) \leq k$.

    If $\nu^*_{CL}(\psi_1) = 1$, I have a winning strategy $\sigma_{me} = \langle N|_{\psi_1}, E_{\sigma_{me}}|_{\psi_1}\rangle$ in $G_{\psi_1,\nu_{CL},me}$ by the induction hypothesis . By playing L, I arrive at r_{ψ_1} and win if I continue playing according to this winning strategy.

    My winning strategy in $G_{\varphi,\nu_{CL},me}$ is the tree $\sigma_{me} = \langle N, E_{\sigma_{me}}|_{\psi_1} \cup \{(r_\varphi, r_{\psi_1})\}\rangle$.

    The case $\nu^*_{CL}(\psi_2) = 1$ is symmetric with my choice of playing R.

    In case both $\nu^*_{CL}(\psi_1) = 1$ and $\nu^*_{CL}(\psi_2) = 1$ I may chose arbitrarily between the winning strategies for both subgames.
  - $\nu^*_{CL}(\psi_1 \vee \psi_2) = 0$: both $\nu^*_{CL}(\psi_1) = 0$ and $\nu^*_{CL}(\psi_2) = 0$. By the induction hypothesis neither do I have a winning strategy in $G_{\psi_1,\nu_{CL},me}$ nor in $G_{\psi_2,\nu_{CL},me}$.

    No matter what I play, we end up in a game where I lose.
- $\varphi = \psi_1 \wedge \psi_2$:
  - $\nu^*_{CL}(\psi_1 \wedge \psi_2) = 1$: the asserted formula being a conjunction makes you the active player. In order to win the game, I have to have winning strategies for both your possible actions.

    From $\nu^*_{CL}(\psi_1 \wedge \psi_2) = 1$, we have that both $\nu^*_{CL}(\psi_1) = 1$ and $\nu^*_{CL}(\psi_2) = 1$.

    Given that both $comp(\psi_1) \leq k$ and $comp(\psi_2) \leq k$, by the induction hypothesis I have winning strategies for both subgames $G_{\psi_1,\nu_{CL},me}$ ($\sigma_{me} = \langle N|_{\psi_1}, E_{\sigma_{me}}|_{\psi_1}\rangle$) and $G_{\psi_2,\nu_{CL},me}$ ($\sigma_{me} = \langle N|_{\psi_2}, E_{\sigma_{me}}|_{\psi_2}\rangle$), just as required.

    Formally my strategy tree is $\langle N, E_{\sigma_{me}}|_{\psi_1} \cup E_{\sigma_{me}}|_{\psi_2} \cup \{(r_\varphi, r_{\psi_1}), (r_\varphi, r_{\psi_2})\}\rangle$.
  - $\nu^*_{CL}(\psi_1 \wedge \psi_2) = 0$: at least one of $\nu^*_{CL}(\psi_1) = 0$ or $\nu^*_{CL}(\psi_2) = 0$. Assume $\nu^*_{CL}(\psi_1) = 0$. You play L, since this leads us to the game $G_{\psi_1,\nu_{CL},me}$, where I do not have a winning strategy by the induction hypothesis.

    The case $\nu^*_{CL}(\psi_2) = 0$ is symmetric with you playing R. If both $\nu^*_{CL}(\psi_1) = 0$ and $\nu^*_{CL}(\psi_2) = 0$, you may choose arbitrarily between R and L, because you have winning strategies for both subgames.
- $\varphi = \neg\psi$:
  - $\nu^*_{CL}(\neg\psi) = 1$: the asserted formula being a negation means that we switch roles, and the game continues with you being the defender of ψ. We have $\nu^*_{CL}(\psi) = 0$ and $comp(\psi) = k$.

    By the induction hypothesis you do not have a winning strategy for $G_{\psi,\nu_{CL},you}$. Since the game is a win-lose game, I have a winning strategy $\sigma_{me} = \langle N|_\psi, E_{\sigma_{me}}|_\psi\rangle$ for $G_{\psi,\nu_{CL},you}$.

    By playing *Neg* I arrive at a game, which I win if I play according to $\langle N|_\psi, E_{\sigma_{me}}|_\psi\rangle$.

    Formally my winning strategy tree is $\sigma_{me} = \langle N, E_{\sigma_{me}}|_\psi \cup \{(r_\varphi, r_\psi\}\rangle$.

- $\nu^*_{CL}(\neg\psi) = 0$: I have no choice but to play Neg, through which we switch roles, and you defend ψ, with $\nu^*_{CL}(\psi) = 1$.

  By the induction hypothesis you have a winning strategy in $G_{\psi,\nu_{CL},you}$ and thus I lose the game as required.

We have established the connection between truth of a propositional formula under a given classical valuation and the existence of a winning strategy for the player defending that formula, in our $\mathcal{H}$-games. □

This is Hintikka's well-known result:

Corollary 3.2.2 (Hintikka)**.** From Lemma 3.2.1 we deduce that the following two statements are equivalent:

- $\nu_{CL}(\varphi) = 1$
- I have a winning strategy in $G_{\varphi,\nu_{CL},I}$.

3.3 $\mathcal{H}$-mv-game for KZ-Logic

As indicated above in subsection 2.5.2 Hintikka-style evaluation games can readily be extended to a fuzzy setting, if we restrict ourselves to KZ-logic, by changing the game rule for the atomic case, and transforming the *win-lose game* into a *constant-sum game*. We adapt our $\mathcal{H}$-games to $\mathcal{H}$-mv-games and prove their adequateness for KZ-logics:

Definition 3.3.1 ($\mathcal{H}$-mv-game on formula φ over many-valued valuation ν_{KZ})**.** The labeling functions for $\mathcal{H}$-mv-games , are the same as those for classical $\mathcal{H}$-games (Definition 3.2.1), with the exception that $l_{win_{me}}$ and $l_{win_{you}}$ are replaced by two payoff labels:

$$l_{payoff_{me}}, l_{payoff_{you}} : N_0 \to [0, 1]$$

The game tree $G_{\varphi,\nu_{KZ},d}$ is defined as:

1. φ is an $\mathcal{A}tom$ a : $G_{a,\nu_{KZ},d}$ is a single leaf r_a with:
   - $l_{form}(r_a) = a$
   - $l_{assert}(r_a) = d$
   - $l_{payoff_d}(r_a) = \nu_{KZ}(a)$
   - $l_{payoff_{d'}}(r_a) = 1 - \nu_{KZ}(a)$
2. The clauses for compound formulas remain as in Definition 3.2.1, items: 2 – 4

Even though it is technically not necessary to give the payoff values for both players explicitly[3], and we deviate from Definition 2.4.11 in representing the payoffs as two separate labels,

[3] In a constant-sum two player game, one player's payoff defines the other's.

instead of one label with a vector of reals, it is evident, that the game presented is a constant-sum game.

$\mathcal{H}$-mv-games characterize KZ-logic analogously to the way $\mathcal{H}$-games characterize classical propositional logic: the initial defender has an r-valued strategy for a game on φ over ν_{KZ}, iff $\nu^*_{KZ}(\varphi) = r$. We present an explicit proof for the result, since it is a generalization of Corollary 3.2.2.

Theorem 3.3.1. A formula φ evaluates to r in a KZ valuation ν_{KZ}, iff the $\mathcal{H}$-mv-game on φ over ν_{KZ} $G_{\varphi,\nu_{KZ},d} = \langle N, E\rangle$ has value r for the initial defender d.

Proof. The argument proceeds by induction on $comp(\varphi)$: Assume $\nu^*_{KZ}(\varphi) = r$ and $d = me$ (the case $d = you$ is symmetric).

If $\varphi = a \in \mathcal{Atom}$, then $G_{a,\nu_{KZ},me} = \langle\{r_a\}, \emptyset\rangle$. By definition we have $l_{payoff_{me}} = r$, which constitutes the value of $G_{a,\nu_{KZ},me}$. My r-valued strategy is $\sigma_{me} = \langle r_a, \emptyset\rangle$.

For the inductive step assume that for each φ, s.t. $comp(\varphi) \leq k$ and $\nu^*_{KZ}(\varphi) = r$, d has a strategy (the min-max strategy from Definition 2.4.18) for $G_{\varphi,\nu_{KZ},d}$, with the value of $G_{\varphi,\nu_{KZ},d}$ being r. We have to consider three cases:

1. $\varphi = (\psi_1 \vee \psi_2)$: we have $\max(\nu^*_{KZ}(\psi_1), \nu^*_{KZ}(\psi_2)) = r$, since $\nu^*_{KZ}(\psi_1 \vee \psi_2) = r$. Assume $\nu^*_{KZ}(\psi_1) = r \geq \nu^*_{KZ}(\psi_2)$. Given a choice, I will play L to maximize my payoff. By the induction hypothesis I have a strategy $\sigma_{me} = \langle N|_{\psi_1}, E_{\sigma_{me}}|_{\psi_1}\rangle$ in $G_{\psi_1,\nu_{KZ},me}$, s.t. $o(\sigma) = o(\sigma_{me}, \sigma_{you}) = r$. By playing L followed by σ_{me} I can force the game to have a value of r for myself. My formal strategy is $\langle N, E_{\sigma_{me}}|_{\psi_1} \cup \{(r_\varphi, r_{\psi_1})\}\rangle$.

   In case $\nu^*_{KZ}(\psi_2) = r$, I would similarly maximize my payoff to a value of r, by playing R, resulting in $\sigma_{me} = \langle N, E_{\sigma_{me}}|_{\psi_2} \cup \{(r_\varphi, r_{\psi_2})\}\rangle$.

   Should we have $\nu^*_{KZ}(\psi_1) = \nu^*_{KZ}(\psi_2) = r$, I can choose arbitrarily between L and R, and both of my strategies would yield a value of r.

2. $\varphi = (\psi_1 \wedge \psi_2)$: at r_φ you are to move and my strategy tree contains both edges (r_φ, r_{ψ_1}) and (r_φ, r_{ψ_2}). From $\nu^*_{KZ}(\psi_1 \wedge \psi_2) = \min(\nu^*_{KZ}(\psi_1), \nu^*_{KZ}(\psi_2)) = r$ we know that $\nu^*_{KZ}(\psi_1) \geq r$ and $\nu^*_{KZ}(\psi_2) \geq r$. Then, by the induction hypothesis, I have strategies $\langle N|_{\psi_1}, E_{\sigma_{me}}|_{\psi_1}\rangle$ and $\langle N|_{\psi_2}, E_{\sigma_{me}}|_{\psi_2}\rangle$ yielding a payoff of at least r for me[4] in $G_{\psi_1,\nu_{KZ},me}$ and $G_{\psi_2,\nu_{KZ},me}$ respectively.

   My formal strategy is $\sigma_{me} = \langle N, E_{\sigma_{me}}|_{\psi_1} \cup E_{\sigma_{me}}|_{\psi_2} \cup \{(r_\varphi, r_{\psi_1}), (r_\varphi, r_{\psi_2})\}\rangle$.

3. $\varphi = \neg\psi$: at the root $r_{\neg\psi}$ of $G_{\neg\psi,\nu_{KZ},me}$, my only move is playing Neg, leading to the root of $G_{\psi,\nu_{KZ},you}$. By the induction hypothesis you have a strategy $\sigma_{you} = \langle N|_\psi, E_{\sigma_{you}}|_\psi\rangle$ in $G_{\psi,\nu_{KZ},you}$ with value $1 - r$, since $\nu^*_{KZ}(\psi) = 1 - r$. However then my payoff in $G_{\psi,\nu_{KZ},you}$ is $1 - (1 - r) = r$, and with $\sigma_{me} = \langle N, E_{\sigma_{you}} \cup \{(r_\phi, r_\psi)\}\rangle$[5] I have a strategy with payoff r for myself in $G_{\neg\psi,\nu_{KZ},me}$ as required.

We have shown that the initial defender d in $G_{\varphi,\nu_{KZ},d}$ has an r-valued strategy, iff $\nu^*_{KZ}(\varphi) = r$. □

[4] At least one of the subgames' values will be r, and you would play the move leading to it.

[5] Note that, due to Theorem 2.4.1, the two players' strategies mutually define each other

CHAPTER 4

Modal Axiomatization of Evaluation Game Trees

4.1 Capturing Classical $\mathcal{H}$-game Trees with Modal Axioms

The idea behind our formal modeling of $\mathcal{H}$-games is pretty straight-forward: the games are representable by trees and these trees can be seen as Kripke models. The interpretation at each world contains information about the respective game-state. Some game tree labels, like who is active at a state, or whether a terminal state is a win for a player are expressed by a proposition.

Other components like the asserted formula are best modeled by a first-order predicate: the formula becomes a term. We use two predicate symbols ($\text{assert}^1_{\text{me}}$ and $\text{assert}^1_{\text{you}}$) representing the asserted formula and the current asserter.

Arguments to predicates are terms, not formulas. Keeping the levels of meta-language and of the object language strictly separated, while not artificially hiding the intended semantics constitutes an important part of the task at hand. As customary we use different sets of symbols for formulas and terms on the different levels: lower-case Greek letters $\varphi, \psi, \varphi_1, \ldots$ are elements of the domain. Their counterpart on the syntactic level, the terms, are marked by small corners, $\ulcorner\varphi\urcorner, \ulcorner\psi\urcorner, \ldots$. For this chapter we use upper-case Greek letters $\Gamma, \Delta, \Gamma_1, \ldots$ for schematic formula variables, deviating from our use of $\varphi, \ldots$ until now.

4.1.1 Syntax Used

We fix the used *signature* for our modeling.

Definition 4.1.1 (functional signature)**.** The functional signature consists of two binary and one unary function symbols:

- $f^2_\wedge$
- $f^2_\vee$

- $f^1_\neg$

In order to keep the reading more natural we use the following infix notation for the function symbols: $\ulcorner \cdot \wedge \cdot \urcorner$ for $f_\wedge(\cdot,\cdot)$, $\ulcorner \cdot \vee \cdot \urcorner$ for $f_\vee(\cdot,\cdot)$ and $\ulcorner \neg \cdot \urcorner$ for $f_\neg(\cdot)$

The terms represent propositional formulas. The interpretation of $f_\wedge$, $f_\vee$ and $f_\neg$ constructs the conjunction, disjunction or negation of their arguments respectively[1].

Definition 4.1.2 (representabilty of formulas)**.** A term $\ulcorner\varphi\urcorner$ is said to *represent* a formula φ, if the formula has the same syntactic structure as $\ulcorner\varphi\urcorner$: $TV(w, \ulcorner\varphi\urcorner) = \varphi$ for $w \in W$

Definition 4.1.3 (Modalities for $\mathcal{H}$-games)**.** We use three *modalities* in the modeling:

- R
- L
- Neg

Definition 4.1.4 (Atomic formulas for $\mathcal{H}$-games)**.** The following *propositions* and monadic *predicate symbols* (Definition 2.1.6) are used for the modeling:

- $\text{turn}^0_{\text{me}}$
- $\text{turn}^0_{\text{you}}$
- win^0_{me}
- $\text{win}^0_{\text{you}}$
- terminal^0
- $\text{assert}^1_{\text{me}}$
- $\text{assert}^1_{\text{you}}$
- true^1
- false^1

The names chosen hint at the intended meaning, e.g. turn_{me} and turn_{you} reflect the active player at a given world and the terminal proposition is true in leaf worlds.

This particular choice for the used predicate symbols is the result of various experiments with transferring the concepts from $\mathcal{H}$-games to a formal syntactic level. For instance the idea of combining the asserted formula with its defender into $\text{assert}_{\text{me}}$ and $\text{assert}_{\text{you}}$, seems like the best trade-off between literally transferring the $\mathcal{H}$-game tree labels and bloating the set of axioms with redundant clauses.

Definition 4.1.5 (language for $\mathcal{H}$-games L_{HG})**.** We call the language over the functional, modal and predicate signature from Definition 4.1.1, Definition 4.1.3 and Definition 4.1.4 respectively the *language for $\mathcal{H}$-games* L_{HG}.

[1] A concept familiar from classical first-order logic as *Herbrand model* [Lei97]

4.1.2 Modal Axioms

In order to avoid redundancy we use variables i, j for the players me and you with $i \neq j$ in what follows. For example the formula

$$\forall x \forall y(\text{assert}_i(\ulcorner x \wedge y \urcorner) \rightarrow \text{turn}_j)$$

actually stands for the two formulas

$$\forall x \forall y(\text{assert}_{\text{me}}(\ulcorner x \wedge y \urcorner) \rightarrow \text{turn}_{\text{you}})$$
$$\forall x \forall y(\text{assert}_{\text{you}}(\ulcorner x \wedge y \urcorner) \rightarrow \text{turn}_{\text{me}})$$

The axioms will model $\mathcal{H}$-games, and can be broadly categorized into three categories: axioms to describe the decomposition of the asserted formula, using modalities, axioms capturing general rules of games and those needed to represent winning conditions, referring to an atomic valuation.

We list all necessary axioms, shortly describing their purpose informally.

Not all conditions on the game tree are representable by formulas from modal correspondence theory, one condition, that can be ensured is partial functionality for all modalities (see Table 2.2): if there is a world reachable in a particular transition relation, it is the only one in this relation.

(HL-1) $\langle R \rangle \Gamma \rightarrow [R]\Gamma$

(HL-2) $\langle L \rangle \Gamma \rightarrow [L]\Gamma$

(HL-3) $\langle Neg \rangle \Gamma \rightarrow [Neg]\Gamma$

The modalities are not independent of each other: in the case the asserted term represents a conjunction or disjunction, there should be exactly two transitions, one in R_R and one in R_L. For negations we want a R_{Neg} transition to be the only one. We additionally require that there are no transitions from leaf worlds.

(HL-4) $\langle L \rangle \top \leftrightarrow \langle R \rangle \top$

(HL-5) $\langle L \rangle \top \rightarrow [Neg]\bot$

(HL-6) $\langle Neg \rangle \top \rightarrow [L]\bot$

(HL-7) terminal $\rightarrow [R]\bot$

(HL-8) terminal $\rightarrow [L]\bot$

(HL-9) terminal $\rightarrow [Neg]\bot$

(HL-10) $([L]\bot \wedge [R]\bot \wedge [Neg]\bot) \rightarrow$ terminal

Players take turns only at non-terminal histories and only one player takes a turn at each state.

(HL-11) $\neg\text{terminal} \leftrightarrow (\text{turn}_i \leftrightarrow \neg\text{turn}_j)$

(HL-12) $\text{terminal} \rightarrow \neg\text{turn}_i$

In leafs only atomic formulas are asserted, thus the term representing the asserted formula, cannot have a function symbol at its root position.

(HL-13) $\forall x \forall y(\text{terminal} \rightarrow \neg\text{assert}_i(\ulcorner x \wedge y \urcorner))$

(HL-14) $\forall x \forall y(\text{terminal} \rightarrow \neg\text{assert}_i(\ulcorner x \vee y \urcorner))$

(HL-15) $\forall x(\text{terminal} \rightarrow \neg\text{assert}_i(\ulcorner \neg x \urcorner))$

The first three axioms below express that the outermost connective determines, which player is to move at a given game-state. The remaining axioms ensure that a left or right choice of a player indeed yields the left or right subformula, and that a negation really results in a role-change of the two players, with the new formula being the formula without the outermost negation.

(HL-16) $\forall x \forall y(\text{assert}_i(\ulcorner x \vee y \urcorner) \rightarrow \text{turn}_i)$

(HL-17) $\forall x \forall y(\text{assert}_i(\ulcorner x \wedge y \urcorner) \rightarrow \text{turn}_j)$

(HL-18) $\forall x(\text{assert}_i(\ulcorner \neg x \urcorner) \rightarrow \text{turn}_i)$

(HL-19) $\forall x \forall y(\text{assert}_i(\ulcorner x \vee y \urcorner) \rightarrow \langle L \rangle \text{assert}_i(x))$

(HL-20) $\forall x \forall y(\text{assert}_i(\ulcorner x \vee y \urcorner) \rightarrow \langle R \rangle \text{assert}_i(y))$

(HL-21) $\forall x \forall y(\text{assert}_i(\ulcorner x \wedge y \urcorner) \rightarrow \langle L \rangle \text{assert}_i(x))$

(HL-22) $\forall x \forall y(\text{assert}_i(\ulcorner x \wedge y \urcorner) \rightarrow \langle R \rangle \text{assert}_i(y))$

(HL-23) $\forall x(\text{assert}_i(\ulcorner \neg x \urcorner) \rightarrow \langle Neg \rangle \text{assert}_j(x))$

Conversely an existing transition also tells us what the outermost connective of the formula asserted possibly is. This is needed to show the direction that an $\mathcal{H}$-game model corresponds to an $\mathcal{H}$-game:

(HL-24) $\exists x \exists y(\langle L \rangle \top \rightarrow (\text{assert}_i(\ulcorner x \wedge y \urcorner) \vee \text{assert}_i(\ulcorner x \vee y \urcorner) \vee \text{assert}_j(\ulcorner x \wedge y \urcorner) \vee \text{assert}_j(\ulcorner x \vee y \urcorner)))$

(HL-25) $\exists x(\langle Neg \rangle \top \rightarrow \text{assert}_i(\ulcorner \neg x \urcorner) \vee \text{assert}_j(\ulcorner \neg x \urcorner))$

Exactly one formula is under consideration at any given world, and only one player is defending it.

(HL-26) $\forall x \forall y \neg(\text{assert}_i(x) \wedge \text{assert}_j(y))$

(HL-27) $\forall x \forall y(\text{assert}_i(x) \wedge \text{assert}_i(y) \rightarrow x = y)$

(HL-28) $\exists x(\text{assert}_i(x) \vee \text{assert}_j(x))$

The defender wins in a terminal state, iff the asserted atom is true, and loses iff it is false. Furthermore if the defender wins the attacker loses.

(HL-29) $\text{win}_i \rightarrow \text{terminal}$

(HL-30) $\forall x(\text{terminal} \rightarrow ((\text{assert}_i(x) \wedge \text{true}(x)) \rightarrow \text{win}_i))$

(HL-31) $\forall x(\text{terminal} \rightarrow ((\text{assert}_i(x) \wedge \text{false}(x)) \rightarrow \text{win}_j))$

Definition 4.1.6 ($\mathcal{H}$-game axioms)**.** The set consisting of axioms (HL-1) – (HL-31) is called *$\mathcal{H}$-game axioms*.

4.1.3 $\mathcal{H}$-game Models

The modeling is achieved through a multi-modal logic with three modalities.

Since Kripke models cannot be restricted to the needed tree-like structure, solely by adding corresponding axioms, characterization of some of those properties remains on the semantic side.

Definition 4.1.7 ($\mathcal{H}$-game model)**.** A *Kripke model* $K = \langle W, w_r, \{R_R, R_L, R_{Neg}\}, D, V\rangle$ over the language for $\mathcal{H}$-games, is called a *$\mathcal{H}$-game model*, if:

- the graph $\langle W, R_R \cup R_L \cup R_{Neg}\rangle$ is a *finite rooted tree*, with w_r as its root. We call it the *underlying tree of* K, T_K.
- $D = \mathcal{P}rop$, the set of propositional formulas
- V assigns for $\varphi, \psi \in D$ uniformly in all $w \in W$ to $f_\wedge, f_\vee$ and $f_\neg$:
  - $V(w, f_\wedge)(\varphi, \psi) = \varphi \wedge \psi$
  - $V(w, f_\vee)(\varphi, \psi) = \varphi \vee \psi$
  - $V(w, f_\neg)(\varphi) = \neg\varphi$
- The interpretation of the predicate symbols and propositions is restricted by the axioms and dependent on a concrete game represented by the model.

Let I be a first-order variable valuation, ν_{CL} a classical valuation and $TV(w, \ulcorner\varphi\urcorner) = \varphi$. We say that K *reflects* ν_{CL}, if:

- $K, I, w \models \text{true}(\ulcorner\varphi\urcorner) \Leftrightarrow \nu_{CL}(\varphi) = 1$
- $K, I, w \models \text{false}(\ulcorner\varphi\urcorner) \Leftrightarrow \nu_{CL}(\varphi) = 0$

hold for all $w \in W$.

Furthermore we say that K is *for* φ *with initial defender* $d \in \{you, me\}$, if:

$$K, I, w_r \models \text{assert}_d(\ulcorner\varphi\urcorner)$$

where $TV(w, \ulcorner\varphi\urcorner) = \varphi$.

We call an $\mathcal{H}$-game model *adequate* if it makes the $\mathcal{H}$-game axioms true.

4.2 Adequateness of the Modeling

Towards our goal of finding a suitable set of axioms that capture $\mathcal{H}$-games, we have defined a suitable candidate in Definition 4.1.6, and restricted the admissible Kripke models on the semantic side where necessary in Definition 4.1.7. In the remainder of this section we show that our axioms restrict the possible models to those which represent valid $\mathcal{H}$-games (see Definition 3.2.1), and that every $\mathcal{H}$-game has a corresponding model.

This entails that any theorem derived from the $\mathcal{H}$-game axioms, is actually a valid statement about $\mathcal{H}$-games.

It is instructive to observe that our definition of an $\mathcal{H}$-game model (see (Definition 4.1.7) permits many Kripke models which would not represent valid $\mathcal{H}$-games, however those $\mathcal{H}$-game models which are models of the $\mathcal{H}$-game axioms, all represent valid $\mathcal{H}$-games.

Talking almost interchangeably about $\mathcal{H}$-game trees on the one hand and the semantic structures introduced as $\mathcal{H}$-game models on the other hand needs a formal definition:

Definition 4.2.1 ($\mathcal{H}$-game tree-model mapping)**.** Let $G_{\varphi,\nu_{CL},d} = \langle N, E \rangle$ be an $\mathcal{H}$-game tree on φ, over ν_{CL} (Definition 3.2.1).

We construct an $\mathcal{H}$-game model $K = K(G) = \langle W, w_r, \{R_R, R_L, R_{Neg}\}, D, V \rangle$ for φ with initial defender d reflecting a valuation ν_{CL} (Definition 4.1.7), based on G through the following mapping:

$T_K = \langle N, E \rangle$. The predicate interpretation is defined by the node labels of G, and the three accessibility relations are disjunct subsets of E based on the edge labels, as follows:

- $l_{move}((u,v)) = R$, iff $(u,v) \in R_R$
- $l_{move}((u,v)) = L$, iff $(u,v) \in R_L$
- $l_{move}((u,v)) = Neg$, iff $(u,v) \in R_{Neg}$
- $l_{turn}(u) = me$, iff $u \in V(\text{turn}_{\text{me}})$
- $l_{turn}(u) = you$, iff $u \in V(\text{turn}_{\text{you}})$
- $l_{win_{me}}(u) = win$, iff $u \in V(\text{win}_{\text{me}})$
- $l_{win_{you}}(u) = lose$, iff $u \in V(\text{win}_{\text{me}})$
- $l_{win_{me}}(u) = lose$, iff $u \in V(\text{win}_{\text{you}})$
- $l_{win_{you}}(u) = win$, iff $u \in V(\text{win}_{\text{you}})$
- $l_{assert}(u) = me$, iff $(u,\varphi) \in V(\text{assert}_{\text{me}})$, for any φ
- $l_{form}(u) = \varphi$, iff $(u,\varphi) \in V(\text{assert}_{\text{me}})$, or $(u,\varphi) \in V(\text{assert}_{\text{you}})$
- $u \in N_0$, iff $u \in V(\text{terminal})$

We say that G is *interpreted as* a model K, or vice versa, when one results from the other according to this mapping.

4.2.1 Every Adequate $\mathcal{H}$-game Model is an $\mathcal{H}$-game

We show that any adequate $\mathcal{H}$-game model, when interpreted as an $\mathcal{H}$-game, is valid according to Definition 3.2.1.

The fact that only one formula is asserted by only one player at each world of an adequate $\mathcal{H}$-game model is needed more than once in the proof of Theorem 4.2.4, therefore we show it separately:

Lemma 4.2.1. For every world $w \in W$ of an adequate $\mathcal{H}$-game model K there is exactly one formula φ for which one of $\text{assert}_{\text{me}}$ or $\text{assert}_{\text{you}}$ holds in w.

Proof. Since (HL-28) holds, we have $K, I, w \models \exists x(\text{assert}_{\text{me}}(x) \vee \text{assert}_{\text{you}}(x))$, there is a φ, s.t. at least one of the following two hold:

$$K, I[x \mapsto \varphi], w \models \text{assert}_{\text{me}}(x)$$
$$K, I[x \mapsto \varphi], w \models \text{assert}_{\text{you}}(x)$$

On the other hand we have (HL-26) $K, I, w \models \forall x \forall y \neg(\text{assert}_{\text{me}}(x) \wedge \text{assert}_{\text{you}}(y))$, thus for all pairs φ, ψ:

$$K, I[x \mapsto \varphi][y \mapsto \psi], w \not\models \text{assert}_{\text{me}}(x) \wedge \text{assert}_{\text{you}}(y)$$

We conclude that there is no pair of formulas, φ, ψ, s.t. both $V(w, \text{assert}_{\text{me}})(\varphi) = 1$ and $V(w, \text{assert}_{\text{you}})(\psi) = 1$.

As a last step we need to verify that not more than one formula is asserted by the current asserter: for any $w \in W$, we have at most one φ, s.t. $V(w, \text{assert}_{\text{me}})(\varphi) = 1$, the argument for $\text{assert}_{\text{you}}$ is identical:

From (HL-27) $K, I, w \models \forall x \forall y(\text{assert}_{\text{me}}(x) \wedge \text{assert}_{\text{me}}(y) \to x = y)$, we get that

$$K, I[x \mapsto \varphi][y \mapsto \psi], w \models \text{assert}_{\text{me}}(x) \wedge \text{assert}_{\text{me}}(y) \to x = y$$

for all φ, ψ. Therefore whenever we have $K, I[x \mapsto \varphi][y \mapsto \psi], w \models \text{assert}_{\text{me}}(x) \wedge \text{assert}_{\text{me}}(y)$, we also have that $\varphi = \psi$, concluding the proof. □

Some of the $\mathcal{H}$-game axioms impose certain restrictions on the form of the tree *underlying* a given $\mathcal{H}$-game model, T_K. We defined the frame of an $\mathcal{H}$-game model as a rooted finite tree. every T_K, where K is *adequate* already has the form of an $\mathcal{H}$-game.

This restricts the possible forms of T_K and our argument that every adequate $\mathcal{H}$-game model can be interpreted as an $\mathcal{H}$-game relies on that fact.

The following two lemmas provide the formal argument for this:

Lemma 4.2.2. The *underlying tree* T_K of an adequate $\mathcal{H}$-game model K has a *width* (see Definition 2.4.4) of at most 2.

Proof. Due to the partial-functionality axioms — $\langle R \rangle \Gamma \to [R]\Gamma$, $\langle L \rangle \Gamma \to [L]\Gamma$ and $\langle Neg \rangle \Gamma \to [Neg]\Gamma$ (and due to those 3 being the only modalities), we know that from any world $w \in W$ there are at most 3 reachable worlds $v_1, v_2, v_3 \in W$:
Assume $(w, v_1) \in R_R$, $(w, v_2) \in R_L$ and $(w, v_3) \in R_{Neg}$.

To show that only 2 worlds can be reachable from w, we consider that $K, I, w \models \langle L \rangle \top \rightarrow [Neg]\bot$ (HL-5): Assuming that v_2 exists, we have $K, I, w \models \langle L \rangle \top$, therefore via the definition of implication:

$$K, I, w \models [Neg]\bot$$

However then v_3 cannot exist, since we would have $K, I, v_3 \models \bot$, which is not possible. We conclude that v_2 and v_3 cannot both be reachable from w, i.e. if there exists an L transition, there cannot be a Neg transition and therefore any world w has at most two transitions to other worlds. □

This section deals with the semantic evaluation of the $\mathcal{H}$-game axioms with respect to $\mathcal{H}$-game models. As can be seen in the proof of Lemma 4.2.1, this mainly consists of collecting various assumptions we make, either because K satisfies the $\mathcal{H}$-game axioms, or because of some local property of the proof. These assumptions are in general of the form $K, I, w \models \Gamma$ or $K, I, w \models \Gamma \rightarrow \Delta$ and are manipulated according to the definition of the semantic satisfiability relation. For example concluding from the two assumptions above that $K, I, w \models \Delta$.

To keep the proofs free from redundant clutter of the form "From x we conclude that y", we adopt a more terse representation of the semantic arguments: an Arabic numeral for later reference, as well as justifications of the rules are written on the left, the semantic satisfiability relation, or the final conclusion itself in the middle, and the right-hand side either contains the symbol $\Rightarrow$, or the word *or*, meaning implication or disjunction on the meta-level respectively. $\Rightarrow$ is possibly subscripted with reference labels, referring to the line numbers of the proof. If $\Rightarrow$ has no subscript, the following line is a direct consequence. The justifications are abbreviated, e.g. $\models_{\rightarrow}$ means: by the definition of the satisfiability relation for implication. We write $\models_{\mathcal{Atom}}$, for the truth condition of atomic first-order formulas. The above argument, that L and Neg transitions cannot coincide, looks as follows in this presentation:

1 $(w, v_2) \in R_L$	$K, I, w \models \langle L \rangle \top$	
2 (HL-5)	$K, I, w \models \langle L \rangle \top \rightarrow [Neg]\bot$	$\Rightarrow_{1,2}$
3 $\models_{\rightarrow}$	$K, I, w \models [Neg]\bot$	$\Rightarrow$
4 $(w, v_3) \in R_{Neg}, \models_{\langle Neg \rangle}$	$K, I, v_3 \models \bot$	$\Rightarrow$
contradiction to $\models_{\bot}$	v_2, v_3 cannot exist simultaneously.	

Lemma 4.2.3. The *underlying tree* T_K of an adequate $\mathcal{H}$-game model K is of one of the forms illustrated in Figure 4.1[2].

Proof. From Lemma 4.2.2 we know that the width of the tree is at most two and have concluded that no Neg transition can occur at the same time as an L transition.
It remains to show that:

1. a single world can be a model of the $\mathcal{H}$-game axioms

[2] Technically it is possible for two worlds $u, v \in W$ to be in R_L and R_R at the same time: this may happen when we have $K, I, u \models \text{assert}_i(\ulcorner \varphi \wedge \varphi \urcorner)$ or $K, I, u \models \text{assert}_i(\ulcorner \varphi \vee \varphi \urcorner)$. However then K would still correspond to the $\mathcal{H}$-game.

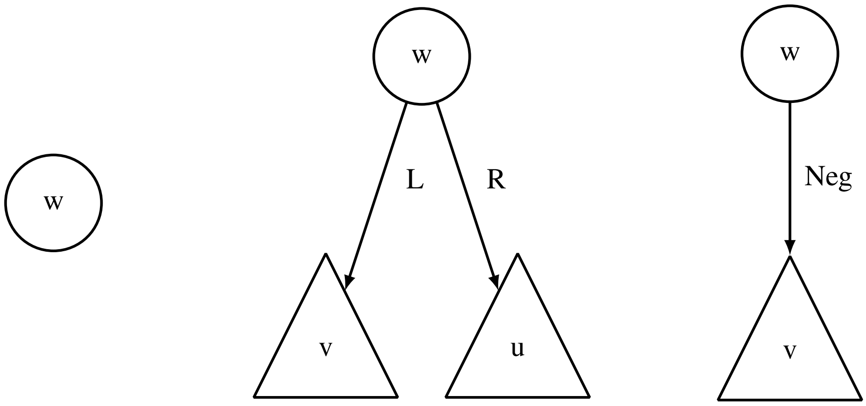


Figure 4.1: possible forms of $\mathcal{H}$-game models — the triangles represent arbitrary subtrees of one of those forms

2. R and L transitions cannot occur independently
3. a Neg transition can only occur by itself

For 1 we only need to consider those $\mathcal{H}$-game axioms containing a modal operator. In addition we can safely ignore axioms, where the modal subformulas are the premise of an implication and not of the form $[m]\bot$, because the premise would evaluate to false and thus make the axiom true. Let w be the single world of K. Since T_K is a tree by assumption we have $R_R = R_L = R_{Neg} = \emptyset$:

1 $R_R = R_L = R_{Neg} = \emptyset$	$K, I, w \models [L]\bot \land [R]\bot \land [Neg]\bot$	
2 (HL-10)	$K, I, w \models ([L]\bot \land [R]\bot \land [Neg]\bot) \rightarrow$ terminal	$\overset{\Rightarrow}{_{1,2}}$
3 $\models_{\rightarrow}$	$K, I, w \models$ terminal	$\Rightarrow$
4 $\models_{\mathcal{Atom}}$	$V(w, \text{terminal}) = 1$	

The three axioms (HL-7) – (HL-9) are true in K because we have $K, I, w \models$ terminal. We show this for (HL-7):

1 $V(w, \text{terminal}) = 1$	$K, I, w \models$ terminal	
2 $R_R = \emptyset$	$K, I, w \models [R]\bot$	$\overset{\Rightarrow}{_{1,2}}$
3 $\models_{\rightarrow}$, (HL-7)	$K, I, w \models$ terminal $\rightarrow [R]\bot$	

The only axioms, which we still have to consider are (HL-19) – (HL-23). The argument is similar for all of them, therefore we only deal with (HL-19) and only with $i = me$ explicitly:

1	$V(w, \text{terminal}) = 1$	$K, I, w \models \text{terminal}$	
2	(HL-14)	$K, I, w \models \forall x \forall y(\text{terminal} \to \neg\text{assert}_{\text{me}}(\ulcorner x \vee y \urcorner))$	$\Rightarrow$
3	$\models_\forall$, for all $\psi_1, \psi_2 \in D$	$K, I_{[x \mapsto \psi_1][y \mapsto \psi_2]}, w \models \text{terminal} \to \neg\text{assert}_{\text{me}}(\ulcorner x \vee y \urcorner)$	$\overset{\Rightarrow}{_{1,3}}$
4	$\models_\to$	$K, I_{[x \mapsto \psi_1][y \mapsto \psi_2]}, w \models \neg\text{assert}_{\text{me}}(\ulcorner x \vee y \urcorner)$	$\Rightarrow$
5	$\models_\neg$	$K, I_{[x \mapsto \psi_1][y \mapsto \psi_2]}, w \not\models \text{assert}_{\text{me}}(\ulcorner x \vee y \urcorner)$	$\Rightarrow$
6	$\models_\to$ the premise is false	$K, I_{[x \mapsto \psi_1][y \mapsto \psi_2]}, w \models \text{assert}_{\text{me}}(\ulcorner x \vee y \urcorner) \to \langle L \rangle \text{assert}_{\text{me}}(x)$	$\overset{\Rightarrow}{_{3,6}}$
7	$\models_\forall$, (HL-19)	$K, I, w \models \forall x \forall y(\text{assert}_{\text{me}}(\ulcorner x \vee y \urcorner) \to \langle L \rangle \text{assert}_{\text{me}}(x))$	

The argument for point 2 is straightforward: assume that a world w has an R-accessible world v:

1	$(w, v) \in R_R$	$K, I, w \models \langle R \rangle \top$	
2	(HL-4)	$K, I, w \models \langle L \rangle \top \leftrightarrow \langle R \rangle \top$	$\Rightarrow$
3	$\models_\leftrightarrow$	$K, I, w \models \langle R \rangle \top \to \langle L \rangle \top$	$\overset{\Rightarrow}{_{1,3}}$
4	$\models_\to$	$K, I, w \models \langle L \rangle \top$	$\Rightarrow$
5	$\models_{\langle L \rangle}$, there has to be a u, s.t. $(w, u) \in R_L$	$K, I, u \models \top$	

We conclude that whenever there is an R transition, there also is an L transition.

For point 3 we just need to observe that R and L transitions cannot occur independently and that it is impossible for an L transition to exist simultaneously with a Neg transition as shown in Lemma 4.2.2. □

Theorem 4.2.4. Every adequate $\mathcal{H}$-game model K for a formula φ with initial defender d reflecting a valuation ν_{CL} corresponds to an $\mathcal{H}$-game for φ over ν_{CL} $G_{\varphi,\nu_{CL},d}$.

Proof. The argument proceeds by induction on the height of T_K. Through Lemma 4.2.3 we already know that T_K has indeed the form of an $\mathcal{H}$-game tree.

It remains to show that the predicates and modalities of K, *interpreted as* labels of T_K (see Definition 4.2.1) fulfill the requirements imposed on $G_{\varphi,\nu_{CL},d}$.

Assume $d = me$. The argument for $d = you$ is symmetric.

The base case is a single world $w = w_r$, corresponding to an $\mathcal{H}$-game for an atomic proposition $G_{a,\nu_{CL},d}$. T_K is $\langle \{w\}, \emptyset \rangle$, thus we have $R_R = R_L = R_{Neg} = \emptyset$.

We have to show that:

a) The single[3] formula φ having $V(w, \text{assert}_{\text{me}})(\varphi) = 1$ is atomic; we have $l_{assert}(w) = me$ and $l_{form}(w) = \varphi$ for $\varphi \in \mathcal{Atom}$.

b) If $\nu_{CL}(\varphi) = 1$, then $V(w, \text{win}_{\text{me}}) = 1$ resulting in $l_{win_{me}}(w) = win$ and $l_{win_{you}}(w) = lose$.

[3] By Lemma 4.2.1 for each $w \in W$ either $V(w, \text{assert}_{\text{me}})$ or $V(w, \text{assert}_{\text{you}})$ is true for exactly one $\varphi \in \mathcal{Prop}$

c) Conversely if $\nu_{CL}(\varphi) = 0$ we have $V(w, \text{win}_{\text{you}}) = 1$ and therefore $l_{win_{me}}(w) = lose$ and $l_{win_{you}}(w) = win$.

d) There is no turn label for w, i.e. $V(w, \text{turn}_{\text{me}}) = V(w, \text{turn}_{\text{you}}) = 0$.

a) $V(w, \text{assert}_{\text{me}})(\varphi) = 1$, for exactly one $\varphi \in \mathcal{Atom}$. Taking into account Lemma 4.2.1, it remains to show that $\varphi \in \mathcal{Atom}$. As already shown for Lemma 4.2.3, we have $V(w, \text{terminal}) = 1$, if the $\mathcal{H}$-game model contains only w:

1 K is for φ, $TV(\ulcorner\varphi\urcorner) = \varphi$	$K, I, w \models \text{assert}_{\text{me}}(\ulcorner\varphi\urcorner)$	
2 (HL-13)	$K, I, w \models \forall x \forall y(\text{terminal} \rightarrow \neg\text{assert}_{\text{me}}(\ulcorner x \wedge y\urcorner))$	$\Rightarrow$
3 $\models_\forall$ for all pairs $\psi_1, \psi_2 \in \mathcal{Prop}$	$K, I[x \mapsto \psi_1][y \mapsto \psi_2], w \models \text{terminal} \rightarrow \neg\text{assert}_{\text{me}}(\ulcorner x \wedge y\urcorner)$	$\Rightarrow$
4 $V(w, \text{terminal}) = 1, \models_\rightarrow, \models_\neg$	$K, I[x \mapsto \psi_1][y \mapsto \psi_2], w \not\models \text{assert}_{\text{me}}(\ulcorner x \wedge y\urcorner)$	$\Rightarrow$
5 $V(w, f_\wedge) = \wedge$, for all ψ_1, ψ_2	$V(w, \text{assert}_{\text{me}})(\psi_1 \wedge \psi_2) = 0$	$\overset{\Rightarrow}{_{1,5}}$
6 for any $\psi_1, \psi_2 \in \mathcal{Prop}$	$\varphi \neq \psi_1 \wedge \psi_2$	

$V(w, \text{assert}_{\text{me}})$ is false for the conjunction of any two propositional formulas. The reasoning for disjunction and negation is similar using and the definition of $V(w, f_\vee) = \vee$[4] and $V(w, f_\neg) = \neg$ and axioms (HL-14) and (HL-15).

Since φ can neither be a conjunction nor a disjunction, nor a negation φ has to be atomic.

b) From $\nu_{CL}(\varphi) = 1$ we get $V(w, \text{win}_{\text{me}}) = 1$, thus $l_{win_{me}} = win$ and $l_{win_{you}} = lose$

1 $V(w, \text{terminal}) = 1$	$K, I, w \models \text{terminal}$	
2 K reflects $\nu_{CL}, TV(\ulcorner\varphi\urcorner) = \varphi$	$K, I, w \models \text{true}(\ulcorner\varphi\urcorner)$	
3 K is for φ	$K, I, w \models \text{assert}_{\text{me}}(\ulcorner\varphi\urcorner)$	
4 (HL-30)	$K, I, w \models \forall x(\text{terminal} \rightarrow ((\text{assert}_{\text{me}}(x) \wedge \text{true}(x)) \rightarrow \text{win}_{\text{me}}))$	$\Rightarrow$
5 $\models_\forall$, for all $\psi \in \mathcal{Prop}$	$K, I[x \mapsto \psi], w \models \text{terminal} \rightarrow (\text{assert}_{\text{me}}(x) \wedge \text{true}(x)) \rightarrow \text{win}_{\text{me}}))$	$\overset{\Rightarrow}{_{1,5}}$
6 particularly for $\psi = \varphi, \models_\rightarrow$	$K, I[x \mapsto \varphi], w \models \text{assert}_{\text{me}}(x) \wedge \text{true}(x) \rightarrow \text{win}_{\text{me}}$	$\overset{\Rightarrow}{_{2,3,6}}$
7 $\models_\wedge, \models_\rightarrow$	$K, I[x \mapsto \varphi], w \models \text{win}_{\text{me}}$	$\overset{\Rightarrow}{_{2,7}}$
	$V(w, \text{win}_{\text{me}}) = 1$	

[4]Shorthand for $V(w, f_\vee)(\varphi, \psi) = \varphi \vee \psi$

c) From $\nu_{CL}(\varphi) = 0$ we get $V(w, \text{win}_{\text{you}}) = 1$, thus $l_{win_{you}} = win$ and $l_{win_{me}} = lose$

1 $V(w, \text{terminal}) = 1$	$K, I, w \models \text{terminal}$	
2 K reflects $\nu_{CL}, TV(\ulcorner\varphi\urcorner) = \varphi$	$K, I, w \models \text{false}(\ulcorner\varphi\urcorner)$	
3 K is for φ	$K, I, w \models \text{assert}_{\text{me}}(\ulcorner\varphi\urcorner)$	
4 (HL-31)	$K, I, w \models \forall x(\text{terminal} \rightarrow ((\text{assert}_{\text{me}}(x) \wedge \text{false}(x)) \rightarrow \text{win}_{\text{you}}))$	$\Rightarrow$
5 $\models_\forall$, for all $\psi \in \mathcal{Prop}$	$K, I[x \mapsto \psi], w \models \text{terminal} \rightarrow (\text{assert}_{\text{me}}(x) \wedge \text{false}(x)) \rightarrow \text{win}_{\text{you}}))$	$\overset{\Rightarrow}{1,5}$
6 particularly for $\psi = \varphi, \models_\rightarrow$	$K, I[x \mapsto \varphi], w \models \text{assert}_{\text{me}}(x) \wedge \text{false}(x) \rightarrow \text{win}_{\text{you}}$	$\overset{\Rightarrow}{2,3,6}$
7 $\models_\wedge, \models_\rightarrow$	$K, I[x \mapsto \varphi], w \models \text{win}_{\text{you}}$	$\overset{\Rightarrow}{2,7}$
	$V(w, \text{win}_{\text{you}}) = 1$	

d)

- $V(w, \text{turn}_{\text{me}}) = 0$:

1 (HL-12)	$K, I, w \models \text{terminal} \rightarrow \neg\text{turn}_{\text{me}}$	
2 $V(w, \text{terminal}) = 1$	$K, I, w \models \text{terminal}$	$\overset{\Rightarrow}{1,2}$
3 $\models_\rightarrow, \models_\neg$	$K, I, w \not\models \text{turn}_{\text{me}}$	$\Rightarrow$
4 $\models_{\mathcal{Atom}}$	$V(w, \text{turn}_{\text{me}}) = 0$	

- $V(w, \text{turn}_{\text{you}}) = 0$ is identical to above, with $K, I, w \models \text{terminal} \rightarrow \neg\text{turn}_{\text{you}}$.

This concludes the base case.

For the inductive step assume that every $\mathcal{H}$-game model $K(G_{\varphi,\nu_{CL},d})$, with an underlying tree of depth $\leq k$ corresponds to the $\mathcal{H}$-game $G_{\varphi,\nu_{CL},d}$. For depth $k+1$ we can restrict attention to those models, where T_K has one of the forms seen in Figure 4.1[5]. We have two main cases:

1. w_r has a single Neg transition to a world v. This entails that $K, I, w_r \models \langle Neg \rangle \top$.
   We have to show that:
   1.a) $V(w_r, \text{win}_{\text{me}}) = V(w_r, \text{win}_{\text{you}}) = 0$: w_r has no win label.
   1.b) $V(w_r, \text{assert}_{\text{me}})(\neg\psi) = 1$ and $V(v, \text{assert}_{\text{you}})(\psi) = 1$. For $\psi \in \mathcal{Prop}$, being a formula that, by the induction hypothesis, has a corresponding $\mathcal{H}$-game model $K(G_{\psi,\nu_{CL},you})$, with root v and T_K of depth k, where the node and edge labels are as needed.
   1.c) $V(w_r, \text{turn}_{\text{me}}) = 1$, and therefore $l_{turn} = me$.
   1.d) $V(w_r, \text{turn}_{\text{you}}) = 0$[6].

[5]By Lemma 4.2.3

[6]The step from valuations of atomic predicates to game tree labels is a direct consequence of Definition 4.2.1, i.e. $V(w_r, \text{turn}_{\text{you}})$ implies $l_{turn} = you$

2. w_r has two reachable worlds u and v, with $(w_r, u) \in R_L$ and $(w_r, v) \in R_R$. We have $K, I, w_r \models \langle L \rangle \top$ and $K, I, w_r \models \langle R \rangle \top$. We have to show that:

   2.a) $V(w_r, \text{win}_{\text{me}}) = V(w_r, \text{win}_{\text{you}}) = 0$

   2.b) $V(w_r, \text{assert}_{\text{me}})(\psi_1 \vee \psi_2) = 1$, or $V(w_r, \text{assert}_{\text{me}})(\psi_1 \wedge \psi_2) = 1$ for $\psi_1, \psi_2 \in \mathcal{Prop}$. By the induction hypothesis we have $\mathcal{H}$-game models $K_1(G_{\psi_1, \nu_{CL}, me})$ and $K_2(G_{\psi_2, \nu_{CL}, me})$ with roots u and v respectively. Both T_{K_1} and T_{K_2} have a depth $\leq k$. In both cases, we have $V(u, \text{assert}_{\text{me}})(\psi_1) = 1$ and $V(v, \text{assert}_{\text{me}})(\psi_2) = 1$.

   2.b.i) If $V(w_r, \text{assert}_{\text{me}})(\psi_1 \vee \psi_2) = 1$, then $V(w_r, \text{turn}_{\text{me}}) = 1$ and $V(w_r, \text{turn}_{\text{you}}) = 0$.

   2.b.ii) If $V(w_r, \text{assert}_{\text{me}})(\psi_1 \wedge \psi_2) = 1$, then $V(w_r, \text{turn}_{\text{you}}) = 1$ and $V(w_r, \text{turn}_{\text{me}}) = 0$.

In both cases 1 and 2 we have $V(w_r, \text{terminal}) = 0$. For the negation case we show this:

1 $(w_r, v) \in R_{Neg}$	$K, I, w_r \models \langle Neg \rangle \top$	
2 (HL-3)	$K, I, w_r \models \langle Neg \rangle \top \to [Neg]\top$	$\overset{\Rightarrow}{_{1,2}}$
3 $\models_{\to}$	$K, I, w_r \models [Neg]\top$	$\Rightarrow$
4 $\models_{[Neg]}$	$K, I, v \models \top$	$\Rightarrow$
5 $\models_{\top}, \models_{\bot}, \models_{[Neg]}$	$K, I, w_r \not\models [Neg]\bot$	$\Rightarrow$
6 (HL-9)	$K, I, w_r \models \text{terminal} \to [Neg]\bot$	$\overset{\Rightarrow}{_{5,6}}$
7 $\models_{\to}$	$K, I, w_r \not\models \text{terminal}$	$\Rightarrow$
8 $\models_{\mathcal{Atom}}$	$V(w_r, \text{terminal}) = 0$	

The argument for 2 is almost identical using (HL-2) and (HL-8) instead of (HL-3) and (HL-9).
1.a) $V(w_r, \text{win}_{\text{me}}) = V(w, \text{win}_{\text{you}}) = 0$. We show explicitly that $V(w_r, \text{win}_{\text{me}}) = 0$:

1 (HL-29)	$K, I, w_r \models \text{win}_{\text{me}} \to \text{terminal}$	
2 $V(w_r, \text{terminal}) = 0$	$K, I, w_r \not\models \text{terminal}$	$\overset{\Rightarrow}{_{1,2}}$
3 $\models_{\to}$	$K, I, w_r \not\models \text{win}_{\text{me}}$	$\Rightarrow$
4 $\models_{\mathcal{Atom}}$	$V(w_r, \text{win}_{\text{me}}) = 0$	

The argument for $V(w, \text{win}_{\text{you}}) = 0$ is identical.
1.b) $V(w_r, \text{assert}_{\text{me}})(\neg\psi) = 1$:

1 $(w_r, v) \in R_{Neg}$	$K, I, w_r \models \langle Neg \rangle \top$	
2 (HL-25)	$K, I, w_r \models \exists x(\langle Neg \rangle \top \to \text{assert}_{\text{me}}(\ulcorner \neg x \urcorner) \vee \text{assert}_{\text{you}}(\ulcorner \neg x \urcorner))$	$\Rightarrow$
3 there is a ψ	$K, I[x \mapsto \psi], w_r \models \langle Neg \rangle \top \to \text{assert}_{\text{me}}(\ulcorner \neg x \urcorner) \vee \text{assert}_{\text{you}}(\ulcorner \neg x \urcorner)$	$\overset{\Rightarrow}{_{1,3}}$
4 $\models_{\to}$	$K, I[x \mapsto \psi], w_r \models \text{assert}_{\text{me}}(\ulcorner \neg x \urcorner) \vee \text{assert}_{\text{you}}(\ulcorner \neg x \urcorner)$	$\Rightarrow$
5 Lemma 4.2.1, $d = me$	$K, I[x \mapsto \psi], w_r \models \text{assert}_{\text{me}}(\ulcorner \neg x \urcorner)$	
6 $V(w_r, f_{\neg}) = \neg, \models_{\mathcal{Atom}}$	$V(w_r, \text{assert}_{\text{me}})(\neg\psi) = 1$	

$V(v, \text{assert}_{\text{you}})(\psi) = 1$:

1 from above $\quad K, I_{[x \mapsto \psi]}, w_r \models \text{assert}_{\text{me}}(\ulcorner \neg x \urcorner)$

2 (HL-23) $\quad K, I, w_r \models \forall x(\text{assert}_{\text{me}}(\ulcorner \neg x \urcorner) \rightarrow \langle Neg \rangle \text{assert}_{\text{you}}(x)) \quad \Rightarrow$

3 $\models_\forall$, especially for ψ $\quad K, I_{[x \mapsto \psi]}, w_r \models \text{assert}_{\text{me}}(\ulcorner \neg x \urcorner) \rightarrow \langle Neg \rangle \text{assert}_{\text{you}}(x) \quad \underset{1,3}{\Rightarrow}$

4 $\models_\rightarrow$ $\quad K, I_{[x \mapsto \psi]}, w_r \models \langle Neg \rangle \text{assert}_{\text{you}}(x) \quad \Rightarrow$

5 $\models_{\langle Neg \rangle}, (w_r, v) \in R_{Neg}$ $\quad K, I_{[x \mapsto \psi]}, v \models \text{assert}_{\text{you}}(x) \quad \Rightarrow$

6 $\models_{\mathcal{A}tom}$ $\quad V(v, \text{assert}_{\text{you}})(\psi) = 1$

1.c) $V(w_r, \text{turn}_{\text{me}}) = 1$:

1 (HL-18) $\quad K, I, w_r \models \forall x(\text{assert}_{\text{me}}(\ulcorner \neg x \urcorner) \rightarrow \text{turn}_{\text{me}}) \quad \Rightarrow$

2 $\models_\forall$, particularly for ψ $\quad K, I[x \mapsto \psi], w_r \models \text{assert}_{\text{me}}(\ulcorner \neg x \urcorner) \rightarrow \text{turn}_{\text{me}}$

3 from above $\quad K, I[x \mapsto \psi], w_r \models \text{assert}_{\text{me}}(\ulcorner \neg x \urcorner) \quad \underset{2,3}{\Rightarrow}$

4 $\models_\rightarrow$ $\quad K, I[x \mapsto \psi], w_r \models \text{turn}_{\text{me}}$

5 $\models_{\mathcal{A}tom}$ $\quad V(w_r, \text{turn}_{\text{me}}) = 1$

1.d) $V(w_r, \text{turn}_{\text{you}}) = 0$:

1 (HL-11) $\quad K, I, w_r \models \neg\text{terminal} \leftrightarrow (\text{turn}_{\text{me}} \leftrightarrow \neg\text{turn}_{\text{you}})$

2 $V(w_r, \text{terminal}) = 0, \models_\neg$ $\quad K, I, w_r \models \neg\text{terminal} \quad \underset{1,2}{\Rightarrow}$

3 $\models_\leftrightarrow, \models_\rightarrow$ $\quad K, I, w_r \models \text{turn}_{\text{me}} \leftrightarrow \neg\text{turn}_{\text{you}}$

4 $V(w_r, \text{turn}_{\text{me}}) = 1$ $\quad K, I, w_r \models \text{turn}_{\text{me}} \quad \underset{3,4}{\Rightarrow}$

5 $\models_\leftrightarrow, \models_\rightarrow$ $\quad K, I, w_r \models \neg\text{turn}_{\text{you}} \quad \Rightarrow$

6 $\models_\neg, \models_{\mathcal{A}tom}$ $\quad V(w_r, \text{turn}_{\text{you}}) = 0$

2.a) $V(w_r, \text{win}_{\text{me}}) = V(w, \text{win}_{\text{you}}) = 0$ — The argument is identical to 1.a).

2.b) $V(w_r, \text{assert}_{\text{me}})(\psi_1 \wedge \psi_2) = 1$, or $V(w_r, \text{assert}_{\text{me}})(\psi_1 \vee \psi_2) = 1$:

1 $(w_r, u) \in R_L$ $\quad K, I, w_r \models \langle L \rangle \top$

2 (HL-24) $\quad K, I, w_r \models \exists x \exists y(\langle L \rangle \top \rightarrow (\text{assert}_{\text{me}}(\ulcorner x \wedge y \urcorner) \vee$
$\text{assert}_{\text{me}}(\ulcorner x \vee y \urcorner) \vee \text{assert}_{\text{you}}(\ulcorner x \wedge y \urcorner) \vee \text{assert}_{\text{you}}(\ulcorner x \vee y \urcorner))) \quad \Rightarrow$

3 $\models_\exists$, there are ψ_1, ψ_2 $\quad K, I_{[x \mapsto \psi_1][y \mapsto \psi_2]}, w_r \models \langle L \rangle \top \rightarrow (\text{assert}_{\text{me}}(\ulcorner x \wedge y \urcorner) \vee$
$\text{assert}_{\text{me}}(\ulcorner x \vee y \urcorner) \vee \text{assert}_{\text{you}}(\ulcorner x \wedge y \urcorner) \vee \text{assert}_{\text{you}}(\ulcorner x \vee y \urcorner)) \quad \underset{1,3}{\Rightarrow}$

4 $\models_\rightarrow$ $\quad K, I_{[x \mapsto \psi_1][y \mapsto \psi_2]}, w_r \models \text{assert}_{\text{me}}(\ulcorner x \wedge y \urcorner) \vee$
$\text{assert}_{\text{me}}(\ulcorner x \vee y \urcorner) \vee \text{assert}_{\text{you}}(\ulcorner x \wedge y \urcorner) \vee \text{assert}_{\text{you}}(\ulcorner x \vee y \urcorner) \quad \Rightarrow$

5 Lemma 4.2.1, $d = me$	$K, I_{[x\mapsto\psi_1][y\mapsto\psi_2]}, w_r \models \text{assert}_{\text{me}}(\ulcorner x \wedge y\urcorner) \vee \text{assert}_{\text{me}}(\ulcorner x \vee y\urcorner)$	$\Rightarrow$
6 $\models_\vee$	$K, I_{[x\mapsto\psi_1][y\mapsto\psi_2]}, w_r \models \text{assert}_{\text{me}}(\ulcorner x \wedge y\urcorner)$	or
	$K, I_{[x\mapsto\psi_1][y\mapsto\psi_2]}, w_r \models \text{assert}_{\text{me}}(\ulcorner x \vee y\urcorner)$	$\Rightarrow$
7 $V(w_r, f_\wedge) = \wedge$, $V(w_r, f_\vee) = \vee$, $TV(x) = \psi_1$,	$V(w_r, \text{assert}_{\text{me}})(\psi_1 \wedge \psi_2) = 1$	or
$TV(y) = \psi_2$	$V(w_r, \text{assert}_{\text{me}})(\psi_1 \vee \psi_2) = 1$	

2.b.i) $V(w_r, \text{assert}_{\text{me}})(\psi_1 \vee \psi_2) = 1$:
$V(u, \text{assert}_{\text{me}})(\psi_1) = 1$:

1 Assumption	$K, I_{[x\mapsto\psi_1][y\mapsto\psi_2]}, w_r \models \text{assert}_{\text{me}}(\ulcorner x \vee y\urcorner)$	
2 (HL-19)	$K, I, w_r \models \forall x \forall y(\text{assert}_{\text{me}}(\ulcorner x \vee y\urcorner) \to \langle L\rangle \text{assert}_{\text{me}}(x))$	$\Rightarrow$
3 $\models_\forall$, especially for ψ_1, ψ_2	$K, I_{[x\mapsto\psi_1][y\mapsto\psi_2]}, w_r \models \text{assert}_{\text{me}}(\ulcorner x \vee y\urcorner) \to \langle L\rangle \text{assert}_{\text{me}}(x)$	$\overset{\Rightarrow}{\scriptstyle 1,3}$
4 $\models_\to$	$K, I_{[x\mapsto\psi_1][y\mapsto\psi_2]}, w_r \models \langle L\rangle \text{assert}_{\text{me}}(x)$	$\Rightarrow$
5 $\models_{\langle L\rangle}$, $(w_r, u) \in R_L$	$K, I_{[x\mapsto\psi_1][y\mapsto\psi_2]}, u \models \text{assert}_{\text{me}}(x)$	$\Rightarrow$
6 $\models_{\mathcal{A}tom}$	$V(u, \text{assert}_{\text{me}})(\psi_1) = 1$	

$V(v, \text{assert}_{\text{me}})(\psi_2) = 1$:

1 Assumption	$K, I_{[x\mapsto\psi_1][y\mapsto\psi_2]}, w_r \models \text{assert}_{\text{me}}(\ulcorner x \vee y\urcorner)$	
2 (HL-20)	$K, I, w_r \models \forall x \forall y(\text{assert}_{\text{me}}(\ulcorner x \vee y\urcorner) \to \langle R\rangle \text{assert}_{\text{me}}(y))$	$\Rightarrow$
3 $\models_\forall$, especially for ψ_1, ψ_2	$K, I_{[x\mapsto\psi_1][y\mapsto\psi_2]}, w_r \models \text{assert}_{\text{me}}(\ulcorner x \vee y\urcorner) \to \langle R\rangle \text{assert}_{\text{me}}(y)$	$\overset{\Rightarrow}{\scriptstyle 1,3}$
4 $\models_\to$	$K, I_{[x\mapsto\psi_1][y\mapsto\psi_2]}, w_r \models \langle R\rangle \text{assert}_{\text{me}}(y)$	$\Rightarrow$
5 $\models_{\langle R\rangle}$, $(w_r, v) \in R_R$	$K, I_{[x\mapsto\psi_1][y\mapsto\psi_2]}, v \models \text{assert}_{\text{me}}(y)$	$\Rightarrow$
6 $\models_{\mathcal{A}tom}$	$V(v, \text{assert}_{\text{me}})(\psi_2) = 1$	

$V(w_r, \text{turn}_{\text{me}}) = 1$ and $V(w_r, \text{turn}_{\text{you}}) = 0$:

1 Assumption	$K, I_{[x\mapsto\psi_1][y\mapsto\psi_2]}, w_r \models \text{assert}_{\text{me}}(\ulcorner x \vee y\urcorner)$	
2 (HL-16)	$K, I, w_r \models \forall x \forall y(\text{assert}_{\text{me}}(\ulcorner x \vee y\urcorner) \to \text{turn}_{\text{me}})$	$\Rightarrow$
3 $\models_\forall$, particularly for ψ_1, ψ_2	$K, I_{[x\mapsto\psi_1][y\mapsto\psi_2]}, w_r \models \text{assert}_{\text{me}}(\ulcorner x \vee y\urcorner) \to \text{turn}_{\text{me}}$	$\overset{\Rightarrow}{\scriptstyle 1,3}$
4 $\models_\to$	$K, I_{[x\mapsto\psi_1][y\mapsto\psi_2]}, w_r \models \text{turn}_{\text{me}}$	$\Rightarrow$
5 $\models_{\mathcal{A}tom}$	$V(w_r, \text{turn}_{\text{me}}) = 1$	

The argument for $V(w_r, \text{turn}_{\text{you}}) = 0$ is as in case 1.d).
2.b.ii) $V(w_r, \text{assert}_{\text{me}})(\psi_1 \wedge \psi_2) = 1$:
The arguments for $V(u, \text{assert}_{\text{me}})(\psi_1) = 1$ and $V(v, \text{assert}_{\text{me}})(\psi_2) = 1$ are similar to case 2.b.i) using (HL-21) and (HL-22) respectively.

It remains to show that $V(w_r, \text{turn}_{\text{you}}) = 1$:

1 Assumption	$K, I_{[x\mapsto\psi_1][y\mapsto\psi_2]}, w_r \models \text{assert}_{\text{me}}(\ulcorner x \wedge y \urcorner)$	
2 (HL-17)	$K, I, w_r \models \forall x \forall y(\text{assert}_{\text{me}}(\ulcorner x \wedge y \urcorner) \rightarrow \text{turn}_{\text{you}})$	$\Rightarrow$
3 $\models_\forall$, particularly for ψ_1, ψ_2	$K, I_{[x\mapsto\psi_1][y\mapsto\psi_2]}, w_r \models \text{assert}_{\text{me}}(\ulcorner x \wedge y \urcorner) \rightarrow \text{turn}_{\text{you}}$	$\underset{1,3}{\Rightarrow}$
4 $\models_\rightarrow$	$K, I_{[x\mapsto\psi_1][y\mapsto\psi_2]}, w_r \models \text{turn}_{\text{you}}$	$\Rightarrow$
5 $\models_{\mathcal{Atom}}$	$V(w_r, \text{turn}_{\text{you}}) = 1$	

The argument for $V(w_r, \text{turn}_{\text{me}}) = 0$ is identical to 1.d), if we use the instance with me and you swapped. □

4.2.2 Every $\mathcal{H}$-game Satisfies the $\mathcal{H}$-game axioms

Having proved that every adequate $\mathcal{H}$-game model represents a valid $\mathcal{H}$-game tree, when *interpreted* as such, we now show the other direction: that every $\mathcal{H}$-game tree interpreted as $\mathcal{H}$-game model satisfies the $\mathcal{H}$-game axioms.

Theorem 4.2.5. Every $\mathcal{H}$-game $G_{\varphi,\nu_{CL},d} = \langle N, E \rangle$ on φ over a classical valuation ν_{CL} *interpreted* as an $\mathcal{H}$-game model $K = K(G)$ satisfies the $\mathcal{H}$-game axioms (Definition 4.1.6).

Proof. Given that $K(G)$ already is an $\mathcal{H}$-game model by our mapping from Definition 4.2.1, it remains to show that K satisfies all $\mathcal{H}$-game axioms.

We group the axioms by expressive similarity and explicitly proof that one of the group holds, since the arguments for the others in the group are analogous.

The symbol $(*)$ in the justifications marks the transition between $\mathcal{H}$-game model and game.

Let w be an arbitrary world in K, and that $l_{assert}(w) = me$[7]:

(HL-1) – (HL-3) $K, I, w \models \langle R \rangle \Gamma \rightarrow [R]\Gamma$:

1 Assumption	$K, I, w \models \langle R \rangle \Gamma$	$\Rightarrow$
2 there exists u, s.t.$(w, u) \in R_R$, $\models_{\langle R \rangle}$	$K, I, u \models \Gamma$	$\Rightarrow$
3 $(*)$, by Definition 4.2.1, $l_{move}((w, u)) = R$, $l_{form}(w) = \varphi \circ \psi, \circ \in \{\wedge, \vee\}$		$\Rightarrow$
4 $(*)w$ has only one outgoing R-labeled edge , $\models_{[R]}$	$K, I, w \models [R]\Gamma$	$\underset{1,4}{\Rightarrow}$
5 $\models_\rightarrow$	$K, I, w \models \langle R \rangle \Gamma \rightarrow [R]\Gamma$	

The last step of the proof, corresponds to an $\rightarrow$-introduction, known from natural deduction calculi for various logics, as can be found in most introductory textbooks to logic, for example *Logic in Computer Science* chapter 1 [HR04].

From now on we omit explicit explanations regarding the mapping between game tree and $\mathcal{H}$-game model and insert an Arabic numeral in boldface, for the respective clause of the $\mathcal{H}$-game definition Definition 3.2.1), in our justifications.

[7] All arguments are symmetric for $l_{assert}(w) = you$

(HL-4) $K, I, w \models \langle L \rangle \top \leftrightarrow \langle R \rangle \top$:

1 Assumption	$K, I, w \models \langle L \rangle \top$	$\Rightarrow$
2 $(w, u) \in R_L, \models_{\langle L \rangle}$	$K, I, u \models \top$	$\Rightarrow$
3 $(*)$, **2** or **3**, there is also v, s.t. $l_{move}((w, v)) = R$	$K, I, v \models \top$	$\Rightarrow$
4 $\models_{\langle R \rangle}$	$K, I, w \models \langle R \rangle \top$	$\Rightarrow$
5 $\models_{\rightarrow}$	$K, I, w \models \langle L \rangle \top \rightarrow \langle R \rangle \top$	

The direction $\langle R \rangle \top \rightarrow \langle L \rangle \top$ is analogous.

(HL-5), (HL-6) $K, I, w \models \langle L \rangle \top \rightarrow [Neg] \bot$:

1 Assumption	$K, I, w \models \langle L \rangle \top$	$\Rightarrow$
2 $(w, u) \in R_L, \models_{\langle L \rangle}$	$K, I, u \models \top$	$\Rightarrow$
3 $(*)$, **2** or **3**, there is no v, s.t. $l_{move}((w, v)) = Neg$	$K, I, w \not\models [Neg] \top$	$\Rightarrow$
4 $\models_{[Neg]}, \models_{\top}, \models_{\bot}$	$K, I, w \models [Neg] \bot$	$\Rightarrow$
5 $\models_{\rightarrow}$	$K, I, w \models \langle L \rangle \top \rightarrow [Neg] \bot$	

The step from 3 to 4 is justified, by the fact that $[Neg]\bot$ is only satisfied in a world without a Neg connection to another world as explained in subsection 2.3.3.

(HL-7) – (HL-9) $K, I, w \models$ terminal $\rightarrow [R] \bot$:

1 Assumption	$K, I, w \models$ terminal	$\Rightarrow$
2 $(*)$, **1**, $w \in N_0$, there is no v, s.t. $l_{move}((w, v)) = R$	$K, I, w \not\models [R] \top$	$\Rightarrow$
3 $\models_{[R]}, \models_{\top}, \models_{\bot}$	$K, I, w \models [R] \bot$	$\Rightarrow$
4 $\models_{\rightarrow}$	$K, I, w \models$ terminal $\rightarrow [R] \bot$	

(HL-10) $K, I, w \models ([L]\bot \wedge [R]\bot \wedge [Neg]\bot) \rightarrow$ terminal:

1 Assumption	$K, I, w \models [L]\bot \wedge [R]\bot \wedge [Neg]\bot$	$\Rightarrow$
2 $(*)$, **1**, $w \in N_0$	$K, I, w \models$ terminal	$\Rightarrow$
3 $\models_{\rightarrow}$	$K, I, w \models ([L]\bot \wedge [R]\bot \wedge [Neg]\bot) \rightarrow$ terminal	$\Rightarrow$

(HL-11) $K, I, w \models \neg$terminal $\leftrightarrow$ (turn$_{\text{me}} \leftrightarrow \neg$turn$_{\text{you}}$)

1 Assumption	$K, I, w \models \neg$terminal	$\Rightarrow$
2 $(*), \models_\neg$, **2 – 4**, there is a u, s.t.$(w, u) \in E$		
$l_{form}(w) \in \{\psi_1 \vee \psi_2, \neg\psi\}$	$K, I, w \models$ turn$_{\text{me}}$	or
$l_{form}(w) = \psi_1 \wedge \psi_2$	$K, I, w \models$ turn$_{\text{you}}$	$\Rightarrow$
3 $(*), l_{turn}(w) = me, \models_\neg$	$K, I, w \models \neg$turn$_{\text{you}}$	or
$l_{turn}(w) = you, \models_\neg$	$K, I, w \models \neg$turn$_{\text{me}}$	$\underset{2,3}{\Rightarrow}$
4 $(*), \models_\leftrightarrow$	$K, I, w \models$ turn$_{\text{me}} \leftrightarrow \neg$turn$_{\text{you}}$	or
$\models_\leftrightarrow$	$K, I, w \models$ turn$_{\text{you}} \leftrightarrow \neg$turn$_{\text{me}}$	$\underset{1,4}{\Rightarrow}$
5 $(*), \models_\leftrightarrow$	$K, I, w \models$ terminal $\leftrightarrow$ (turn$_{\text{me}} \leftrightarrow \neg$turn$_{\text{you}}$)	or
$\models_\leftrightarrow$	$K, I, w \models$ terminal $\leftrightarrow$ (turn$_{\text{you}} \leftrightarrow \neg$turn$_{\text{me}}$)	

The argument for $\models_\leftrightarrow$ in the last two steps, stems from the fact that there is only one l_{turn} per non-terminal node and is not purely logic consequence.

(HL-12) $K, I, w \models$ terminal $\rightarrow \neg$turn$_{\text{me}}$

1 Assumption	$K, I, w \models$ terminal	$\Rightarrow$
2 $(*)$, **1**, $w \in N_0$	$K, I, w \not\models$ turn$_{\text{me}}$	$\underset{1,2}{\Rightarrow}$
3 $\models_\neg, \models_\rightarrow$	$K, I, w \models$ terminal $\rightarrow \neg$turn$_{\text{me}}$	

(HL-13) – (HL-15) : $K, I, w \models \forall x \forall y$(terminal $\rightarrow \neg$assert$_{\text{me}}(\ulcorner x \wedge y \urcorner)$)

1 Assumption	$K, I, w \models$ terminal	$\Rightarrow$
2 $(*)$, **1**, $l_{form}(w) \in \mathcal{Atom}$, there are no ψ_1, ψ_2, s.t. $l_{form}(w) = \psi_1 \wedge \psi_2$, $V(w, f_\wedge) = \wedge$	$K, I_{[x \mapsto \psi_1][y \mapsto \psi_2]}, w \not\models$ assert$_{\text{me}}(\ulcorner x \wedge y \urcorner)$)	$\underset{1,2}{\Rightarrow}$
3 $\models_\neg, \models_\rightarrow$	$K, I_{[x \mapsto \psi_1][y \mapsto \psi_2]}, w \models$ terminal $\rightarrow \neg$assert$_{\text{me}}(\ulcorner x \wedge y \urcorner)$	$\Rightarrow$
4 $\models_\forall$	$K, I, w \models \forall x \forall y$(terminal $\rightarrow \neg$assert$_{\text{me}}(\ulcorner x \wedge y \urcorner)$)	

(HL-16) – (HL-18) $K, I, w \models \forall x \forall y$(assert$_{\text{me}}(\ulcorner x \vee y \urcorner) \rightarrow$ turn$_{\text{me}}$) :

1 Assumption	$K, I_{[x \mapsto \psi_1][y \mapsto \psi_2]}, w \models$ assert$_{\text{me}}(\ulcorner x \vee y \urcorner)$	$\Rightarrow$
2 $(*), V(w, f_\vee) = \vee$, **2**, $l_{turn}(w) = me$	$K, I_{[x \mapsto \psi_1][y \mapsto \psi_2]}, w \models$ turn$_{\text{me}}$	$\underset{1,2}{\Rightarrow}$
3 $\models_\rightarrow$	$K, I_{[x \mapsto \psi_1][y \mapsto \psi_2]}, w \models$ assert$_{\text{me}}(\ulcorner x \vee y \urcorner) \rightarrow$ turn$_{\text{me}}$	$\Rightarrow$
4 $V(w, \text{assert}_{\text{me}}) = \{(\psi_1 \vee \psi_2)\}, \models_\forall$	$K, I, w \models \forall x \forall y$(assert$_{\text{me}}(\ulcorner x \vee y \urcorner) \rightarrow$ turn$_{\text{me}}$)	

The justification for the last step, lies in the fact that $\psi_1 \vee \psi_2$ is the only formula which makes the premise true, for all other $\varphi \in \mathcal{Prop}$ the axiom holds trivially by $\models_\rightarrow$.

(HL-19) – (HL-23) $K, I, w \models \forall x(\text{assert}_{\text{me}}(\ulcorner\neg x\urcorner) \rightarrow \langle Neg\rangle\text{assert}_{\text{you}}(x))$

1 Assumption	$K, I_{[x\mapsto\psi]}, w \models \text{assert}_{\text{me}}(\ulcorner\neg x\urcorner)$	$\Rightarrow$
2 $(*), V(w, f_{\neg}) = \neg$, **4**, there is v, s.t. $l_{move}((w, v)) = Neg$, $l_{form}(v) = \psi$	$K, I_{[x\mapsto\psi]}, v \models \text{assert}_{\text{you}}(x)$	$\Rightarrow_{1,2}$
3 $\models_{\langle Neg\rangle}$	$K, I_{[x\mapsto\psi]}, w \models \langle Neg\rangle\text{assert}_{\text{you}}(x)$	$\Rightarrow_{1,3}$
4 $\models_{\rightarrow}$	$K, I_{[x\mapsto\psi]}, w \models \text{assert}_{\text{me}}(\ulcorner\neg x\urcorner) \rightarrow \langle Neg\rangle\text{assert}_{\text{you}}(x)$	$\Rightarrow$
5 $\models_{\forall}$	$K, I, w \models \forall x(\text{assert}_{\text{me}}(\ulcorner\neg x\urcorner) \rightarrow \langle Neg\rangle\text{assert}_{\text{you}}(x))$	

(HL-24), (HL-25) $K, I, w \models \exists x(\langle Neg\rangle\top \rightarrow \text{assert}_{\text{me}}(\ulcorner\neg x\urcorner) \vee \text{assert}_{\text{you}}(\ulcorner\neg x\urcorner))$

1 Assumption	$K, I, w \models \langle Neg\rangle\top$	$\Rightarrow$
2 $(*), l_{assert}(w) = me$, **4**, there is v, s.t. $l_{move}((w, v)) = Neg$, $l_{form}(w) = \neg\psi$, $V(w, f_{\neg}) = \neg$	$K, I_{[x\mapsto\psi]}, w \models \text{assert}_{\text{me}}(\ulcorner\neg x\urcorner)$	$\Rightarrow$
3 $\models_{\vee}$	$K, I_{[x\mapsto\psi]}, w \models \text{assert}_{\text{me}}(\ulcorner\neg x\urcorner) \vee \text{assert}_{\text{you}}(\ulcorner\neg x\urcorner)$	$\Rightarrow_{1,3}$
3 $\models_{\rightarrow}$	$K, I_{[x\mapsto\psi]}, w \models \langle Neg\rangle\top \rightarrow (\text{assert}_{\text{me}}(\ulcorner\neg x\urcorner) \vee \text{assert}_{\text{you}}(\ulcorner\neg x\urcorner)$	$\Rightarrow$
4 ψ is a witness , $\models_{\exists}$	$K, I, w \models \exists x(\langle Neg\rangle\top \rightarrow \text{assert}_{\text{me}}(\ulcorner\neg x\urcorner) \vee \text{assert}_{\text{you}}(\ulcorner\neg x\urcorner))$	

(HL-26) $K, I, w \models \forall x\forall y\neg(\text{assert}_{\text{me}}(x) \wedge \text{assert}_{\text{you}}(y))$

1 $(*), l_{assert}(w) = me$, $V(w, \text{assert}_{\text{you}}) = \{\}$, for all ψ	$K, I_{[y\mapsto\psi]}, w \not\models \text{assert}_{\text{you}}(y)$	$\Rightarrow_{1,2}$
2 $\models_{\wedge}, \models_{\neg}$	$K, I_{[y\mapsto\psi]}, w \models \neg(\text{assert}_{\text{me}}(x) \wedge \text{assert}_{\text{you}}(y))$	$\Rightarrow$
3 $\models_{\forall}$	$K, I, w \models \forall y\neg(\text{assert}_{\text{me}}(x) \wedge \text{assert}_{\text{you}}(y))$	$\Rightarrow$
4 $\models_{\forall}$	$K, I, w \models \forall x\forall y\neg(\text{assert}_{\text{me}}(x) \wedge \text{assert}_{\text{you}}(y))$	

The justification for the last step stems from the fact that the conjunction remains false, independently of the value of x.

(HL-27) – (HL-28) $K, I, w \models \forall x\forall y(\text{assert}_{\text{me}}(x) \wedge \text{assert}_{\text{me}}(y) \rightarrow x = y)$

1 Assumption	$K, I_{[x\mapsto\psi_1][y\mapsto\psi_2]}, w \models \text{assert}_{\text{me}}(x) \wedge \text{assert}_{\text{me}}(y)$	$\Rightarrow$
2 $(*), l_{form}(w) = \psi$, $V(w, \text{assert}_{\text{me}}) = \{\psi\}$, $\psi_1 = \psi_2 = \psi$	$K, I_{[x\mapsto\psi][y\mapsto\psi]}, w \models \text{assert}_{\text{me}}(x) \wedge \text{assert}_{\text{me}}(y)$	$\Rightarrow$
3 $\models_{=}$	$K, I_{[x\mapsto\psi][y\mapsto\psi]}, w \models x = y$	$\Rightarrow_{1,3}$
4 $\models_{\rightarrow}$	$K, I_{[x\mapsto\psi][y\mapsto\psi]}, w \models \text{assert}_{\text{me}}(x) \wedge \text{assert}_{\text{me}}(y) \rightarrow x = y$	$\Rightarrow$
5 $\models_{\forall}$	$K, I, w \models \forall x\forall y(\text{assert}_{\text{me}}(x) \wedge \text{assert}_{\text{me}}(y) \rightarrow x = y)$	

(HL-29) $K, I, w \models \text{win}_{\text{me}} \rightarrow \text{terminal}$:

1 Assumption	$K, I, w \models \text{win}_{\text{me}}$	$\Rightarrow$
2 $(*), \mathbf{1}, w \in N_0$	$K, I, w \models \text{terminal}$	$\overset{\Rightarrow}{1,2}$
3 $\models_{\rightarrow}$	$K, I, w \models \text{win}_{\text{me}} \rightarrow \text{terminal}$	

(HL-30) – (HL-31) $K, I, w \models \forall x(\text{terminal} \rightarrow ((\text{assert}_{\text{me}}(x) \wedge \text{true}(x)) \rightarrow \text{win}_{\text{me}}))$:

1 Assumption	$K, I[x \mapsto \psi], w \models \text{assert}_{\text{me}}(x) \wedge \text{true}(x)$	
2 Assumption	$K, I, w \models \text{terminal}$	$\overset{\Rightarrow}{1,2}$
3 $(*), \mathbf{1}, w \in N_0$, $\nu_{CL}(\psi) = 1$, $l_{form}(w) = \psi$, $l_{win_{me}}(w) = win$	$K, I[x \mapsto \psi], w \models \text{win}_{\text{me}}$	$\overset{\Rightarrow}{1,3}$
4 $\models_{\rightarrow}$	$K, I[x \mapsto \psi], w \models (\text{assert}_{\text{me}}(x) \wedge \text{true}(x)) \rightarrow \text{win}_{\text{me}}$	$\overset{\Rightarrow}{2,4}$
5 $\models_{\rightarrow}$	$K, I[x \mapsto \psi], w \models \text{terminal} \rightarrow ((\text{assert}_{\text{me}}(x) \wedge \text{true}(x)) \rightarrow \text{win}_{\text{me}})$	$\Rightarrow$
6 $\models_{\forall}$	$K, I, w \models \forall x(\text{terminal} \rightarrow ((\text{assert}_{\text{me}}(x) \wedge \text{true}(x)) \rightarrow \text{win}_{\text{me}}))$	

We have shown that every $\mathcal{H}$-game can be interpreted as an adequate $\mathcal{H}$-game model. □

This concludes the proof. $\mathcal{H}$-game axioms indeed capture all necessary conditions for $\mathcal{H}$-games. All theorems derived from them are automatically valid statements for $\mathcal{H}$-games.

4.3 Adapting the Modeling to a Many-Valued Setting

Just as with the formalization of the evaluation game, the changes needed to expand our modeling to a many-valued context happens in the clauses and axioms for atomic formulas, while leaving most of the rules for decomposing a formula basically untouched.

One fundamental change, which needs to be expressed syntactically and semantically, is the introduction of real truth values for formulas. In the classical case we did encode the truth values into the two unary predicate symbols true^1 and false^1. This approach cannot be carried over to a many-valued setting, since we would need to introduce uncountably many predicate symbols for the truth values.

A more natural translation for the atomic valuation is using a binary predicate symbol value(x, y) meaning that the interpretation of x as a formula yields the interpretation of y as its value in the current valuation. This changes our models quite substantially — the *domain* is not solely the set of propositional formulas anymore, we need to add the real unit-interval to it. A system to formalize concepts which need to talk about multiple domains is *many-sorted logic* — see the chapter by Manzano [Man93] for an overview and introduction.

In order to stay close to our modeling of the classical game we use the translation of a many-sorted language (known as *relativization of quantifiers*) and structure (*unification of domains*) into a single-sorted, or first-order language.

4.3.1 Syntactic Adaptation

By introducing real values for propositional atoms, and explicitly expressing the payoff values for both players, we need to encode subtraction from 1 into the functional part of our language.

Definition 4.3.1 (functional signature for L_{HmvG})**.** We add a unary function symbol f^1_{1-} to the functional signature of L_{HG}[8] from Definition 4.1.1.

The modal signature remains identical to the classical case. The predicate signature on the other hand needs adaptation, akin to the changes of the labeling functions between the classical and the many-valued version of the game and we need to add the qualification predicates for the two sorts of elements in our *unified* domain of reals and propositional formulas.

Definition 4.3.2 (predicate signature for L_{HmvG})**.** The following unary and binary *predicate symbols* are added to the predicate signature of L_{HG} (see Definition 4.1.4):

- prop^1
- real^1
- value^2
- $\mathrm{payoff}^1_{\mathrm{me}}$
- $\mathrm{payoff}^1_{\mathrm{you}}$.

While the following symbols are removed:

- $\mathrm{win}^0_{\mathrm{me}}$
- $\mathrm{win}^0_{\mathrm{you}}$
- true^1
- false^1.

The change from two unary predicates to a binary symbol for the for the possible truth values, enables us to reason about a constant-sum game in a direct fashion.

We refer to the language over this adapted signature as L_{HmvG}.

4.3.2 $\mathcal{H}$-mv-game Models

Adapting our $\mathcal{H}$-game model from Definition 4.1.7 to a many-valued setting is achieved by fitting it to the syntactic changes introduced above:

Definition 4.3.3 ($\mathcal{H}$-mv-game model)**.** A *Kripke model* $K = \langle W, w_r, \{R_R, R_L, R_{Neg}\}, D, V\rangle$ over L_{HmvG}, is called a *$\mathcal{H}$-mv-game model*, if:

- the graph $\langle W, R_R \cup R_L \cup R_{Neg}\rangle$ is as defined in Definition 4.1.7.
- $D = \mathcal{P}rop \cup [0, 1]$, the union of the set of propositional formulas with the real unit-interval.
- $V(w, f^1_{1-})(r) = 1 - r$ for all $w \in W$, in addition to the *interpretation* for the function symbols from Definition 4.1.7.

[8] We use the infix notation with small corners on the syntactic level ($\ulcorner 1 - \cdot \urcorner$) here as well.

- The following clauses are added globally for all worlds to the *predicate interpretation*:
  - $V(w, \text{prop})(\varphi) = 1$, iff $\varphi \in \mathcal{Prop}$, for all $w \in W$
  - $V(w, \text{real})(r) = 1$, iff $r \in [0, 1]$, for all $w \in W$
  - The interpretation of the remaining predicate symbols and propositions is restricted by the axioms and dependent on a concrete game represented by the model.

Let ν_{KZ} be a KZ valuation, $TV(w, \ulcorner\varphi\urcorner) = \varphi$ and $TV(w, \ulcorner r\urcorner) = r$. We say that K *reflects* ν_{KZ}, if:

$$K, I, w \models \text{value}(\ulcorner\varphi\urcorner, \ulcorner r\urcorner) \Leftrightarrow \nu_{KZ}(\varphi) = r$$

hold for all $w \in W$.

All other points from Definition 4.1.7 remain unchanged.

The formal definition of the *mapping* between $\mathcal{H}$-game trees and $\mathcal{H}$-game models (Definition 4.2.1) is directly adapted.

Definition 4.3.4 ($\mathcal{H}$-mv-game tree-model mapping)**.** Starting from the mapping for $\mathcal{H}$-game trees to models (Definition 4.2.1), we drop the items referring to $l_{win_{me}}(u)$ and $l_{win_{you}}(u)$, and add the following:

- $l_{payoff_{me}}(u) = r$, iff $(u, r) \in V(\text{payoff}_{\text{me}})$
- $l_{payoff_{you}}(u) = r$, iff $(u, r) \in V(\text{payoff}_{\text{you}})$

Additionally references to the classical valuation ν_{CL} are replaced by ones for a KZ valuation ν_{KZ}.

4.3.3 Aligning the Axioms

Modifying the $\mathcal{H}$-game axioms is a two-fold process. On the one hand we need to replace the axioms for the winning conditions by formulas, that capture the payoff for both players.

Therefore we drop axioms (HL-29) – (HL-31) and add the following three formulas:

(HmL-32) $\forall x(\text{payoff}_i(x) \to \text{terminal})$

(HmL-33) $\forall x \forall y((\text{prop}(x) \wedge \text{real}(y) \wedge \text{terminal}) \to ((\text{assert}_i(x) \wedge \text{value}(x, y)) \to \text{payoff}_i(y)))$

(HmL-34) $\forall x \forall y((\text{prop}(x) \wedge \text{real}(y) \wedge \text{terminal}) \to ((\text{assert}_i(x) \wedge \text{value}(x, y)) \to \text{payoff}_j(\ulcorner 1 - y\urcorner)))$

On the other hand we need to modify most present axioms to deal with the extended domain by binding the occurring variables to represent propositional formulas through the introduced predicate symbol prop[1]. Formulas not containing any variables can be taken without modification. Thus (HL-1) – (HL-12) are left unmodified. We keep the numbers and the order of the individual $\mathcal{H}$-game axioms as they are. However, to emphasize the addition of the type-predicates we refer to them with a *HmL* prefix. For instance:

(HL-13) $\forall x \forall y(\text{terminal} \rightarrow \neg\text{assert}_i(\ulcorner x \wedge y \urcorner))$, becomes

(HmL-13) $\forall x \forall y(\text{prop}(x) \wedge \text{prop}(y)) \rightarrow (\text{terminal} \rightarrow \neg\text{assert}_i(\ulcorner x \wedge y \urcorner))$.

Adding the type-predicates prop[1] has to happen for all bound variables in each formula, forcing them to be evaluated as propositional formulas as intended.

All $\mathcal{H}$-game axioms are in *prenex normal form*, and all formulas only use either universal or existential quantifiers. For each occurring variable we add the type-predicates joined with a conjunction.

For universally quantified formulas we add the necessary type predicates as a premise of an implication to the axiom, as was done in (HmL-13). Note that if the connective at the root of an axiom is already an implication, equivalently the guard-clauses can be added as conjuncts to the premises of the top-most implication. For instance (HmL-13) is equivalent to $\forall x \forall y((\text{prop}(x) \wedge \text{prop}(y) \wedge \text{terminal}) \rightarrow \neg\text{assert}_i(\ulcorner x \wedge y \urcorner))$. In the existentially quantified case we use a conjunction to join the type predicates to the *matrix*.

Definition 4.3.5 ($\mathcal{H}$-mv-game Axioms)**.** We call the set obtained from (HmL-1) – (HmL-28) in addition to (HmL-32) – (HmL-34), the set of $\mathcal{H}$-mv-game axioms.

4.3.4 Adequateness Results for $\mathcal{H}$-mv-games

The results obtained in Lemma 4.2.1, Lemma 4.2.2 and Lemma 4.2.3 can be directly transferred to the many-valued setting. Lemma 4.2.2 is not referring to any axiom containing variables, and thus the proof remains as is. For the other two it should be obvious that the proof can be transferred with the changes needed to account for the type-predicates.

However we explicitly show the many-valued version of Lemma 4.2.1 in our introduced *terse representation*, to formally show the insignificance of the type-predicates for the result:

Lemma 4.3.1. For every $w \in W$ of an adequate $\mathcal{H}$-mv-game model K there is exactly one formula φ for which one of $\text{assert}_{\text{me}}$ or $\text{assert}_{\text{you}}$ holds in w. We show that $\text{assert}_{\text{me}}$ holds for one formula.

Proof.

1 $\varphi \in \mathcal{Prop}$	$K, I[x \mapsto \varphi], w \models \text{prop}(x)$	
2 $\psi \in \mathcal{Prop}$	$K, I[y \mapsto \psi], w \models \text{prop}(y)$	$\overset{\Rightarrow}{1,2}$
3 $\models_\wedge$	$K, I[x \mapsto \varphi][y \mapsto \psi], w \models \text{prop}(x) \wedge \text{prop}(y)$	
4 (HmL-26)	$K, I, w \models \forall x \forall y((\text{prop}(x) \wedge \text{prop}(y)) \rightarrow$ $\neg(\text{assert}_{\text{me}}(x) \wedge \text{assert}_{\text{you}}(y)))$	$\Rightarrow$
5 $\models_\forall$ for all $\psi_1, \psi_2 \in D$	$K, I[x \mapsto \psi_1][y \mapsto \psi_2], w \models (\text{prop}(x) \wedge \text{prop}(y)) \rightarrow$ $\neg(\text{assert}_{\text{me}}(x) \wedge \text{assert}_{\text{you}}(y))$	$\Rightarrow$

6 especially for	$K, I[x\mapsto\varphi][y\mapsto\psi], w \models (\text{prop}(x) \wedge \text{prop}(y)) \rightarrow$	
$\psi_1 = \varphi, \psi_2 = \psi$	$\neg(\text{assert}_{\text{me}}(x) \wedge \text{assert}_{\text{you}}(y))$	$\overset{\Rightarrow}{3,6}$
7 $\models_{\rightarrow}, \models_{\neg}$	$K, I[x\mapsto\varphi][y\mapsto\psi], w \not\models \text{assert}_{\text{me}}(x) \wedge \text{assert}_{\text{you}}(y)$	
8 (HmL-28)	$K, I, w \models \exists x(\text{prop}(x) \wedge (\text{assert}_{\text{me}}(x) \vee \text{assert}_{\text{you}}(x)))$	$\Rightarrow$
9 $\models_{\exists}$ there is $\varphi \in D$	$K, I[x\mapsto\varphi], w \models \text{prop}(x) \wedge (\text{assert}_{\text{me}}(x) \vee \text{assert}_{\text{you}}(x))$	$\overset{\Rightarrow}{1,9}$
10 $\models_{\wedge}$ weakening	$K, I[x\mapsto\varphi], w \models \text{assert}_{\text{me}}(x) \vee \text{assert}_{\text{you}}(x)$	$\overset{\Rightarrow}{7,10}$
11 either	$K, I[x\mapsto\varphi], w \models \text{assert}_{\text{me}}(x)$	or
12	$K, I[x\mapsto\varphi], w \models \text{assert}_{\text{you}}(x)$	
13	$V(w, \text{assert}_{\text{me}})(\varphi) = 1$ or $V(w, \text{assert}_{\text{you}})(\varphi) = 1$, but not both	

Only one of the two assert_i atomic formulas can hold. Thus only one player is asserting at least one formula in any given world. The argument, that it is exactly one is shown for $i = me$, while referring to the deductions above:

3 φ, ψ arbitrary	$K, I[x\mapsto\varphi][y\mapsto\psi], w \models \text{prop}(x) \wedge \text{prop}(y)$	
11	$K, I[x\mapsto\varphi], w \models \text{assert}_{\text{me}}(x)$	$\overset{\Rightarrow}{3,11}$
14 $\models_{\wedge}$	$K, I[x\mapsto\varphi][y\mapsto\psi], w \models \text{prop}(x) \wedge \text{prop}(y) \wedge \text{assert}_{\text{me}}(x)$	
15 assume $\varphi \neq \psi, \models_{\neg}$	$K, I[x\mapsto\varphi][y\mapsto\psi], w \not\models x = y$	
16 assume		
$V(w, \text{assert}_{\text{me}})(\psi) = 1$	$K, I[x\mapsto\varphi][y\mapsto\psi], w \models \text{assert}_{\text{me}}(y)$	$\overset{\Rightarrow}{14,16}$
17 $\models_{\wedge}$	$K, I[x\mapsto\varphi][y\mapsto\psi], w \models \text{prop}(x) \wedge \text{prop}(y) \wedge \text{assert}_{\text{me}}(x) \wedge \text{assert}_{\text{me}}(y)$	$\overset{\Rightarrow}{15,17}$
18 $\models_{\rightarrow}$	$K, I[x\mapsto\varphi][y\mapsto\psi], w \not\models (\text{prop}(x) \wedge \text{prop}(y) \wedge \text{assert}_{\text{me}}(x) \wedge$	
	$\text{assert}_{\text{me}}(y)) \rightarrow x = y$	
19 (HmL-27)	$K, I, w \models \forall x \forall y(\text{prop}(x) \wedge \text{prop}(y) \wedge \text{assert}_{\text{me}}(x) \wedge$	
	$\text{assert}_{\text{me}}(y)) \rightarrow x = y$	$\overset{\Rightarrow}{18,19}$
20 $\models_{\forall}$	contradiction to $\varphi \neq \psi$	

As in the modeling of the classical case only one formula is asserted by only one player in each state. □

Theorem 4.3.2. Every adequate $\mathcal{H}$-mv-game model K for a formula φ with initial defender d *reflecting* a KZ-valuation ν_{KZ} corresponds to an $\mathcal{H}$-mv-game for φ over ν_{KZ} $G_{\varphi,\nu_{KZ},d}$.

Proof. The argument proceeds by structural induction on the height of T_K and is similar to the proof for Theorem 4.2.4.

For the base case the main focus is showing that the altered labeling functions for the *payoff* are as required. However we show all of the following points explicitly, to demonstrate that the type-predicates do not introduce any fundamental difference. We show the case $d = me$, $d = you$ is symmetric:

a) The single formula φ for which we have $V(w, \text{assert}_{\text{me}})(\varphi) = 1$ is atomic.

b) Assuming $\nu_{KZ}(\varphi) = r$ we have $l_{payoff_{me}}(w) = r$ and $l_{payoff_{you}}(w) = 1 - r$.

c) There is no turn label for w, i.e. $V(w, \text{turn}_{\text{me}}) = V(w, \text{turn}_{\text{you}}) = 0$.

a) $V(w, \text{assert}_{\text{me}})(\varphi) = 1$, for $\varphi \in \mathcal{Atom}$. As shown in Theorem 4.2.4 we have $V(w, \text{terminal}) = 1$.
By Definition 4.3.3, we have $V(w, \text{prop})(\varphi) = 1$, for all $\varphi \in \mathcal{Prop}$.

1 K is for φ, $TV(\ulcorner\varphi\urcorner) = \varphi$	$K, I, w \models \text{assert}_{\text{me}}(\ulcorner\varphi\urcorner)$	
2 (HmL-13)	$K, I, w \models \forall x \forall y((\text{prop}(x) \land \text{prop}(y)) \rightarrow$ $(\text{terminal} \rightarrow \neg\text{assert}_{\text{me}}(\ulcorner x \land y\urcorner)))$	$\Rightarrow$
3 $\models_\forall$ for all pairs $\psi_1, \psi_2 \in \mathcal{Prop}$	$K, I[x \mapsto \psi_1][y \mapsto \psi_2], w \models (\text{prop}(x) \land \text{prop}(y)) \rightarrow$ $(\text{terminal} \rightarrow \neg\text{assert}_{\text{me}}(\ulcorner x \land y\urcorner))$	
4 K is $\mathcal{H}$-mv-game model, $\psi_1, \psi_2 \in \mathcal{Prop}, \models_\land$	$K, I[x \mapsto \psi_1][y \mapsto \psi_2], w \models \text{prop}(x) \land \text{prop}(y)$	$\overset{\Rightarrow}{3,4}$
5 $\models_\rightarrow$	$K, I[x \mapsto \psi_1][y \mapsto \psi_2], w \models (\text{terminal} \rightarrow \neg\text{assert}_{\text{me}}(\ulcorner x \land y\urcorner))$	$\Rightarrow$
6 $V(w, \text{terminal}) = 1, \models_\rightarrow, \models_\neg$	$K, I[x \mapsto \psi_1][y \mapsto \psi_2], w \not\models \text{assert}_{\text{me}}(\ulcorner x \land y\urcorner)$	$\Rightarrow$
7 $V(w, f_\land) = \land$, for all ψ_1, ψ_2	$V(w, \text{assert}_{\text{me}})(\psi_1 \land \psi_2) = 0$	$\overset{\Rightarrow}{1,7}$
8 for any $\psi_1, \psi_2 \in \mathcal{Prop}$	$TV(\ulcorner\varphi\urcorner) = \varphi \neq \psi_1 \land \psi_2$	

The formula asserted in w cannot be a conjunction. Equivalent deductions can be made with (HmL-14) and (HmL-15) to show that it is neither a disjunction nor a negation, leaving only an atom as possible asserted function.

This proof illustrates that the addition of the type-predicates introduces two further inference steps — referring to the definition of $\mathcal{H}$-mv-game models to show that $V(w, \text{prop})(\varphi) = 1$ for all $\varphi \in \mathcal{Prop}$ and using that to eliminate the premise of the root implication. It is clear that these steps can be performed in all inferences used in the proof of Theorem 4.2.4.
b) Assume $\nu_{KZ}(\varphi) = r$.

1 $\varphi \in \mathcal{Prop}$, $TV(\ulcorner\varphi\urcorner) = \varphi$	$K, I, w \models \text{prop}(\ulcorner\varphi\urcorner)$	
2 $r \in [0,1]$, $TV(\ulcorner r\urcorner) = r$	$K, I, w \models \text{real}(\ulcorner r\urcorner)$	
3 $V(w, \text{terminal}) = 1$	$K, I, w \models \text{terminal}$	$\overset{\Rightarrow}{1,2,3}$
4 $\models_\land$	$K, I, w \models \text{prop}(\ulcorner\varphi\urcorner) \land \text{real}(\ulcorner r\urcorner) \land \text{terminal}$	
5 K is for φ, $TV(\ulcorner\varphi\urcorner) = \varphi$	$K, I, w \models \text{assert}_{\text{me}}(\ulcorner\varphi\urcorner)$	
6 K reflects $\nu_{KZ}, \nu_{KZ}(\varphi) = r$, $TV(\ulcorner r\urcorner) = r$	$K, I, w \models \text{value}(\ulcorner\varphi\urcorner, \ulcorner r\urcorner)$	$\overset{\Rightarrow}{5,6}$
7 $\models_\land$	$K, I, w \models \text{assert}_{\text{me}}(\ulcorner\varphi\urcorner) \land \text{value}(\ulcorner\varphi\urcorner, \ulcorner r\urcorner)$	
8 (HmL-33)	$K, I, w \models \forall x \forall y((\text{prop}(x) \land \text{real}(y) \land \text{terminal}) \rightarrow$ $((\text{assert}_{\text{me}}(x) \land \text{value}(x, y)) \rightarrow \text{payoff}_{\text{me}}(y)))$	

9 $\models_{\forall}$, for all $\psi, r' \in D$	$K, I_{[x\mapsto\psi][y\mapsto r']}, w \models (\text{prop}(x) \wedge \text{real}(y) \wedge \text{terminal}) \rightarrow$ $((\text{assert}_{\text{me}}(x) \wedge \text{value}(x,y)) \rightarrow \text{payoff}_{\text{me}}(y))$	
10 particularly for $\psi = \varphi, r' = r$	$K, I_{[x\mapsto\varphi][y\mapsto r]}, w \models (\text{prop}(x) \wedge \text{real}(y) \wedge \text{terminal}) \rightarrow$ $((\text{assert}_{\text{me}}(x) \wedge \text{value}(x,y)) \rightarrow \text{payoff}_{\text{me}}(y))$	$\overset{\Rightarrow}{4,10}$
11 $TV(w, \ulcorner\varphi\urcorner) = \varphi$, $TV(w, \ulcorner r\urcorner) = r, \models_{\rightarrow}$	$K, I_{[x\mapsto\varphi][y\mapsto r]}, w \models (\text{assert}_{\text{me}}(x) \wedge \text{value}(x,y)) \rightarrow$ $\text{payoff}_{\text{me}}(y)$	$\overset{\Rightarrow}{7,11}$
12 $TV(w, \ulcorner\varphi\urcorner) = \varphi$, $TV(w, \ulcorner r\urcorner) = r, \models_{\rightarrow}$	$K, I_{[x\mapsto\varphi][y\mapsto r]}, w \models \text{payoff}_{\text{me}}(y)$	$\Rightarrow$
13 $\models_{\mathcal{A}tom}$	$V(w, \text{payoff}_{\text{me}})(r) = 1$	

Thus the payoff-label, and the value of the game, for me is r. Showing that the value for you is $1 - r$ needs premises from above and refers to the derivation above:

14 (HmL-34)	$K, I, w \models \forall x \forall y((\text{prop}(x) \wedge \text{real}(y) \wedge \text{terminal}) \rightarrow$ $((\text{assert}_{\text{me}}(x) \wedge \text{value}(x,y)) \rightarrow \text{payoff}_{\text{you}}(\ulcorner 1 - y\urcorner)))$	
15 $\models_{\forall}$, particularly for $\varphi, r \in D$	$K, I_{[x\mapsto\varphi][y\mapsto r]}, w \models (\text{prop}(x) \wedge \text{real}(y) \wedge \text{terminal}) \rightarrow$ $((\text{assert}_{\text{me}}(x) \wedge \text{value}(x,y)) \rightarrow \text{payoff}_{\text{you}}(\ulcorner 1 - y\urcorner))$	$\overset{\Rightarrow}{4,15}$
16 $TV(w, \ulcorner\varphi\urcorner) = \varphi$, $TV(w, \ulcorner r\urcorner) = r, \models_{\rightarrow}$	$K, I_{[x\mapsto\varphi][y\mapsto r]}, w \models (\text{assert}_{\text{me}}(x) \wedge \text{value}(x,y)) \rightarrow$ $\text{payoff}_{\text{you}}(\ulcorner 1 - y\urcorner))$	$\overset{\Rightarrow}{7,16}$
17 $TV(w, \ulcorner\varphi\urcorner) = \varphi$, $TV(w, \ulcorner r\urcorner) = r, \models_{\rightarrow}$	$K, I_{[x\mapsto\varphi][y\mapsto r]}, w \models \text{payoff}_{\text{you}}(\ulcorner 1 - y\urcorner)$	$\Rightarrow$
18 $V(w, f^1_{1-})(r) = 1 - r$	$V(w, \text{payoff}_{\text{you}})(1 - r) = 1$	

The proof for point c) is identical to the one from the proof Theorem 4.2.4, since (HmL-12) contains no variables.

Most of the argumentation for the inductive step can be readily adapted from the proof for the classical $\mathcal{H}$-game model.

The only change, which needs to be shown explicitly is the non-existence of payoff labels in *intermediate worlds*. We show it explicitly for negation being the root connective of the asserted formula. The argument for $\wedge, \vee$ is identical: Assume the root-world w_r has a single *Neg* transition to a world v. This entails $K, I, w_r \not\models$ terminal.[9]

1 $V(w_r, \text{terminal}) = 0$	$K, I, w_r \not\models \text{terminal}$	
2 (HmL-32)	$K, I, w_r \models \forall x(\text{payoff}_{\text{me}}(x)) \rightarrow \text{terminal}$	$\Rightarrow$
3 $\models_\forall$, for all $r \in D$	$K, I[x\mapsto r], w_r \models (\text{payoff}_{\text{me}}(x)) \rightarrow \text{terminal}$	$\overset{\Rightarrow}{_{1,3}}$
4 $\models_\rightarrow$	$K, I[x\mapsto r], w_r \not\models \text{payoff}_{\text{me}}(x)$	$\Rightarrow$
5 $\models_\neg, \models_\forall$	$K, I, w_r \models \forall x \neg \text{payoff}_{\text{me}}(x)$	$\Rightarrow$
6 for all $r \in D$	$V(w_r, \text{payoff}_{\text{me}})(r) = 0$	

We conclude that every $\mathcal{H}$-mv-game model for φ reflecting ν_{KZ}, which satisfies the $\mathcal{H}$-mv-game axioms, corresponds to an $\mathcal{H}$-mv-game. □

Showing that the many-valued $\mathcal{H}$-mv-games do satisfy the $\mathcal{H}$-mv-game axioms is the last step missing towards our goal of proving that our axiomatization indeed captures formal evaluation games for KZ-logic.

Theorem 4.3.3. Every $\mathcal{H}$-mv-game $G_{\varphi,\nu_{KZ},d} = \langle N, E\rangle$ on φ over a KZ valuation ν_{KZ} *interpreted as* an $\mathcal{H}$-mv-game model $K = K(G)$ satisfies the $\mathcal{H}$-mv-game axioms.

Proof. $K(G)$ is, by our mapping from Definition 4.3.4, an $\mathcal{H}$-mv-game model (Definition 4.3.3). It remains to show that K satisfies all $\mathcal{H}$-mv-game axioms.

The axioms (HmL-1) – (HmL-12) remain unchanged from the classical case.

We first show that the new axioms (HmL-32) – (HmL-34) hold, focusing on $d = me$. The case $d = you$ is symmetric.

(HmL-32) $K, I, w \models \forall x\ \text{payoff}_{\text{me}}(x) \rightarrow \text{terminal}$

1 Assumption	$K, I[x\mapsto r], w \models \text{payoff}_{\text{me}}(x)$	$\Rightarrow$
2 $(*)$, **1**, $w \in N_0$	$K, I[x\mapsto r], w \models \text{terminal}$	$\overset{\Rightarrow}{_{1,2}}$
3 $\models_\rightarrow$	$K, I[x\mapsto r], w \models \text{payoff}_{\text{me}}(x)) \rightarrow \text{terminal}$	$\Rightarrow$
4 $\models_\forall$	$K, I, w \models \forall x\ \text{payoff}_{\text{me}}(x)) \rightarrow \text{terminal}$	

The argument for the last inference step is that the implication holds for all domain elements with $V(w, \text{payoff})(d) = 0$ by definition of $\models_\rightarrow$.

[9] Argument exactly like in item 1.a) in the proof of Theorem 4.2.4.

(HmL-33) $K, I, w \models \forall x \forall y((\text{prop}(x) \wedge \text{real}(y) \wedge \text{terminal}) \rightarrow ((\text{assert}_{\text{me}}(x) \wedge \text{value}(x, y)) \rightarrow \text{payoff}_{\text{me}}(y)))$

1 Assumption	$K, I_{[x \mapsto \psi][y \mapsto r]}, w \models \text{prop}(x) \wedge \text{real}(y) \wedge \text{terminal}$	
2 Assumption	$K, I_{[x \mapsto \psi][y \mapsto r]}, w \models \text{assert}_{\text{me}}(x) \wedge \text{value}(x, y)$	$\overset{\Rightarrow}{_{1,2}}$
3 $(*)$, **1**, $w \in N_0$, $r \in [0,1], \psi \in \mathcal{Prop}$, $\nu_{KZ}(\psi) = r$, $l_{form}(w) = \psi$, $l_{assert}(w) = me$, $l_{payoff_{\text{me}}}(w) = r$	$K, I_{[x \mapsto \psi][y \mapsto r]}, w \models \text{payoff}_{\text{me}}(y)$	$\overset{\Rightarrow}{_{2,3}}$
4 $\models_{\rightarrow}$	$K, I_{[x \mapsto \psi][y \mapsto r]}, w \models (\text{assert}_{\text{me}}(x) \wedge \text{value}(x, y)) \rightarrow \text{payoff}_{\text{me}}(y)$	$\overset{\Rightarrow}{_{1,4}}$
5 $\models_{\rightarrow}$	$K, I_{[x \mapsto \psi][y \mapsto r]}, w \models (\text{prop}(x) \wedge \text{real}(y) \wedge \text{terminal}) \rightarrow ((\text{assert}_{\text{me}}(x) \wedge \text{value}(x, y)) \rightarrow \text{payoff}_{\text{me}}(y))$	$\Rightarrow$
6 $\models_{\forall}$	$K, I, w \models \forall x \forall y(\text{prop}(x) \wedge \text{real}(y) \wedge \text{terminal}) \rightarrow ((\text{assert}_{\text{me}}(x) \wedge \text{value}(x, y)) \rightarrow \text{payoff}_{\text{me}}(y))$	

(HmL-34) $K, I, w \models \forall x \forall y((\text{prop}(x) \wedge \text{real}(y) \wedge \text{terminal}) \rightarrow ((\text{assert}_{\text{me}}(x) \wedge \text{value}(x, y)) \rightarrow \text{payoff}_{\text{you}}(\ulcorner 1 - y \urcorner)))$

1 Assumption	$K, I_{[x \mapsto \psi][y \mapsto r]}, w \models \text{prop}(x) \wedge \text{real}(y) \wedge \text{terminal}$	
2 Assumption	$K, I_{[x \mapsto \psi][y \mapsto r]}, w \models \text{assert}_{\text{me}}(x) \wedge \text{value}(x, y)$	$\overset{\Rightarrow}{_{1,2}}$
3 $(*)$, **1**, $w \in N_0, r \in [0,1]$, $\psi \in \mathcal{Prop}, \nu_{KZ}(\psi) = r$, $l_{form}(w) = \psi, l_{assert}(w) = me$, $V(w, f_{1-})(r) = 1 - r$, $l_{payoff_{\text{you}}}(w) = 1 - r$	$K, I_{[x \mapsto \psi][y \mapsto r]}, w \models \text{payoff}_{\text{you}}(\ulcorner 1 - y \urcorner)$	$\overset{\Rightarrow}{_{2,3}}$
4 $\models_{\rightarrow}, \models_{\rightarrow}, \models_{\forall}$	$K, I, w \models \forall x \forall y(\text{prop}(x) \wedge \text{real}(y) \wedge \text{terminal}) \rightarrow ((\text{assert}_{\text{me}}(x) \wedge \text{value}(x, y)) \rightarrow \text{payoff}_{\text{you}}(\ulcorner 1 - y \urcorner))$	

It remains to show that (HmL-13) – (HmL-28) are satisfied by an $\mathcal{H}$-mv-game tree, when interpreted as $\mathcal{H}$-mv-game model:

(HmL-13) $K, I, w \models \forall x \forall y(\text{prop}(x) \wedge \text{prop}(y)) \rightarrow (\text{terminal} \rightarrow \neg\text{assert}_{\text{me}}(\ulcorner x \wedge y \urcorner))$

1 Assumption	$K, I, w \models \text{terminal}$	$\Rightarrow$
2 $(*)$, **1**, $l_{form}(w) \in \mathcal{Atom}$, there are no $\psi_1, \psi_2 \in \mathcal{Prop}$, s.t. $l_{form}(w) = \psi_1 \wedge \psi_2$, $V(w, f_\wedge) = \wedge$	$K, I[x \mapsto \psi_1][y \mapsto \psi_2], w \not\models \text{assert}_{\text{me}}(\ulcorner x \wedge y \urcorner)$	$\overset{\Rightarrow}{2,3}$
3 $\models_\neg, \models_\rightarrow$	$K, I[x \mapsto \psi_1][y \mapsto \psi_2], w \models \text{terminal} \rightarrow \neg\text{assert}_{\text{me}}(\ulcorner x \wedge y \urcorner)$	
4 $l_{form}(w) = \psi_1 \wedge \psi_2 \in \mathcal{Prop}$	$K, I[x \mapsto \psi_1][y \mapsto \psi_2], w \models \text{prop}(x) \wedge \text{prop}(y)$	$\overset{\Rightarrow}{3,4}$
5 $\models_\rightarrow$	$K, I[x \mapsto \psi_1][y \mapsto \psi_2], w \models (\text{prop}(x) \wedge \text{prop}(y)) \rightarrow (\text{terminal} \rightarrow \neg\text{assert}_{\text{me}}(\ulcorner x \wedge y \urcorner))$	$\Rightarrow$
6 $\models_\forall$	$K, I, w \models \forall x \forall y(\text{prop}(x) \wedge \text{prop}(y)) \rightarrow (\text{terminal} \rightarrow \neg\text{assert}_{\text{me}}(\ulcorner x \wedge y \urcorner))$	

(HmL-25) $K, I, w \models \exists x(\text{prop}(x) \wedge (\langle Neg \rangle \top \rightarrow \text{assert}_{\text{me}}(\ulcorner \neg x \urcorner) \vee \text{assert}_{\text{you}}(\ulcorner \neg x \urcorner)))$

1 Assumption	$K, I, w \models \langle Neg \rangle \top$	$\Rightarrow$
2 $(*)$, $l_{assert}(w) = me$, **4**, there is v, s.t. $l_{move}((w, v)) = Neg$, $l_{form}(w) = \neg\psi$	$K, I[x \mapsto \psi], w \models \text{assert}_{\text{me}}(\ulcorner \neg x \urcorner)$	$\Rightarrow$
3 $\models_\vee$	$K, I[x \mapsto \psi], w \models \text{assert}_{\text{me}}(\ulcorner \neg x \urcorner) \vee \text{assert}_{\text{you}}(\ulcorner \neg x \urcorner)$	$\overset{\Rightarrow}{1,3}$
4 $\models_\rightarrow$	$K, I[x \mapsto \psi], w \models \langle Neg \rangle \top \rightarrow (\text{assert}_{\text{me}}(\ulcorner \neg x \urcorner) \vee \text{assert}_{\text{you}}(\ulcorner \neg x \urcorner)$	
5 $l_{form}(w) = \psi \in \mathcal{Prop}$	$K, I[x \mapsto \psi], w \models \text{prop}(x)$	$\overset{\Rightarrow}{4,5}$
6 $\models_\wedge$	$K, I[x \mapsto \psi], w \models \text{prop}(x) \wedge (\langle Neg \rangle \top \rightarrow \text{assert}_{\text{me}}(\ulcorner \neg x \urcorner) \vee \text{assert}_{\text{you}}(\ulcorner \neg x \urcorner))$	$\Rightarrow$
4 ψ is a witness , $\models_\exists$	$K, I, w \models \exists x(\text{prop}(x) \wedge (\langle Neg \rangle \top \rightarrow \text{assert}_{\text{me}}(\ulcorner \neg x \urcorner) \vee \text{assert}_{\text{you}}(\ulcorner \neg x \urcorner)))$	

From the proof for the two axioms above, we see that the type predicates are satisfied since the labeling function for the currently asserted formula has $\mathcal{Prop}$ as range. ($l_{form} : N \rightarrow \mathcal{Prop}$).

We omit explicitly giving proofs for the remaining axioms. □

The axiomatization was extended in a straight-forward manner, and the equivalence results are working just like in the classical setting.

CHAPTER 5

Conclusion and Outlook

5.1 Reflection and Conclusion

This thesis was originally started in December 2011, the scope and expected main results were defined and laid out before September 2012[1]. In the time passed until completion, the comparatively young and highly active research in the intersection between mathematical fuzzy logic and game theory, has proved to be very rewarding and has advanced considerably. It even has matured to a point of deserving a dedicated chapter in the third volume of the *Handbook of Mathematical Fuzzy Logic* [Fer16, CFN16]. We want to highlight the paper by Fermüller [Fer14], which provides a well laid-out bridge between previous presentations of fuzzy evaluation games and Giles' Game along with several extensions. The characterization of the $\mathcal{H}$-mv-game, in the paper was instructive in closing a few minor gaps in the formalizations specified here.

Initially the work was planned to investigate three logical evaluation games with the apparatus of Game Logic as defined in the *Handbook of Modal Logic* [BvBW07]. This proved too broad as the scope of a Master's thesis and additionally a few technical omissions in the previous definitions of the evaluation games needed addressing in advance. The logical axiomatization of the game trees as presented in chapter 4, which in turn resulted in the alignment of our $\mathcal{H}$-games and $\mathcal{H}$-mv-games with game-theoretical primitives as introduced in chapter 3 is the result of these adaptations.

The precise definition of the evaluation games as extensive games of perfect information, along with accompanying notions of strategies, as developed here, is, to our best knowledge, unique to this thesis, although they were discussed in a more abstract manner in previous work [vB03, FR12, Fer14]. Introducing negation as a regular move of one player "giving" the formula to the other, has been hinted at in other papers [MC09, Fer14], its definition in context of formal game trees was accomplished here. The axiomatization of these trees with multi-modal logic, might seem as a simple and trivial exercise on the first sight, but proved particularly useful in finding those spots in the definitions, that needed further attention. The extensive and com-

[1]The author's career advancements led to an unpredictable interruption for five years.

prehensive proofs conducted in section 4.2 and subsection 4.3.4 constitutes a large amount of diligent work. A necessary task, if we hope to generate new knowledge from our axiomatization for which a master's thesis provides an adequate setting.

The use of formal logic to get a very deep understanding of a topic, is probably the central motivation for logic and in no way novel. However, axiomatizing evaluation game trees by using modal logic along with semantic correspondence proofs is a contribution of this thesis.

The merit of a formal logical axiomatization is expressed very aptly by Pauly and Parikh:

> At this point the reader may have the feeling, "If all that logic does is to tell us a laborious way to find out something we already knew, then aren't we better off without logic?"
>
> We offer an analogy to answer this charge which is fair. Suppose you are going from your house to the nearby drugstore. You do not look at the ground because you know the way. On arriving at the store you discover you no longer have your wallet which you did take when you left the house. Now you will go back over the same route, and you will look carefully at each step on the way. So logic may be needed when something is lost, or when you need to make sure nothing will be lost. And then the extra care and the extra labor are worth it.
>
> Moreover, doing a logically correct proof, even an informal one, makes one realize that objects we use in real life have logical properties and we use them to ensure the correctness of algorithms....
>
> A logical analysis can reveal hidden assumptions and can put us on our guard when these assumptions fail. [PP03]

Summarizing we think that this thesis presented us with a good opportunity to apply the theoretical knowledge acquired during the master's studies in an exciting field. Working independently and deciding freely on a concrete presentation of the work, gave insight into the challenges encountered in theoretical work[2]. Challenges, that were far less evident during regular courses with a clear laid out path.

On a final note, we want to point out another very positive development that took place during the past six years with respect to scientific research, and that probably is not so evident for people working in science on a daily basis: the amount and quality of scientific resources readily available on-line has increased significantly over the timespan. Searching for papers, textbooks and other resources used to constitute a substantial part of the time spent on researching and led to pay-walls and second-rate copies or long visits to libraries, if the material was available at all. Nowadays most papers, chapters, or whole books, possibly in a pre-print version are among the first hits of a web-search. This observation was a particularly pleasant surprise, noticed upon starting to work on this thesis again.

[2]A minor example is our design choice of including $\top$ and $\bot$ into the atomic valuation, since they cannot be defined by the other connectives in KZ logic.

5.2 Open Questions and Future Work

As noted above, and as evident in the extensive treatments of mathematical fuzzy logic and modal logic in chapter 2, this thesis originally had a vaster scope, primarily focusing on game logic. We use this section to shortly sketch and conjecture a few ideas, that we deem worth investigating further.

5.2.1 Axiomatizing $\mathcal{H}$-mv-games Using Many-sorted Logic

In the extension of the axiomatization from $\mathcal{H}$-games to $\mathcal{H}$-mv-games undertaken in section 4.3, we aimed primarily for minimizing the changes to the classical axiomatization. This motivated the use of type predicates to address the many-sorted nature of the game for KZ-logic. The conservative approach has its downside in being verbose on the syntactic level, and thus hiding the straight-forward intuition of having formulas and their values as two separate entities to model.

A more concise and elegant axiomatization could be achieved by using a many-sorted language for the axiomatization, modeling formulas and reals as separate sorts.

The use of many-sorted logic could either be restricted to indexing the quantifiers and the predicate- and function symbols, preserving the first-order modal satisfiability relation with minor adaptations.

Alternatively it would also be possible to adapt the approach presented in the textbook *Many-sorted Logics and Its Applications* [Man93], which defines truth-values on the meta-level as a further sort. We feel that this approach would emphasize the difference between truth on the object level of the evaluation games and truth on the meta level. However, great care needs to be taken in the translation of the modal component into the many-sorted formalism.

5.2.2 Describing Solution Concepts with Modalities

The modal axiomatization in chapter 4 captures the logical evaluation games trees from chapter 3. A logical axiomatization makes certain implicit assumptions explicit, but it does not address many fundamental questions, that arise when looking at the games. The prime example is that the current axiomatization is not expressive enough to reproduce the equivalence result of the games to Tarski semantics.

The modeling lacks axioms for describing notions like strategies, strategy profiles and means to compare them with respect to the utility functions. We cannot find the equilibria of the games or succinctly define a winning strategy.

Two very promising approaches [Bon02, HMvdHW03] for augmenting our formalization are described with unified notation in chapter 14 of the *Handbook of Modal Logic*, where the results presented below are taken from. In both papers game theoretic concepts are directly represented by modalities.

The first one [Bon02] introduces one modality R_i per player i, which connects a world w where i is active to all worlds reachable from w. Additionally it has a *recommendation* modality R_* as primitive, used to model the backward-induction vector in a game tree, and thus the path of the subgame perfect Nash equilibrium. R_* is a subset of the transitive closure of the union of all players' modalities: $R_* \subseteq R^T$[3]. At each intermediate node a recommendation must be made, and for $(w, u) \in R_*$ all other paths connecting w and u in R^T need to be in R_*. Additionally they use propositions to model outcomes of the game $u_i = p_i$ and players preferences $(q \leq p)$. With this interpretation the formula:

$$\langle * \rangle (u_i = p_i) \rightarrow [i](((u_i = q_i) \vee \langle * \rangle (u_i = q_i)) \rightarrow q_i \leq p_i)$$

describes a subgame perfect Nash equilibrium.

The second paper [HMvdHW03] takes preference relations $[i]$ for each player i, strategy profiles $[\sigma]$ and strategy profiles, where one player deviates from a given recommendation $[i, \sigma]$ as first-class modalities. The system obtained is called *Extensive Game Logic*. For a model M over these modalities a subgame perfect Nash equilibrium (σ) translates to[4]:

$$M \models \bigwedge_{i \in Pl} (\langle i, \sigma \rangle [i] \varphi \rightarrow [\sigma] \varphi)$$

Combining the ideas of these approaches with our modeling seems very promising. For the first one our axiomatization already provides the needed propositions for the turn indicators — we need not define the modalities for the players externally. Additionally our signature contains the predicates and propositions for the payoff of a game.

On the other hand we can also model the recommendation with logic. After all, our logical evaluation games serve as alternative characterization to logical truth, which is present in the satisfiability relation of the modal meta logic. In fact an early experiment for our classical evaluation games already contained the following axioms:

VAL-1) $\forall x(\text{true}(x) \leftrightarrow \neg\text{false}(x))$

VAL-2) $\forall x \forall y((\text{true}(x) \vee \text{true}(y)) \rightarrow \text{true}(\ulcorner x \vee y \urcorner))$

VAL-3) $\forall x \forall y((\text{true}(x) \wedge \text{true}(y)) \rightarrow \text{true}(\ulcorner x \wedge y \urcorner))$

VAL-4) $\forall x(\text{false}(x) \rightarrow \text{true}(\ulcorner \neg x \urcorner))$

These could be used to define the recommendation relation from [Bon02] directly within the formal language. For the $\mathcal{H}$-mv-games the valuation would need to be encoded with functions symbols $(\min, \max, 1-)$ instead.

Note that both papers consider only *generic games*. Logical evaluation games do not belong to this category; in the game for the formula $\varphi \wedge \varphi$ the players have no preference in which leaf the game ends. This implies that, at least, the functionality axiom for the strategy profile modalities needs to be relaxed to partial functionality.

[3] With R^T being the transitive closure of $\bigcup_{i \in Pl} R_i$

[4] $M \models \varphi$ means that φ holds in all worlds.

5.2.3 Axiomatizing Games for Stronger Languages

Our axiomatization were done for classical logic as a familiar field for initial experimentation, and afterwards extended to KZ logic, due to the pleasant properties of its semantics, and due to its vast scope of applications. However, Hájeks framework, imposes the question of extending any result accomplished for KZ logic to a richer logic, or even the class of t-norm based fuzzy logics.

As indicated in subsection 2.5.3 many approaches for providing game-theoretic semantics to fuzzy logics exist, and Łukasiewicz logic usually provides the prime focus for the initial discussion. The rules for the t-norm connectives: strong conjunction, implication, and strong disjunction need to address the central point that evaluation in propositional Łukasiewicz logic cannot be directly achieved solely by looking at one atomic subformula.

For the two approaches sketched in subsection 2.5.3, we feel that both should best be addressed in a many-sorted reformulation of the current axiomatization, although the Hintikka style game introduced by Majer and Cintula [MC09], that contains a value in addition to a formula in its states could be axiomatized in our current formulation. Giles Game on the other hand would profit to a far greater extent from a many-sorted language, since the needed multisets would be just another type in addition to real values and formulas.

Extending the axiomatization to one for games for first-order fuzzy logic on the other hand cannot be done in a elegant way in our current first-order modal formalism and would profit from a stronger typed language.

5.2.4 Implementing the Axiomatization in a Theorem Prover

One extension to the work presented here of particular interest to the author is the implementation of the axiomatization in chapter 4 in a appropriate tool for automated reasoning.

Since the start of this thesis the list of suitable tools has not only increased, additionally some of the projects have matured to the state of being considered software packages rather than a large playing field for experimentation and research[5]. Six years ago investigating possibilities to implement the axiomatization presented here would have merited a master's thesis by itself. A short investigation carried out now suggests that a straight-forward implementation could be achieved with significantly less effort.

Our preliminary research shows that modal logic has been implemented in various ways in at least two renowned proof assistants: Coq[6] [BWP15b, DS11] and Isabelle[7] [BR13, BWP15a].

The approach of formalizing first-order and higher-order modal logic, by translating it to higher-order classical logic in Coq as investigated by Benzmüller and Woltzenlogel Paleo [BWP15b] seems particularly promising as a starting point.

[5] We were particularly surprised of reading about proving theorems in Coq as a hobby on a well-known CS and IT news-site: https://www.stephanboyer.com/post/134/my-unusual-hobby.

[6] https://coq.inria.fr/

[7] https://isabelle.in.tum.de/

5.2.5 Decidability Problems

Our axiomatization deals with finite models only, since the propositional formulas are of finite length, and clearly evaluation of propositional formulas of the two logics is a decidable problem. However, relating to automatically deducing theorems from our modeling we should consider the expressiveness of the underlying logic formalism.

Using a normal multi-modal first-order logic for the axiomatization provides us with an elegant way of expressing all necessary aspects of evaluation game trees. The elegance in expression comes at the cost of decidability, because most first-order modal logics are undecidable [WZ01, Ham16], [BvBW07, chapter 9][8].

Most decidable fragments of first-order classical logic, are not decidable in the modal setting: the monadic fragment, and the fragment using only two distinct variables of most modal logics is undecidable. The single variable fragment of some modal logics is decidable[9] [WZ01].

Both axiomatizations in chapter 4 belong to fragments which are undecidable: both use two variables, and only the axiomatization of the game for classical logic contains no binary predicates.

Some undecidability results involve an interplay of modalities with free variables in a fundamental way. The article by Wolter and Zakharyaschev [WZ01] shows that restricting the language in a way, that a modal operator has at most one variable in its scope, seems to improve the situation. Our axiomatization of $\mathcal{H}$-games belongs to this category, however the article analyzes logic without equality or function symbols.

[8] Even logics like quantified S5, that has the same complexity as classical logic in the propositional case.

[9] It is also comparatively weak from a expressibility point of view - quantifiers in that case have the expressive power of propositional variables and an S45 modality

APPENDIX A

Notation used

$G = \langle Pl, H, Tu, \{\succeq_i\}_{i \in Pl} \rangle$

extensive game of perfect information G, with Pl — the set of players, H as a set of histories (h is an individual history, consisting of a sequence of actions $a \in A$), $Z \subseteq H$ are the terminal histories and Tu the turn-function Definition 2.4.2. $\succeq_i$ are the preference relations for each player, $u(h)$ is a payoff function used in place of the preference relations.

σ_i, τ_i

strategies of player i in an extensive game σ denotes strategy profiles. $o(\sigma)$ denotes the history resulting from a strategy profile.

$\varphi, \psi, \ldots$

arbitrary formulas.

$\mathcal{Prop}$

The set of all syntactically valid propositional formulas.

$\mathcal{Atom}$

The set of all syntactically valid propositional atomic formulas: propositional variables $p, q, p_1, \ldots$ and the two constants $\top, \bot$.

ν_{CL}, ν_{KZ}

valuation for atomic formulas in classical and KZ logic respectively

ν^*_{CL}, ν^*_{KZ}

valuation for arbitrary formulas in classical and KZ logic respectively

$K = \langle W, \{R_1, R_2, \ldots\}, D, V \rangle$

constant domain, modal first-order Kripke-model K with modalities $\{1, 2, \ldots\}$, domain D and rigid interpretation V Definition 2.3.3

I, J

first-order variable valuations Definition 2.1.9

TV

Term valuation function Definition 2.1.11.

$G_{\varphi,\nu_{CL},d}, G_{\varphi,\nu_{KZ},d}$

$\mathcal{H}$-game tree for φ over valuation ν_{CL} or $\mathcal{H}$-mv-game tree for φ over valuation ν_{KZ}, with d as the initial defender Definition 3.2.1.

$\Gamma, \Delta, \ldots$

arbitrary formulas used in chapter 4 to distinguish them from elements of the domain.

$\ulcorner\varphi\urcorner, \ulcorner\psi\urcorner, \ldots$

terms representing formulas in the axiomatization in chapter 4

$l_t : N \to L$ / $l_t : E \to L$

labeling function for the label-type t with labels $\in L$ Definition 2.4.6

List of Figures

Bibliography

[AGM09] Stefano Aguzzoli, Brunella Gerla, and Vincenzo Marra, *Algebras of Fuzzy Sets in Logics Based on Continuous Triangular Norms*, Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 10th European Conference, ECSQARU 2009, Verona, Italy, July 1-3, 2009. Proceedings (Claudio Sossai and Gaetano Chemello, eds.), Lecture Notes in Computer Science, vol. 5590, Springer, 2009, pp. 875–886.

[Bal17] Roberta Ballarin, *Modern Origins of Modal Logic*, The Stanford Encyclopedia of Philosophy (Edward N. Zalta, ed.), Metaphysics Research Lab, Stanford University, summer 2017 ed., 2017.

[Bin92] Ken Binmore, *Fun and Games*, Heath, Lexington, MA, 1992.

[Bon02] Giacomo Bonanno, *Modal logic and game theory: two alternative approaches*, Risk, Decision and Policy **7** (2002), no. 3, 309–324.

[BR13] Christoph Benzmüller and Thomas Raths, *HOL based First-order Modal Logic Provers*, Proceedings of the 19th International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR) (Stellenbosch, South Africa) (Kenneth L. McMillan, Aart Middeldorp, and Andrei Voronkov, eds.), LNCS, vol. 8312, Springer, 2013, pp. 127–136.

[BvBW07] Patrick Blackburn, Johan van Benthem, and Frank Wolter, *Handbook of Modal Logic, Volume 3 (Studies in Logic and Practical Reasoning)*, Elsevier Science Inc., New York, NY, USA, 2007.

[BWP15a] Christoph Benzmüller and Bruno Woltzenlogel Paleo, *Higher-Order Modal Logics: Automation and Applications*, Reasoning Web 2015 (Berlin, Germany) (Adrian Paschke and Wolfgang Faber, eds.), LNCS, no. 9203, Springer, 2015, (Invited paper), pp. 32–74.

[BWP15b] Christoph Benzmüller and Bruno Woltzenlogel Paleo, *Interacting with Modal Logics in the Coq Proof assistant*, Computer Science - Theory and Applications - 10th International Computer Science Symposium in Russia, CSR 2015, Listvyanka, Russia, July 13-17, 2015, Proceedings (Lev D. Beklemishev and Daniil V. Musatov, eds.), LNCS, vol. 9139, Springer, 2015, pp. 398–411.

[CFN16] Petr Cintula, Christian G. Fermüller, and Carles Noguera, *Handbook of mathematical fuzzy logic - volume 3*, College Publications, 2016.

[CFN17] Petr Cintula, Christian G. Fermüller, and Carles Noguera, *Fuzzy Logic*, The Stanford Encyclopedia of Philosophy (Edward N. Zalta, ed.), Metaphysics Research Lab, Stanford University, fall 2017 ed., 2017, Available at https://plato.stanford.edu/archives/fall2017/entries/logic-fuzzy/.

[CHN11] Petr Cintula, Petr Hájek, and Carles Noguera (eds.), *Introduction to Mathematical Fuzzy Logic*, ch. 1, College Publications, 2011.

[DS11] Christian Doczkal and Gert Smolka, *Constructive Formalization of Hybrid Logic with Eventualities*, Certified Programs and Proofs (Berlin, Heidelberg) (Jean-Pierre Jouannaud and Zhong Shao, eds.), Springer Berlin Heidelberg, 2011, pp. 5–20.

[Dum59] Michael Dummett, *A Propositional Calculus with Denumerable Matrix*, Journal of Symbolic Logic **24** (1959), no. 2, 97–106.

[Fer14] Christian G. Fermüller, *Hintikka-Style Semantic Games for Fuzzy Logics*, Foundations of Information and Knowledge Systems - 8th International Symposium, FoIKS 2014, Bordeaux, France, March 3-7, 2014. Proceedings (Christoph Beierle and Carlo Meghini, eds.), Lecture Notes in Computer Science, vol. 8367, Springer, 2014, pp. 193–210.

[Fer16] Christian G. Fermüller, *On Semantic Games for Fuzzy Logics.*, ch. XIII, pp. 969–1038, in [CFN16], 2016.

[FM98] Melvin Fitting and Richard L. Mendelsohn, *First-Order Modal Logic*, Kluwer Academic Press, 1998.

[FM09] Christian G. Fermüller and George Metcalfe, *Giles's Game and the Proof Theory of Łukasiewicz Logic*, Studia Logica **92** (2009), 27–61.

[FM15] Christian G. Fermüller and Ondrej Majer, *Equilibrium Semantics for IF Logic and Many-Valued Connectives*, Logic, Language, and Computation - 11th International Tbilisi Symposium, TbiLLC 2015, Tbilisi, Georgia, September 21-26, 2015, Revised Selected Papers (Helle Hvid Hansen, Sarah E. Murray, Mehrnoosh Sadrzadeh, and Henk Zeevat, eds.), Lecture Notes in Computer Science, vol. 10148, Springer, 2015, pp. 290–312.

[FR12] Christian G. Fermüller and Christoph Roschger, *Randomized Game Semantics for Semi-fuzzy Quantifiers*, IPMU (4) (Salvatore Greco, Bernadette Bouchon-Meunier, Giulianella Coletti, Mario Fedrizzi, Benedetto Matarazzo, and Ronald R. Yager, eds.), Communications in Computer and Information Science, vol. 300, Springer, 2012, pp. 632–641.

[Gar16] James Garson, *Modal Logic*, The Stanford Encyclopedia of Philosophy (Edward N. Zalta, ed.), Metaphysics Research Lab, Stanford University, spring 2016 ed., 2016.

[Gil77] Robin Giles, *A non-classical Logic for Physics*, Selected Papers on Łukasiewicz sentential calculi (Ryszard Wójcicki and Malinowski Grzegorz, eds.), Polish Academy of Sciences, 1977, pp. 10–51.

[Göd32] Kurt Gödel, *Zum Intuitionistischen Aussagenkalkül*, Anzeiger der Akademie der Wissenschaften in Wien **5** (1932), 65–66.

[Gol97] Martin Goldstern, *The Complexity of Fuzzy Logic*, Available at https://arxiv.org/abs/math/9707205, July 1997.

[Gol13] ______, *The Complexity of Łukasiewicz Logic*, 43rd IEEE International Symposium on Multiple-Valued Logic, ISMVL 2013, Toyama, Japan, May 22-24, 2013, IEEE Computer Society, 2013, pp. 176–181.

[Háj98] Petr Hájek, *Metamathematics of Fuzzy Logic*, Trends in Logic, Kluwer, 1998.

[Háj11] ______, *Fuzzy Logic*, The Stanford Encyclopedia of Philosophy (Edward N. Zalta, ed.), Metaphysics Research Lab, Stanford University, summer 2011 ed., 2011, Available at http://plato.stanford.edu/archives/fall2010/entries/logic-fuzzy/.

[Ham16] Christopher Hampson, *Decidable first-order modal logics with counting quantifiers*, Advances in Modal Logic 11, proceedings of the 11th conference on "Advances in Modal Logic," held in Budapest, Hungary, August 30 - September 2, 2016 (Lev D. Beklemishev, Stéphane Demri, and András Maté, eds.), College Publications, 2016, pp. 382–400.

[HGE96] Petr Hájek, Lluis Godo, and Francesc Esteva, *A complete many-valued logic with product-conjunction*, Archive for Mathematical Logic **35** (1996), no. 3, 191–208.

[HGE13] ______, *Fuzzy Logic and Probability*, CoRR **abs/1302.4953** (2013), 237–244, Available at http://arxiv.org/abs/1302.4953.

[Hin73a] Jaakko Hintikka, *Language-Games for Quantifiers*, pp. 53–82, in [Hin73b], 1973.

[Hin73b] ______, *Logic, Language-Games and Information*, Oxford University Press, Oxford, UK, 1973.

[Hin73c] ______, *Quantifiers, Language-Games and Transcendental Arguments*, pp. 98–122, in [Hin73b], 1973.

[Hin82] ______, *Game-Theoretical Semantics: Insights and Prospects*, Notre Dame Journal of Formal Logic **23** (1982), no. 2, 219–241.

[HMvdHW03] Paul Harrenstein, John-Jules Meyer, Wiebe van der Hoek, and Cees Witteveen, *A Modal Characterization of Nash Equilibrium*, Fundamenta Informaticae **57** (2003), no. 2–4, 281–321.

[Hod13] Wilfrid Hodges, *Logic and games*, The Stanford Encyclopedia of Philosophy (Edward N. Zalta, ed.), Metaphysics Research Lab, Stanford University, spring 2013 ed., 2013.

[HR04] Michael Huth and Mark Ryan, *Logic in Computer Science: Modelling and Reasoning About Systems*, Cambridge University Press, New York, NY, USA, 2004.

[Kei11] Laurent Keiff, *Dialogical Logic*, The Stanford Encyclopedia of Philosophy (Edward N. Zalta, ed.), Metaphysics Research Lab, Stanford University, summer 2011 ed., 2011, Available at http://plato.stanford.edu/archives/sum2011/entries/logic-dialogical/.

[KMP00] Erich Peter Klement, Radko Mesiar, and Endre Pap, *Triangular Norms*, 1 ed., Springer, 2000.

[Kri63] Saul Kripke, *Semantical Analysis of Modal Logic I Normal Modal Propositional Calculi*, Mathematical Logic Quarterly **9** (1963), no. 5-6, 67–96.

[Kri80] ______, *Naming and Necessity*, Harvard University Press, Cambridge, Massachusetts, USA, 1980.

[ŁB70] Jan Łukasiewicz and Ludwik Borkowski, *Jan Łukasiewicz Selected Works*, North-Holland Publishing Company, 1970.

[Lei97] Alexander Leitsch, *The Resolution Calculus*, Springer-Verlag New York, Inc., New York, NY, USA, 1997.

[Łuk20] Jan Łukasiewicz, *O logice trójwartościowej*, Ruch Filozoficzny (1920), no. 5, 170–171, English translation in [ŁB70].

[Man93] María Manzano, *Introduction to Many-sorted Logic*, Many-sorted Logic and Its Applications (K. Meinke and J. V. Tucker, eds.), John Wiley & Sons, Inc., New York, NY, USA, 1993, pp. 3–86.

[MC09] Ondrej Majer and Petr Cintula, *Towards Evaluation Games for Fuzzy Logics*, Games: Unifying Logic, Language, and Philosophy (Ondrej Majer, Ahti-Veikko Pietarinen, and Tero Tulenheimo, eds.), Logic, Epistemology, and the Unity of Science, Springer, 2009, pp. 117–131.

[Nov87] Vilém Novák, *First-Order Fuzzy Logic*, Studia Logica **46** (1987), no. 1, 87–109.

[NW05] Hung T. Nguyen and Elbert A. Walker, *A First Course in Fuzzy Logic (3. ed.)*, Chapman&Hall/CRC Press, 2005.

[OR94] Martin J. Osborne and Ariel Rubinstein, *A Course in Game Theory*, The MIT Press, July 1994.

[Par85] Rohit Parikh, *The Logic of Games and Its Applications*, Selected Papers of the International Conference on "Foundations of Computation Theory" on Topics in the Theory of Computation (New York, NY, USA), Elsevier North-Holland, Inc., 1985, pp. 111–139.

[PP03] Marc Pauly and Rohit Parikh, *Game Logic - An Overview*, Studia Logica **75** (2003), no. 2, 165–182.

[Sha13] Stewart Shapiro, *Classical Logic*, The Stanford Encyclopedia of Philosophy (Edward N. Zalta, ed.), Metaphysics Research Lab, Stanford University, winter 2013 ed., 2013, Available at http://plato.stanford.edu/archives/win2013/entries/logic-classical/.

[vB83] Johan van Benthem, *Modal Logic and Classical Logic*, Bibliopolis, Naples, Italy, 1983.

[vB01] ______, *Correspondence Theory*, Handbook of Philosophical Logic, Springer, 2001, pp. 325–408.

[vB02] ______, *Extensive Games as Process Models*, Journal of Logic, Language and Information **11** (2002), 289–313.

[vB03] ______, *Logic Games are Complete for Game Logics*, Studia Logica **75** (2003), no. 2, 183–203, Also available at http://staff.science.uva.nl/~johan/GL=LG.pdf.

[WZ01] Frank Wolter and Michael Zakharyaschev, *Decidable fragments of first-order modal logics*, The Journal of Symbolic Logic **66** (2001), no. 3, 1415–1438.

[Zad65] Lotfi A. Zadeh, *Fuzzy Sets*, Information and Control **8** (1965), no. 3, 338–353.

[Zad94] ______, *The role of fuzzy logic in modeling, identification and control*, Modeling, Identification and Control **15** (1994), no. 3, 191–203.

Index