

Faculty of Science and Bio-Engineering Sciences
Department of Computer Science



Arenberg Doctoral School
Faculty of Engineering Science
Department of Computer Science

Fixpoint Theory and Biworlds

The Power of Approximations in Knowledge Representation

Dissertation submitted in fulfilment of the requirements for the joint degree of

Doctor of Sciences and Doctor of Engineering Science (PhD): Computer Science

Samuele Pollaci

Brussels, October 2026

Promotors: Prof. Dr. Bart Bogaerts (Vrije Universiteit Brussel & KU Leuven)
Prof. Em. Dr. Marc Denecker (KU Leuven)

Alle rechten voorbehouden. Niets van deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotokopie, microfilm, elektronisch of op welke andere wijze ook, zonder voorafgaande schriftelijke toestemming van de auteur.

All rights reserved. No part of this publication may be produced in any form by print, photoprint, microfilm, electronic or any other means without permission from the author.

Use of Generative AI: I did not use generative AI assistance tools during the research/writing process of my thesis, except for mere language assistance. The text/code/images in this thesis are my own (unless otherwise specified) and generative AI has only been used in accordance with the KU Leuven guidelines and appropriate references have been addded. I have reviewed and edited the content as needed and I take full responsiility for the content of the thesis.

Printed by
Crazy Copy Center Productions
VUB Pleinlaan 2, 1050 Brussel
Tel : +32 2 629 33 44
crazycopy@vub.be
www.crazycopy.be

ISBN : 9789493461864
NUR CODE : 984
THEMA : UYQ

Members of the Jury

Prof. dr. Geraint Wiggins	Vrije Universiteit Brussel, BE Queen Mary University of London, UK (Chair)
Prof. dr. Bart Bogaerts	Vrije Universiteit Brussel, BE KU Leuven, BE (Promotor)
Prof. Em. dr. Marc Denecker	KU Leuven, BE (Promotor)
Prof. dr. Pieter Bonte	KU Leuven, BE
Prof. dr. Dominique Devriese	KU Leuven, BE
Prof. dr. Lynn Houthuys	Vrije Universiteit Brussel, BE
Prof. dr. Yanhong Annie Liu	Stony Brook University, US
Prof. dr. Panos Rondogiannis	University of Athens, GR
Prof. dr. Joost Vennekens	Vrije Universiteit Brussel, BE

Summary

Approximations arise in almost every scientific field and application: either by design, to allow for certain constructions or to make an otherwise too complicated problem more tractable; or by lack of information, like in the case of predictions, and empirical measurements. Thus, formally representing and being able to use this kind of incomplete information is fundamental. In the context of Knowledge Representation and Reasoning, this is achieved through various devices, like partially defined functions and objects, or three- and four-valued semantics.

In this thesis, we explore and showcase the capabilities of approximations by employing them to define different semantics through two main formalisms: first, via the machinery of Approximation Fixpoint Theory (AFT); secondly, in the context of multi-agent autoepistemic logic (AEL), using the possible worlds of a Kripke structure obtained from the novel concept of biworlds. Even though the two settings are rather different, how we use approximations shares some similarities: in both cases, we encode an approximation as a pair of bounds (either a couple of elements from a lattice, or a biworld), and we define a precision order on the set of such approximations. Then, the versatility and relative simplicity of the approximations allow for the successful construction of complex objects, which are fundamental to defining the desired semantics.

In the first part of this thesis, we focus on the AFT framework. AFT turned out to be an extremely powerful algebraic tool to unify the semantics of different non-monotonic formalisms, like logic programming, default logic, and autoepistemic logic, with applications ranging from abstract argumentation to active integrity constraints and stream reasoning, among others.

We start by providing yet another application of AFT, namely DatalogMTL$^\neg$: an extension of Datalog with metric temporal operators and enriched with unstratifiable negation. By using the solid mathematical formalism of AFT, we are able to define the stable, well-founded, Kripke-Kleene, and supported model semantics for DatalogMTL$^\neg$ in a straightforward way. Moreover, we prove that the stable model semantics obtained via AFT coincides with the one defined in previous work, which employed pairs of interpretations stemming from the logic of Here-and-There.

We then move our attention to the generalization of AFT. Despite its success, AFT is not readily applicable to higher-order definitions. To solve such an issue, we devise a formal mathematical framework employing concepts drawn from Category Theory. In

particular, we make use of the notion of Cartesian closed category to inductively construct higher-order approximation spaces while preserving the structures necessary for the correct application of AFT. We show that this novel theoretical approach successfully extends various versions of AFT to a higher-order environment.

In the second part of this thesis, we are concerned with multi-agent AEL. In particular, we study the interplay of the concepts of common knowledge and only knowing: they capture two intuitive and natural notions that have proven to be useful in a variety of settings, for example, to reason about coordination or agreement between agents, or to analyse the knowledge of knowledge-based agents. While these two epistemic operators have been extensively studied in isolation, the approaches made to encode their complex interplay failed to capture some essential properties of only knowing. We propose a novel solution by defining a notion of μ-biworld for countable ordinals μ, which approximates not only the worlds that an agent deems possible, but also those deemed impossible. In a way, this picks up the idea of lower and upper bounds that is predominant in AFT. This approach allows us to define a multi-agent epistemic logic with common knowledge and only knowing operators, and a three-valued model semantics for it. Moreover, we show that we only really need biworlds of depth at most $\omega^2 + 1$. Based on this observation, we define a Kripke semantics on a canonical Kripke structure and show that this semantics coincides with the model semantics. Finally, we discuss issues arising when combining negative introspection or truthfulness with only knowing, and show how positive introspection can be integrated into our logic.

Samenvatting

Benaderingen komen in vrijwel elk wetenschappelijk vakgebied en elke toepassing voor: ofwel opzettelijk, om bepaalde constructies mogelijk te maken of om een anders te complex probleem beter hanteerbaar te maken; ofwel door een gebrek aan informatie, zoals in het geval van voorspellingen en empirische metingen. Het formeel representeren en kunnen gebruiken van dit soort onvolledige informatie is daarom fundamenteel. In de context van kennisrepresentatie en redenering wordt dit bereikt door middel van verschillende methoden, zoals gedeeltelijk gedefinieerde functies en objecten, of drie- en vierwaardige semantiek.

In dit proefschrift onderzoeken en demonstreren we de mogelijkheden van benaderingen door ze te gebruiken om verschillende semantieken te definiëren via twee belangrijke formalismen: ten eerste via de machinerie van de Approximation Fixpoint Theory (AFT); ten tweede, in de context van multi-agent autoepistemic logic (AEL), gebruikmakend van de mogelijke werelden van een Kripke-structuur die is afgeleid van het nieuwe concept van biwerelden. Hoewel de twee contexten nogal verschillend zijn, vertonen de manieren waarop we benaderingen gebruiken enkele overeenkomsten: in beide gevallen coderen we een benadering als een paar grenzen (ofwel een paar elementen uit een rooster, ofwel een biwereld), en definiëren we een precisieordening op de verzameling van dergelijke benaderingen. Vervolgens maken de veelzijdigheid en relatieve eenvoud van de benaderingen de succesvolle constructie van complexe objecten mogelijk, die essentieel zijn voor het definiëren van de gewenste semantiek.

In het eerste deel van dit proefschrift richten we ons op het AFT-raamwerk. AFT bleek een uiterst krachtig algebraïsch hulpmiddel te zijn om de semantiek van verschillende niet-monotone formalismen, zoals logisch programmeren, “default” logica, en auto-epistemische logica, te verenigen, met toepassingen variërend van abstracte argumentatie tot actieve integriteitsbeperkingen en streamredenering, om er maar een paar te noemen.

We beginnen met een nieuwe toepassing van AFT, namelijk DatalogMTL$^\neg$: een uitbreiding van Datalog met metrische temporele operatoren en verrijkt met niet-stratificeerbare negatie. Door gebruik te maken van het solide wiskundige formalisme van AFT, kunnen we op een eenvoudige manier de stable, well-founded, Kripke-Kleene en supported modelsemantiek voor DatalogMTL$^\neg$ definiëren. Bovendien bewijzen we dat de stabiele modelsemantiek die via AFT wordt verkregen, samenvalt met die gedefinieerd in eerder werk, waarin paren van interpretaties werden gebruikt die voortkomen uit de logica van “Here-and-There”.

Vervolgens richten we onze aandacht op de generalisatie van AFT. Ondanks het succes is AFT niet direct toepasbaar op definities van hogere orde. Om dit probleem op te lossen, ontwikkelen we een formeel wiskundig raamwerk met concepten uit de categorietheorie. In het bijzonder maken we gebruik van het begrip Cartesiaanse gesloten categorie om inductief hogere-orde benaderingsruimten te construeren, terwijl we de structuren behouden die nodig zijn voor de correcte toepassing van AFT. We laten zien dat deze nieuwe theoretische benadering verschillende versies van AFT succesvol uitbreidt naar een hogere-orde omgeving.

In het tweede deel van dit proefschrift houden we ons bezig met multi-agent AEL. In het bijzonder bestuderen we de wisselwerking tussen de concepten van "common knowledge" en "only knowing": dit zijn twee intuïtieve en natuurlijke begrippen die nuttig zijn gebleken in diverse contexten, bijvoorbeeld om te redeneren over coördinatie of overeenstemming tussen agenten, of om de kennis van kennisgebaseerde agenten te analyseren. Hoewel deze twee epistemische operatoren uitgebreid afzonderlijk zijn bestudeerd, slaagden de benaderingen die werden gebruikt om hun complexe wisselwerking te coderen er niet in om enkele essentiële eigenschappen van *only knowing* te vatten. We stellen een nieuwe oplossing voor door een begrip van μ-biworld te definiëren voor telbare ordinalen μ, dat niet alleen de werelden benadert die een agent mogelijk acht, maar ook die welke hij onmogelijk acht. In zekere zin sluit dit aan bij het idee van onder- en bovengrenzen dat dominant is in AFT. Deze aanpak stelt ons in staat om een multi-agent epistemische logica te definiëren met *common knowledge* en *only knowing* operatoren, en een driewaardige modelsemantiek ervoor. Bovendien laten we zien dat we eigenlijk alleen biwerelden nodig hebben met een diepte van maximaal ω^2+1. Op basis van deze observatie definiëren we een Kripke-semantiek op een canonieke Kripke-structuur en laten we zien dat deze semantiek samenvalt met de modelsemantiek. Ten slotte bespreken we de problemen die ontstaan bij het combineren van negatieve introspectie of waarachtigheid met *only knowing*, en laten we zien hoe positieve introspectie in onze logica kan worden geïntegreerd.

Preface

During the past four years, I have worked on several projects concerning a variety of research topics. In this thesis, I chose to collect a few that represent, albeit only partly, my PhD. Nevertheless, a PhD journey is far more than its research output. It is four years of studying, learning, exchanging ideas and points of view with challenging and interesting conversations. It is the joy and pride of each discovery and achievement, and the frustration of each failure and dead-end. When I started my PhD, I did not think I would have experienced and grown so much, both as a researcher and personally. I have taught courses, prepared and corrected exams, written papers, designed posters, and I had the chance to attend conferences all around the world to present my and my collaborators' work and meet fellow researchers from various fields and backgrounds; we shared opinions, ideas, and unforgettable moments in the most disparate places. All of this would not have been possible without the support and care of the people around me.

My most profound and heartfelt thanks go to my two supervisors: Bart at VUB[1], and Marc at KU Leuven. When I started my PhD in 2022, Bart was my only supervisor. He has been a thoughtful and extremely knowledgeable mentor from the very beginning, providing precious insights and valuable suggestions when needed most. I would not have known where to start if not for his guidance. To Bart, thank you for allowing me to pursue my PhD, for believing in me and motivating me, for your always prompt replies, for your witty sense of humour, and for the board games and pizza evenings at the AI Lab.
I started collaborating with Marc a few months into my PhD, and he became my supervisor at the beginning of 2024, when the joint nature of my PhD was made official. Marc appears to have a Pandora's box of ideas, questions, drafts, and old work to finish coming from years of ongoing research, and he is generous enough to share this never-ending source of information. To Marc, thank you for all the inspiring conversations, for challenging my views, for the talks in the cafeteria, and for the lengthy and always interesting emails. I truly believe having a good PhD supervisor is half the battle; imagine having two! Bart, Marc, I am immensely grateful to you.

I thank all members of my jury for taking the time to read this thesis carefully and for being present at my defences. I am sincerely grateful for the interest you have shown, the challenging questions, and stimulating remarks. For similar reasons, I would like to

[1]Ironically, Bart moved to KU Leuven halfway through my PhD, turning it into a joint PhD with both supervisors at the partner university and none at the home institution.

acknowledge Prof. Em. dr. Maurice Bruynooghe, who read an earlier version of this thesis and provided precise, sensible feedback on each chapter, contributing to perfecting them.

I would like to thank all my VUB and KU Leuven colleagues, all my collaborators, and all the people from the Knowledge Representation and Reasoning community (and beyond) I was lucky enough to cross paths with in the last four years. I am amazed by how welcoming and kind-hearted each of these people is, and how I found not only research partners, but also some fantastic friends. Since thanking each of you would be a never-ending task (and I would probably miss someone), I refrain from doing so. Nevertheless, I owe a few special mentions.
To Çagri and Dieter, my first office mates at the VUB, thank you for your nice company, all the tea drunk, and all the meals shared. You constituted a valid reason to get to the office early enough (you know I am far from being a morning person).
To Simon, who became one of my first and dearest friends in Brussels, thank you for your kindness, for the deep conversations, for all the parties, garden fairs, walks and brocantes. You have been there for me since (almost) the beginning of my life chapter in Brussels, and you made me feel less lonely.
To Linde, my academic sister from KU Leuven, thank you for the endless talks, the laughter, and the support (and your mini freezer in the office). You are not only a valuable collaborator but also, and most importantly, a great friend: I am glad I got to share most of my PhD journey, with its joys and frustrations, with you.
To Robbe, one of my office mates at KU Leuven and academic sibling, thank you for all your dad jokes, your light spirit, and all the famous Alma burgers savored together. I have truly enjoyed collaborating and sharing my Tuesdays in Leuven with you.
To my Greek friends and collaborators, Angelos, Babis, Giannos, and Panos, thank you for warmly welcoming me in Athens for a four-month research visit. I will not easily forget the amazing Greek food and the trips to the beach (I am still dreaming about Loutsa).
I also thank all the other researchers with whom I collaborated on some publications: Marco Calautti, Youri Coppens, Marcos Cramer, Jiahong Li, Andreas Pieris, Willem Röpke, and Bram Vandenbogaerde.

I am deeply grateful to all the amazing friends I made in Brussels who have contributed to making these last years memorable. A special thanks goes to my little Italian gang in Belgium for all the fun, shared experiences, countless dinners, profound and silly conversations, and unconditional love. To all my past and present housemates, thank you for allowing me to make the coloc in Cervantes my home, for all the simple-life moments, the birthday parties and the movie nights, the summer barbecues and the winter raclettes.

Finally, I would like to thank my family and long-standing friends, who have always supported me from afar. Living in different countries is something I believe I will never get used to. Nevertheless, every time we manage to end up in the same place (in Italy or in any other corner of the world), it seems like we have never been apart. It is rather difficult

for me to put into words how much each one of them means to me, and it feels already too sappy for my taste, so I will keep it short. Thank you for all the messages and video calls; thank you for helping me escape the Belgian winter by joining me in the Canaries; thank you for the shared Christmas holidays, for the "party glaciali", and for the restoring summer retreats in our lovely Zena. I would not be the person I am today without you.

List of Acronyms

AEL Autoepistemic Logic

AFT Approximation Fixpoint Theory

AI Artificial Intelligence

cpo chain-complete poset

CT Category Theory

DL Default Logic

FO First-Order logic

glb greatest lower bound

HT Here-and-There

IGP Interlattice Greatest lower bound Property

ILP Interlattice Least upper bound Property

KR Knowledge Representation

KRR Knowledge Representation and Reasoning

LP Logic Programming

lub least upper bound

MEL Metric Equilibrium Logic

MKNF Minimal Knowledge and Negation as Failure

MTL Metric Temporal Logic

NMR Non-Monotonic Reasoning

OT Order Theory

PI positively introspective

poset partially ordered set

List of Symbols

$\mathcal{P}$	A poset, page 16
$\langle P, \leq \rangle$	A poset with underlying set P and order $\leq$, page 16
$\mathcal{P}^{op}$	The opposite poset of $\mathcal{P}$, page 16
2^S	The power set of S, page 16
$\mathbb{N}^*$	The set of natural numbers without 0, page 16
$\mathrm{glb}_P(S)$	The greatest lower bound of S in P, page 18
$\mathrm{glb}(S)$	The greatest lower bound of S (implicit environment $P \supseteq S$), page 18
$\mathrm{lub}_P(S)$	The least upper bound of S in P, page 18
$\mathrm{lub}(S)$	The least upper bound of S (implicit environment $P \supseteq S$), page 18
$\bot_S$	The bottom of S, page 18
$\mathcal{D}$	A diagram, page 66
$\mathbf{O}$	The empty category, page 67
$\mathbb{H}$	A hierarchy of types, page 68
$\mathfrak{L}$	A higher-order language, page 68
τ	A type, page 68
$\Pi_{i \in I} \tau_i$	The product type of $\{\tau_i \mid i \in I\}$, page 68
$\tau_1 \to \tau_2$	The morphism type of τ_1 and τ_2, page 68
o	The boolean type, page 68
ι	The type of individuals, page 68
$\mathcal{B}_{\mathbb{H}}$	The set of base types of $\mathbb{H}$, page 68
$\top_S$	The top of S, page 18

E_τ	The semantics of τ, page 68
V	A vocabulary, page 69
s	A symbol of a vocabulary, page 69
$\tau(s)$	The type of s, page 69
$\mathbb{K}$	A subset of a type hierarchy, page 69
$S_\mathbb{K}$	The set of types obtained from $\mathbb{K}$, page 69
$\mathcal{B}_\mathbb{K}$	The set of base types of $\mathbb{H}$ belonging to $\mathbb{K}$, page 69
$\mathrm{App}(E_\tau)$	The approximation space of E_τ given App, page 69
$f^{-1}(y)$	The preimage of y via f, page 70
Approx	An approximation category, page 71
$\mathrm{glb}_\leq(S)$	The greatest lower bound of S w.r.t. $\leq$, page 18
App	An approximating function, page 71
$\mathfrak{p}^0_\tau$	A representation function for the base type τ, page 71
$\mathcal{E}_\tau$	The set of exact elements of type τ, page 72
$\mathfrak{p}_\tau$	A representation function for τ, page 76
SqLat	The category of square lattices, page 77
$\overline{\mathcal{L}}$	The square lattice $\mathcal{L} \times \mathcal{L}^{op}$, page 77
$L \otimes U$	The set $\{(x, y) \mid x \in L, y \in U, x \leq y\}$, page 81
$(L, U \leq)$,	An approximating tuple, page 81
LUcons	The approximation category for higher-order consistent AFT, page 82
$\{*\}$	A set with one element, page 83
$\mathrm{lub}_\leq(S)$	The least upper bound of S w.r.t. $\leq$, page 18
$\mathcal{HOL}$	The higher-order language defined by [Charalambidis et al., 2018], page 85
σ	A functional type, page 85

π	A predicate type, page 85
ρ	A parameter type, page 85
$\mathcal{H}_{\mathfrak{P}}$	The set of Herbrand interpretations for $\mathfrak{P}$, page 85
$\mathfrak{s}$	A state, page 87
$\mathfrak{s}[\mathsf{R}_1/d_1, \dots, \mathsf{R}_n/d_n]$	The state that assigns d_i to R_i for every $i \in \{1, \dots, n\}$, and coincides with $\mathfrak{s}$ anywhere else, page 87
$\mathcal{E}^{states}$	The set of exact states, page 88
$\mathrm{App}_{24}(E_\tau)$	The approximation space for the elements of type τ as characterized in [Bogaerts et al., 2024], page 101
$\mathcal{T}^\tau$	The set of tight elements of type τ, page 102
Σ	A vocabulary, page 19
T^τ	The tightification function for τ, page 102
$\mathbb{L}^\tau$	The inverse tightification function for τ, page 102
G	A set of agents, page 106
C_G	The common-knowledge operator for G, page 106
$\mathcal{A}$	An agent, page 106
$O_{\mathcal{A}}$	The only-knowing operator for $\mathcal{A}$, page 106
$\mathcal{K}$	A Kripke structure, page 106
W	A set of worlds, page 106
$R_{\mathcal{A}}$	The accessibility relation for $\mathcal{A}$, page 106
$\boldsymbol{\mathcal{A}}$	The set of all agents, page 107
I, J	Two-valued interpretations, page 19
ω	The first limit ordinal, page 108
w	A prebiworld or a biworld, page 109
$\mathcal{A}^w$	The set of (pre)biworlds that can be extended to a (pre)biworld $\mathcal{A}$ deems possible in w , page 109

$\bar{\mathcal{A}}^w$	The set of (pre)biworlds that can be extended to a (pre)biworld $\mathcal{A}$ deems impossible in w , page 109
w^{obj}	The objective world of w, page 110
$\mathcal{W}_p^\mu$	The set of μ-prebiworlds, page 110
$d(w)$	The depth of w , page 110
$(w)_\alpha$	The α-(pre)biworld which is the element in position α in the transfinite sequence represented by a λ-(pre)biworld w., page 110
$w\|_\alpha$	The restriction of w to α, page 111
$\leq_{\mathcal{W}}$	The precision order on prebiworlds, page 111
t	The truth value true, page 19
$\mathcal{W}^\mu$	The set of μ-biworlds, page 112
$\mathcal{A}\!\uparrow\! w$	See Definition 5.7, page 115
$\bar{\mathcal{A}}\!\uparrow\! w$	See Definition 5.7, page 115
R_μ	Restrictability holds at μ , page 119
MoC_μ	Monotonicity of completedness holds at μ , page 119
C_μ	Completability holds at μ , page 119
CC_μ	The completedness condition holds at μ , page 119
$pl(\mu)$	The largest limit ordinal smaller or equal to μ, page 119
$\mathcal{COL}$	The logic of common knowledge and only knowing, page 127
$K_{\mathcal{A}}$	The knowledge operator for $\mathcal{A}$, page 128
f	The truth value false, page 19
$M_{\mathcal{A}}$	The knowing-at-most operator for $\mathcal{A}$, page 128
E_G	The everybody-knows operator for G, page 128
$\varphi^{\mathcal{K},w}$	The standard two-valued valuation of φ in w in $\mathcal{K}$, page 129
φ^w	The three-valued valuation of φ in w , page 130

E_G^k	The nesting of E_G k-times, page 130
$MD(\varphi)$	The modal depth of φ, page 132
$\mathcal{K}^*$	The ω^2+1-completed Kripke structure, page 133
$\mathfrak{U}$	The set of all worlds, page 133
$wR_{\mathcal{A}}w'$	Equivalent to $(w, w') \in R_{\mathcal{A}}$, page 133
$R_{\mathcal{A}}^w$	The set of all worlds $R_{\mathcal{A}}$-accessible from w, page 133
$(P_1 \to P_2)$	The set of all functions from P_1 to P_2, page 20
Γ	A set of formulas, page 136
$\mathcal{K}_{PI}^*$	The Kripke substructure of $\mathcal{K}^*$ where the underlying world set is the set of all recursively PI worlds, page 141
$\mathfrak{U}^{PI}$	The set of all recursively PI worlds, page 141
$R_{\mathcal{A}}^{PI}$	Accessibility relation for $\mathcal{A}$ in $\mathcal{K}_{PI}^*$, page 141
$\models_{PI}$	The entailment relation in $\mathcal{K}_{PI}^*$, page 142
O	An operator, page 20
$\mathrm{gfp}(O)$	The greatest fixpoint of O, page 21
$\mathrm{lfp}(O)$	The least fixpoint of O, page 21
$\mathfrak{P}$	A (possibly higher-order) logic program, page 21
$\varphi, \psi, \ldots$	Formulas, page 21
$T^{\mathfrak{P}}$	The two-valued immediate consequence operator for $\mathfrak{P}$, page 21
$(x_\alpha)_{\alpha \leq \mu}$	The (possibly transfinite) sequence of elements $x_0, x_1, \ldots x_\mu$, page 22
$\mu, \alpha, \beta, \ldots$	Countable ordinals, page 22
λ	A limit ordinal, page 22
$\mathcal{A}_L$	An approximation space for L, page 25
A	An approximator, page 25
$\langle L^2, \leq_{tr}, \leq_p \rangle$	A square bilattice with underlying set L, page 26

$\leq_{tr}$	The truth order, page 26
$\leq_{p}$	The precision order, page 26
$\mathcal{A}_L^{\mathfrak{s}}$	The approximation space for $\langle L, \leq \rangle$ in standard AFT, page 26
$[x, y]$	The closed interval with greatest lower bound x and least upper bound y, page 27
$\mathbf{u}$	The truth value unkown, page 27
$\mathbf{i}$	The truth value inconsistent, page 27
f_i	The ith component of a function f, page 28
$\mathcal{S}_A$	The stable revision operator of A, page 28
$\Psi^{\mathfrak{P}}$	The three- or four-valued immediate consequence operator for $\mathfrak{P}$, page 29
$\mathcal{A}_L^{\mathfrak{c}}$	The approximation space of $\langle L, \leq \rangle$ in consistent AFT, page 30
$\mathcal{U}_O$	The ultimate approximator of O, page 32
$\mathcal{T}$	A timeline, page 36
$\langle \mathbb{Z}, +, \cdot \rangle$	The ring of integers, page 36
$\langle \mathbb{Q}, +, \cdot \rangle$	The field of rationals, page 36
δ	A ($\mathcal{T}$-)interval, page 36
$\mathsf{P}, \mathsf{Q}, \ldots$	Predicates, page 36
$\mathbf{s}$	A tuple of constants and variables, page 36
$\mathsf{P}(\mathbf{s})$	A relational atom, page 36
M	A metric atom, page 36
$\top$	Truth, page 36
$\bot$	Falsity, page 36
$\mathcal{M}$	The set of ground head atoms, page 36
$\mathcal{D}$	A datalogMTL dataset, page 36

$\mathfrak{I}$	The complete lattice of two-valued interpretations (for DatalogMTL), page 37
r	A rule, page 37
$M \leftarrow B$	The rule with head M and body B, page 37
$\mathcal{P}$	A DatalogMTL$^\neg$ program, page 37
$\mathsf{ground}(\mathcal{P})$	The set of all the ground rules that can possibly be obtained from rules in $\mathcal{P}$ by replacing variables with constants, page 37
$[\![\cdot]\!](I,t)$	The two-valued semantics (for I and t) for DatalogMTL$^\neg$, page 38
$\diamondminus_\delta$	The past-oriented diamond operator for δ, page 38
$\diamondplus_\delta$	The future-oriented diamond operator for δ, page 38
$\boxminus_\delta$	The past-oriented square operator for δ, page 38
$\boxplus_\delta$	The future-oriented box operator for δ, page 38
$\mathcal{S}_\delta$	The since operator for δ, page 38
$\mathcal{U}_\delta$	The until operator for δ, page 38
$\boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{J}}$	Three- or four-valued interpretations, page 39
$[\![\cdot]\!](\boldsymbol{\mathcal{I}},t)$	The three-valued semantics (for $\boldsymbol{\mathcal{I}}$ and t) for DatalogMTL$^\neg$, page 40
$\boldsymbol{\mathcal{F}}$	The "time-unraveling" function for head metric atoms, page 41
$T^{\mathcal{D},\mathcal{P}}$	The two-valued immediate consequence operator for $\mathcal{D}$ and $\mathcal{P}$, page 41
$\Psi^{\mathcal{D},\mathcal{P}}$	The three-valued immediate consequence operator for $\mathcal{D}$ and $\mathcal{P}$, page 42
$\mathcal{R}^\bot_\mathcal{P}$	The set of all $\bot$-rules of a $\bot$-program $\mathcal{P}$., page 52
$\mathbf{C}$, $\mathbf{D}$, …	Categories, page 62
$\mathrm{Ob}(\mathbf{C})$	The collection of objects of $\mathbf{C}$, page 62
$\mathrm{Mor}(\mathbf{C})$	The collection of morphisms of $\mathbf{C}$, page 62
$s(f)$	The source or domain of f, page 62

$t(f)$	The target or codomain of f, page 62
id_X	The identity morphism over X, page 62
$g \circ f$	The composite of f and g, page 62
$X \cong Y$	X and Y are isomorphic, page 63
F	A functor, page 63
F_0	The object function of F, page 63
F_1	The morphism function of F, page 63
Set	The category of sets, page 63
$\hom_{\mathbf{C}}(X, Y)$	The set of morphisms of **C** with domain X and codomain Y, page 63
$(X \to Y)$	The set of functions with domain X and codomain Y, page 63
$\mathbf{C} \subseteq \mathbf{D}$	**C** is a full subcategory of **D**, page 63
POSet	The category of posets, page 63
CPO	The category of cpo's, page 63
CJSLat	The category of complete join semilattices, page 63
CLat	The category of complete lattices, page 63
$(X_1 \times X_2, \pi_1, \pi_2)$	The (categorical) product of X_1 and X_2, page 64
π_1	The first projection morphism, page 64
π_2	The second projection morphism, page 64
$(X_2^{X_1}, ev)$	The (categorical) exponential of X_1 and X_2, page 65
ev	The evaluation morphism (of an exponential), page 65
$\{X_i\}_{i \in I}$	A family of objects indexed by I, page 66
$(\Pi_{i \in I} X_i, (\pi_i)_{i \in I})$	The generalized (categorical) product of $\{X_i\}_{i \in I}$, page 66
π_i	The ith projection morphism, page 66

Contents

Members of the Jury i

Summary iii

Samenvatting v

Preface vii

List of Acronyms xi

List of Symbols xiii

Contents xxi

1 Introduction 1

1 Approximations in Knowledge Representation and Reasoning 2

2 Approximation Fixpoint Theory 5

2.1 A Novel Application: DatalogMTL$^{\neg}$ 6

2.2 Higher-Order AFT 7

3 Multi-Agent Autoepistemic Logic 9

4 Reading Guide 13

2 Background 15

1 An Introduction to Order Theory 15

1.1 Operators and Fixpoints 20

2 Approximation Fixpoint Theory 24

2.1 Standard AFT 26

2.2 Consistent AFT 30

3 Fixpoint Semantics for DatalogMTL with Negation **33**
1 Introduction . . . 33
2 Preliminaries . . . 36
3 The Three-Valued Semantics . . . 39
4 Approximation Fixpoint Theory for DatalogMTL$^{\neg}$ . . . 44
5 Comparison with Previous Work . . . 48
6 Conclusion . . . 54

4 Category-Theory for Higher-Order AFT **57**
1 Introduction . . . 57
2 Preliminaries . . . 61
2.1 Consistent AFT for Higher-Order LP . . . 61
2.2 Category Theory . . . 62
3 The Approximation System . . . 68
4 Higher-Order Standard AFT . . . 77
5 Revised Higher-Order Consistent AFT . . . 80
5.1 The Approximation Category . . . 81
5.2 An Approximation System . . . 84
5.3 A New Approximator . . . 86
6 Different Approaches to Higher-Order Consistent AFT for LP . . . 100
7 Conclusion . . . 103

5 Only Knowing and Common Knowledge **105**
1 Introduction . . . 106
2 Construction of Biworlds . . . 109
3 Properties of Biworlds . . . 115
4 The Logic $\mathcal{COL}$ . . . 127
5 Truthfulness and Introspection . . . 140
6 An Open Challenge: Common Only Knowledge . . . 143
7 Related Work . . . 148
8 Conclusion . . . 150

6 Conclusion **153**

1 Contributions 153

2 Future Directions 155

Bibliography **159**

List of Publications **175**

1

Introduction

Approximations arise in countless real-life situations and in almost any field of study. For instance, when some renovation, construction, or infrastructure work is being planned, an estimated price is issued. When unforeseen problems or emergencies occur, a time estimate for the fix is provided. When taking empirical measurements, depending on the precision of the tool, we get a range of possible values. When we predict a future event, from the weather to the financial market, we make forecasts with a degree of uncertainty. When playing poker (or many other multi-player games), a player only knows the cards in their hand and the ones laid on the table: in order to win, they would have to make guesses and approximate, based on the information available, which cards the other players may hold in their hands.

In any of these situations, the lack of complete information, either about a current or future state, or about someone else's knowledge, compels the use of approximations. Nevertheless, approximations are not always a byproduct of limited information: sometimes they are intentional, as it may be advantageous not to retain all the information, but only part of it. Memory and computability power are limited, both in computers and in humans, and using approximations may speed up processes and alleviate the burden of precise, complete information. Consider when one has to do a quick computation on the spot, or when a professional has to provide highly technical information to someone who is not as proficient in the field. In any case, the careful usage of approximations makes the handling of otherwise too complicated problems easier.

Approximations are ubiquitous, and humans are extremely accustomed to them. Hence, it is fundamental to be able to represent, reason about, and use them formally. In this thesis, we explore and showcase the capabilities of approximations in the field of Knowledge Representation and Reasoning (KRR). Before diving deeper into the specific topics and contributions of this thesis, let us provide an overview of some of the fundamental roles approximations take in KRR.

1 Approximations in Knowledge Representation and Reasoning

Constituting one of the core branches of Artificial Intelligence (AI), KRR [Baral, 2003] aims to model, convey, understand, and interpret knowledge formally. It draws ideas from logic, philosophy, and psychology, and its scope ranges over every aspect of human reasoning processes, such as inductions, deductions, abductions, beliefs, and memory, among others. As previously extensively exemplified, approximations permeate human reasoning. Accordingly, KRR found various ways of encoding such partial information and used it in different modes throughout the years. One of the most glaring examples of this is the development of several three-, four-, and many-valued logics [Kleene, 1952; Priest, 1979; Bonzio et al., 2017; Dunn, 1976; Belnap, 1977; Hájek, 1998; Cintula and Hájek, 2006; Marcos et al., 2026]. While in Boolean logic [Boole, 2009] there are only two truth values, namely *true* and *false*, in a three-, four-, or many-valued logic there are three, four, or (possibly infinitely) many truth values, respectively. For instance, in a three-valued logic, a third truth value **u** is added, which is usually meant to represent something "neither true nor false", *unknown* or *undefined*: Kleene [1952] associated **u** to predicates which are "undecidable by the algorithms whether true or false"; Putnam [1957] used **u** for statements that cannot be physically verified or falsified; Prior [1953] and Taylor [1957] used Łukasiewicz three-valued logic for the problem of *future contingents*, associating **u** to statements about a future yet to be determined; de Finetti [1995] regarded **u** as "a doubt between yes or no", linking it with lack of knowledge of an agent.

Interestingly, the value **u** can also be used to indicate the undefined portions of a partially defined concept: for instance, we may regard the function $f(n) := \frac{1}{n}$ as a partial function $f : \mathbb{N} \to \mathbb{N}$ sending 0 to **u** and $n \neq 0$ to $\frac{1}{n}$. Partially defined objects constitute another extremely common example of approximations in KRR: the aforementioned function f can be thought of as an approximation of all total (and partial) functions $g : \mathbb{N} \to \mathbb{N}$ that agree with f on the elements of the domain for which f is defined, i.e., with $g(n) = f(n)$ for all $n \in \mathbb{N} \setminus \{0\}$. The view of partial objects as approximations is of particular relevance in the context of constructive definitions, as they provide an approximation of the object being

defined by only encoding the information available at a specific stage of the construction process.

Constructive Definitions. Constructive definitions, like inductive and recursive ones, are disseminated in almost any branch of mathematics and computer science, and have been extensively studied in both fields [Post, 1943; Spector, 1961; Kreisel, 1963; Feferman, 1970; Martin-Löf, 1971; Moschovakis, 1974; Aczel, 1977; Roberts, 1986; Soare, 1987; Hallnäs, 1991; Denecker and Vennekens, 2007; Vanbesien et al., 2024]. These types of constructions have been integrated and employed in several languages [Kozen, 1983; Calvanese et al., 1999; Denecker and Ternovska, 2008; Afanasiev et al., 2008; Rubio-Sánchez, 2018], and they can be successfully used to define some of the most fundamental objects in the field. For instance, the set $\mathbb{N}$ of natural numbers can be easily defined as the smallest set of real numbers satisfying the following informal rules:

1. $0 \in \mathbb{N}$;
2. If $n \in \mathbb{N}$, then $n + 1 \in \mathbb{N}$.

Considering that the above rules define an infinite set, such a definition is strikingly simple and extremely clear in its essentiality. This is indeed thanks to its constructive nature: the set of natural numbers is not defined directly in its entirety, but rather obtained via the recipe comprising the above simple rules. At each step of the construction, the application of a rule provides a new, stricter *lower bound*, telling which elements must be in the set of natural numbers *at least*: at the first step, at least 0 is in the set (base step); then, by applying rule 2 (induction step), we have that also 1 belongs to the set; and so on. Notice that, for instance, at the third inductive step, we would assign *unknown* to the statement "73 is a natural number", and only later on in the process we can determine whether it is *true* or *false*: throughout the construction process, we obtain a sequence of *approximations* of what $\mathbb{N}$ is, given by the lower bounds. The limit of such a sequence marks the end of this process and coincides with the object being defined. Such an employment of approximations is essential for any constructive definition, and allows the use of clear and relatively simple rules to define, sometimes rather complex, objects, making this type of definitions extremely versatile and widely adopted.

Notice that the induction process we consider above is *monotonic*, and the application of rules is "safe": if we can apply a rule at a certain stage of the construction, then the same rule can be applied at any later stage to derive the same fact. When non-monotonicity comes into play, the situation becomes more interesting.

Non-Monotonic Reasoning Non-Monotonic Reasoning (NMR) is a key subfield of KRR which concerns all those reasoning techniques for which adding more information

may invalidate previously derived statements. This is indeed the case for several types of human reasoning mechanisms: "commonsense reasoning is 'nonmonotonic' in the sense that we often draw, on the basis of partial information, conclusions that we later retract when we are given more complete information", Moore [1985] wrote. For instance, Logic Programming (LP), Default Logic (DL), and Autoepistemic Logic (AEL) are non-monotonic logics. Let us briefly consider the latter, which sits at the core of Chapter 5. There are multiple views on AEL. One view is that it is a logic to express the knowledge of a rational agent who is aware of their own knowledge and ignorance. In this respect, the knowledge base of said agent may contain axioms that reflect on their own knowledge or ignorance, and use them to derive other facts. This is achieved by enriching First-Order logic (FO) with a modal operator K, encoding the knowledge of the agent: statements of the form $K\varphi$ ought to be read as *the agent knows* φ. The non-monotonicity of such a logic should not come as a surprise: if the agent obtains new information resulting in an extension of its knowledge base T, certain statements previously considered true may become false. For instance, suppose that I am sitting at a restaurant, about to eat my food, and consider the statement $\neg K\mathsf{Poisoned}(\mathsf{food}) \Rightarrow \mathsf{Eat}(\mathsf{food})$, which reads as "*if I do not know that my food has been poisoned, then I eat my food*". Since I do not have enough enemies to believe my food has been poisoned (or, more formally, we cannot derive $K\mathsf{Poisoned}(\mathsf{food})$), we can derive that $\mathsf{Eat}(\mathsf{food})$ holds true, and I proceed to eat my food carelessly. However, if I am provided with some additional information resulting in an extension of my knowlegde base with a fact $\mathsf{Poisoned}(\mathsf{food})$ (or any other fact entailing it), then $K\mathsf{Poisoned}(\mathsf{food})$ is entailed instead of $\mathsf{Eat}(\mathsf{food})$ (and you can rest assured that I am not going to consume my meal). Another prototypical example is a theory T containing a *default*. This could be, for instance, an expression of the form $K\mathsf{Bird}(\mathsf{x}) \wedge \neg K\neg\mathsf{CanFly}(\mathsf{x}) \Rightarrow \mathsf{CanFly}(\mathsf{x})$: "*if I know that* x *is a bird and I do not know* x *cannot fly, then conclude that* x *can fly*". If the only information I know about $Tweety$ is that $Tweety$ is a bird, then $\mathsf{CanFly}(Tweety)$ is entailed because of the default in T. However, if we enlarge my knowledge base with the fact that $Tweety$ is a penguin and penguins cannot fly, then $\neg\mathsf{CanFly}(Tweety)$ is entailed instead.

Non-monotonicity may characterize the induction process of a definition as well. Looking back at the induction proposed in the previous paragraph about natural numbers, we can easily see that its construction process is indeed monotonic: when the antecedent of a rule is satisfied, it remains so at any later stage of the process. Hence, rules can be applied in any order, and they produce a sequence of increasing lower bounds, i.e., of increasingly precise approximations, that lead to the object being defined. In the presence of *non-monotonicity*, this is not the case anymore: a rule applied at a certain stage may not be applicable to derive the same fact at a later stage. Hence, the order in which rules are applied matters in this case.

To circumvent the issues and challenges posed by non-monotonicity, Denecker and his collaborators [Denecker et al., 2000; Denecker and Vennekens, 2007] developed an

innovative framework, called Approximation Fixpoint Theory (AFT). This allows us to obtain the desired objects for several forms of non-monotonic inductions, like inductions over a well-founded order, and unifies the semantics of various non-monotonic logics. Instead of using only a lower bound as an approximation, like for the induction used to define natural numbers, in AFT, two bounds are employed, a lower and an upper one.

In this thesis, we use all the forms of approximations briefly introduced so far in this section (and some new ones) to obtain interesting results in two main areas of NMR: Approximation Fixpoint Theory, and multi-agent Autoepistemic Logic. All these results have been published in some of our papers: [Pollaci, 2025; Pollaci et al., 2024, 2025] for the advancements in AFT, and [Cramer et al., 2023a,b] for the ones in multi-agent AEL. In the following sections, we provide an introduction to both AFT and AEL, with a focus on the problems tackled by the papers considered in this thesis.

2 Approximation Fixpoint Theory

Approximation Fixpoint Theory (AFT) [Denecker et al., 2000] was developed as a powerful algebraic framework to unify the semantics of various non-monotonic logics: AFT successfully characterizes the semantics of LP, AEL [Moore, 1985], and DL [Reiter, 1980], and resolved longstanding problems on the relation between these formalisms [Denecker et al., 2011].

The core ideas of AFT are relatively simple: we are interested in fixpoints of an operator O on a given lattice $\langle L, \leq \rangle$. For monotonic operators, Tarski's fixpoint theory [Tarski, 1955] guarantees the existence of a least fixpoint. In many applications, this least fixpoint is indeed of interest. For non-monotonic operators, the existence of fixpoints is not guaranteed, and even when fixpoints exist, it is not clear which would be "good" fixpoints. AFT generalizes Tarki's theory for monotonic operators to non-monotonic ones, by making use of a so-called *approximator*; this is an operator $A : L^2 \to L^2$, that operates on L^2, and that is monotonic with respect to the precision order $\leq_p$ (defined for all $x, y, u, v \in L$ by $(x, y) \leq_p (u, v)$ iff $x \leq u$ and $v \leq y$)). The intuition is that elements of L^2 approximate elements of L: $(x, y) \in L^2$ approximates all $z \in L$ such that $x \leq z \leq y$, i.e., when $x \leq y$, the tuple (x, y) can be thought of as an interval of L. Analogously, the approximator A approximates the behaviour of O: by studying the fixpoints of A, we are able to obtain information about the original operator O we started with. In particular, given an approximator, AFT defines a few types of fixpoints of interest: supported fixpoints, a Kripke-Kleene fixpoint, stable fixpoints, and a well-founded fixpoint.

In several fields of non-monotonic reasoning, including the ones mentioned above, it is relatively straightforward to define an approximator. Moreover, it turns out that the different types of fixpoints then correspond to existing semantics in the field. In this way,

AFT clarifies on the one hand how different semantics in a single domain relate, and on the other hand what the relation between different (non-monotonic) logics is.

Since its creation, several variants of AFT have been developed, like *consistent* AFT [Denecker et al., 2004], *categorical* AFT [Charalambidis and Rondogiannis, 2023], and *non-deterministic* AFT [Heyninck et al., 2024]; and the domains of applications have grown steadily, including abstract argumentation [Strass, 2013; Bogaerts, 2019], active integrity constraints [Bogaerts and Cruz-Filipe, 2018], stream reasoning [Antic, 2020], integrity constraints for the semantic web [Bogaerts and Jakubowski, 2021], and Datalog [Pollaci, 2025].

In this thesis, we contribute to both the applicative and the theorical sides of AFT: in Chapter 3, we present the fixpoint semantics of DatalogMTL$^\neg$, obtained using consistent AFT; whereas in Chapter 4 we define a framework to extend various versions of AFT to allow for higher-order definitions. In the next two subsections, we introduce these topics and briefly list our contributions.

2.1 A Novel Application: DatalogMTL$^\neg$

Datalog [Abiteboul et al., 1995] is a declarative query language that extends relational algebra with recursion. A Datalog program is an inductive definition of new symbols in terms of some facts. For instance, we could consider the rules defining the transitive closure $\mathsf{TransCL}$ of a graph:

$$\mathsf{TransCL}(x, y) \leftarrow \mathsf{Edge}(x, y)$$
$$\mathsf{TransCL}(x, y) \leftarrow \mathsf{TransCL}(x, z) \wedge \mathsf{Edge}(z, y)$$

where Edge is the edge relation. While the rules above are generic, in the sense that they are not referring to a specific graph, we can additionally provide a *dataset* $\mathcal{D}$, i.e., a set of facts, containing the edges of the graph we want to query about; for example, $\mathcal{D} := \{\mathsf{Edge}(a, b), \mathsf{Edge}(b, c), \mathsf{Edge}(b, d)\}$. We can then query whether the edge (a, d) is in the transitive closure of a graph with edges provided by the facts in $\mathcal{D}$, i.e., whether $\mathsf{TransCl}(a, d)$ holds.

Enabling recursion and the simple declarative interface made handling graph and network queries with Datalog easy and accessible. This resulted in a long array of applications for both Datalog and its later extensions, like declarative networking [Loo et al., 2006], security [Marczak et al., 2010], graph analysis [Seo et al., 2015; Shkapsky et al., 2016], business analytics [Aref et al., 2015], program analysis [Whaley and Lam, 2004; Smaragdakis and Balatsouras, 2015], ontology-based query answering [Brandt et al., 2018; Kalayci et al., 2018; Kikot et al., 2018; Koopmann, 2019], stream reasoning [Walega et al., 2019, 2023],

and reasoning for the financial sector [Nissl and Sallinger, 2022; Mori et al., 2022; Colombo et al., 2023], amongst others.

Among all the variants of Datalog that have emerged during the last decades, we are interested in DatalogMTL [Brandt et al., 2018] and its extension Datalog MTL$^\neg$ [Walega et al., 2024]. The former enriches the Datalog language with operators from Metric Temporal Logic (MTL) [Koymans, 1990], while the latter further extends DatalogMTL with unstratifiable negation under the stable model semantics. MTL provides temporal operators that are constrained over a time interval, which may refer to the past or future. For instance, consider we want to express a rule regarding the renewal of driving licences. For safety reasons, senior citizens over 80 are required to renew their licence every two years. We could then write the following rule using the language of Datalog MTL$^\neg$:

$$\mathsf{Expired}(\ell) \leftarrow \mathsf{Licence}(\ell) \wedge \mathsf{Person}(x) \wedge \mathsf{LicenceOf}(\ell, x) \wedge \underset{[80,\infty)}{\diamondminus} \mathsf{Birth}(x) \wedge \neg \underset{(0,2)}{\diamondminus} \mathsf{Renew}(\ell),$$

which reads as "*ℓ is expired if ℓ is a driving licence, ℓ is assigned to a person x, x's birth happened at least 80 years before now, and ℓ has not been renewed in the last two years*".

Contributions. As with many other applications, the strong mathematical foundation of AFT and its broad range of general results work wonders, reducing the amount and complexity of proofs, and providing reliability to the research outcome. Our contributions to the study of DatalogMTL$^\neg$, contained in Chapter 3 and published in [Pollaci, 2025], can be summarized as follows:

- We define several semantics of interest for DatalogMTL$\neg$ using consistent AFT: the Kripke-Kleene, well-founded, stable, and supported model semantics. We achieve this by defining the correct consistent approximator, which is nothing more than a natural adaptation of Fitting's three-valued immediate consequence operator to the setting of DatalogMTL$\neg$.
- We prove that the stable models obtained via AFT are equivalent to the ones defined by Walega et al. [2024] through interpretations stemming from the logic of Here-and-There (HT).

2.2 Higher-Order AFT

Higher-order abstract concepts are ubiquitous, and reasoning formally about them may be rather complex. For instance, one could be interested in defining a second-order predicate IsTransitive that takes as input a first-order predicate P of arity 2, i.e., a binary relation, and returns true if P is transitive. Over the course of the last decades, a few devices have been studied to deal with higher-order objects, like macros and templates [Dasseville et al.,

2015], a higher-order syntax has been developed for HiLog [Chen et al., 1993] and λprolog [Nadathur and Miller, 1988], higher-order logics have been proposed [Dasseville et al., 2016; Charalambidis et al., 2018], and higher-order functions are supported in several popular functional programming languages.

AFT is not readily applicable to higher-order definitions: using pairs of interpretations no longer allows for an obvious way to evaluate formulas in an approximation. For instance, suppose we have a logic program in which the second order predicate IsTransitive introduced above is defined. A pair of interpretations (I, J) provides an approximation of the truth value of IsTransitive(S) for any set S of pairs: IsTransitive(S) evaluates to true (resp. false) in (I, J) if IsTransitive(S) is true (resp. false) in both I and J, and IsTransitive(S) evaluates to *undefined* in (I, J) if it is false in I but true in J. One could even consider a fourth scenario: IsTransitive(S) evaluates to true in I and to false in J, in which case we would say that IsTransitive(S) is *inconsistent* in (I, J). So far so good: this is what AFT already performs successfully. Suppose now that, in the same program where IsTransitive is defined, a first-order predicate Q of arity 2 is also defined, and that IsTransitive(Q) appears in the body of a rule of such a program. While Q is being defined, its interpretation via (I, J) is not a set, but rather a partially defined set, i.e., an approximation. Thus, figuring out what IsTransitive(Q) evaluates to in (I, J) becomes challenging: the approximation of an interpretation of IsTransitive should take as argument not only (interpretations of) binary relations, but also *approximations* of them, i.e., partially defined ones. This calls for the definition of new, richer approximation spaces where approximate objects can be applied to other approximate objects. Intuitively, the approximation spaces of higher-order objects should be defined inductively from lower-order ones, following the type hierarchy. This has been attempted before in the context of consistent AFT for LP by Charalambidis et al. [2018]. Nevertheless, such work lacks the notion of exactness, making it impossible to recognise two-valued stable models, and the obtained well-founded semantics behaves oddly on certain instances. The main question of Chapter 4 is how to define approximation spaces for higher-order objects in a generic way that works, not only for consistent AFT, but for most of the present (and future) versions of AFT, and hence, it can be employed in all applications of AFT. We answer this question by drawing a concept from the field of Category Theory (CT).

Category Theory. After being introduced by Eilenberg and MacLane [1945], the concept of category evolved over the years [Marquis, 2026; Riehl, 2017]. In short, one could view a category as consisting of a collection of *objects* and a collection of *morphisms*, i.e., relations between objects. For instance, we could define a category of lattices, where objects are lattices and morphisms are monotonic functions. In our problem, we want to apply the same AFT techniques on approximation spaces at any hierarchy level; thus, we need to preserve the same mathematical structure: this can be achieved using *Cartesian*

closed categories. The objects of a Cartesian closed category $\mathbf{C}$ satisfy a property that can be intuitively understood as follows: if A and B are two objects of $\mathbf{C}$, then the set of morphisms from A to B is also an object of $\mathbf{C}$. In our setup, if the base approximation spaces are objects of a Cartesian closed category, then the same category contains the full hierarchy of spaces we are aiming for.

Contributions. Our contributions to the development of higher-order AFT are contained in Chapter 4 and published in [Pollaci et al., 2024, 2025]. They can be summarised as follows:

- We define the concepts of approximation category, approximation system, and exactness, which provide a mathematical foundation for using various accounts of AFT for higher-order definitions.
- We generalize the work of Charalambidis et al. [2018] to a category-theoretic setting. In doing so, we shed light on the general principles underlying their constructions for higher-order logic programming and make it applicable to arbitrary current and future non-monotonic reasoning formalisms captured by AFT.
- We obtain an approximation system that recovers the framework of standard AFT and extends it to higher-order objects.
- We improve on the work of Charalambidis et al. [2018] by defining a new approximator, which provides the expected well-founded semantics. Moreover, by studying the concept of *exactness*, previously missing, we allow the use of the theory to define exact stable models instead of focusing purely on the well-founded model.

3 Multi-Agent Autoepistemic Logic

Epistemology, i.e., the study of knowledge, is one of the core areas of both philosophy and formal logic, with applications ranging from politics to game theory, from sociology to economics. Epistemology focuses on all the aspects of knowledge and its related concepts, like ignorance, belief, memory, and faith, and strives to answer questions on how knowledge is acquired, lost, transferred, justified, and so on.

While the word "epistemology" has only been coined in the 19th century, the study of knowledge finds its roots in the first Greek, Chinese, and Indian philosophers. In the context of Knowledge Representation (KR), the formal study of knowledge and how to reason correctly about it dates back at least to the 1960s [Hintikka, 1962]. A couple of decades later, Moore [1985] proposed his Autoepistemic Logic (AEL) with the intent to model the knowledge or beliefs of a rational agent with perfect introspection, i.e., the

agent knows what they know and do not know. To do so, Moore [1985] enriches FO with a novel modal operator K: statements of the form $K\varphi$ ought to be read as *I know* φ.

The study of knowledge becomes particularly interesting (and rather complex) when multiple agents are involved, and they are able to reason about each other's knowledge. For instance, one may want to formally express something like "*Alex knows that Bart does not know that Carlos knows …*". This type of statements and reasoning are rather common in several real-life situations. Single-agent epistemic logic or autoepistemic logic is unsuitable to express such information. While AEL allows only for one agent, in recent years there have been several attempts at extending AEL to the multi-agent setting under different conditions ([Morgenstern, 1990; Morgenstern and Guerreiro, 1996; Vlaeminck et al., 2012] and [Van Hertum et al., 2016] among others). In this thesis, next to the standard knowledge operator K, we are concerned with two epistemic operators, namely *common knowledge* and *only knowing*, and their intricate interplay in the multi-agent setting.

Common Knowledge. When developing intelligent agents, it is important that they can reason not just about the world they are placed in, but also about the knowledge or beliefs of other agents, and groups thereof, in their environment. For instance, consider the situation in which two anglophones meet. After having realised they are both English-speaking, they start a conversation about cherry blossoms, and the first person uses the word "tree" while talking, with no hesitation. This may seem trivial, but the free usage of the word "tree" (or any other word) is supported by some implicit assumptions: on the one hand, the first person knows the meaning of the word "tree"; on the other hand, the same person also knows (or believes) that the second person knows the meaning of the word "tree" and that the first person also knows this. It is this knowledge about the other person's knowledge that allows the conversation to happen. In this case, we may assume that *the meaning of the word "tree" is common knowledge among anglophones.* More generally, we say that a statement φ is *common knowledge* among a group G of agents (and denote it as $C_G\varphi$) if each agent in G knows φ, and also knows that every agent in G knows φ, and knows that everyone in G knows that everyone in G knows φ, and so on.

The notion of common knowledge made its first appearence in the philosophical literature, in a book by Lewis [1969], in which the author strives to analyse the concept of *convention* and to explain how "*language - like many other activities - is governed by conventions*". Almost a decade later, the concept of common knowledge made its way into mathematical text [Aumann, 1976], and it is now well-known[1] and has been extensively studied [Fagin et al., 1995; Meyer and van der Hoek, 1995].

[1] One may say that *common knowledge is common knowledge among a group of fellow logicians.*

Only Knowing. In the context of games, politics, and strategic planning, for instance, it may be interesting to express what an adversary knows *at most*, to choose the best next move to make. This is particularly useful when we consider that knowledge-based agents do not know anything except for what follows from their knowledge base. We say that an agent *only knows* a statement φ (and denote this $O_A\varphi$) if the agent knows φ and moreover *everything* they know follows from φ (so whenever the agent knows ψ, it must be the case that φ entails ψ).

Since the Eighties, several concepts closely related to only knowing were formalised, like ignorance [Halpern and Moses, 1984; Konolige, 1982], minimal knowledge [van der Hoek and Thijsse, 2002], and total knowledge [Pratt-Hartmann, 2000]. The notion of only knowing, as we intend it, came to life shortly after, in a single-agent logic framework by Levesque [1990]. This was later successfully studied and implemented [Halpern and Lakemeyer, 1995; Rosati, 2000; Levesque and Lakemeyer, 2000; Lakemeyer and Levesque, 2005], revealing that it is also possible to capture DL [Reiter, 1980] and a variant of AEL proposed by Konolige [1988].

The journey to extend only knowing to a multi-agent logic had several turns [Halpern, 1993; Lakemeyer, 1993; Halpern and Lakemeyer, 2001; Waaler, 2004; Waaler and Solhaug, 2005; Belle and Lakemeyer, 2010]. In 2015, Belle and Lakemeyer [2015b] successfully extended Levesque's logic, while keeping the original idea of Levesque's worlds and generalizing its features at the same time: they proposed to use models limited to a finite depth k, i.e., models at which beliefs can be nested at most k times, by introducing the concept of k-structure to represent an agent's epistemic state.

The difficulties of Joining Common Knowledge and Only Knowing Even though there have been a few attempts at defining a semantics that joins common knowledge and only knowing [Aucher and Belle, 2015; Belle and Lakemeyer, 2015a; Van Hertum, 2016], none of these approaches fully captures the nature of only knowing and some of its essential properties. For instance, the following two properties of the only-knowing operator, while quite simple and natural, are not satisfied by any other previous work combining common knowledge and only knowing:

- If an agent only knows a formula φ, then all formulas not entailed by φ are not known;
- For each formula φ and agent $\mathcal{A}$, there exists a world[2] in which $\mathcal{A}$ only knows φ.

Let us take a step back. A standard way of defining a semantics for autoepistemic logic is by using *Kripke structures*. A Kripke structure consists of a set W of *possible worlds*,

[2]Informally, a "world" is a representation of a possible situation, describing both the *objective state of affairs* and the *knowledge of the agents*. The formal definition of *world*, when considering both common knowledge and only knowing, is rather complicated, and it is going to be the central topic of Chapter 5.

and an *accessibility relation*, i.e., a binary relation on W, or, in the case of multi-agent autoepistemic logic, an accessibility relation $R_{\mathcal{A}}$ for each agent $\mathcal{A}$. Intuitively, a possible world can be thought of as a description of a possible state of affairs. Looking back at the restaurant example proposed in the paragraph about non-monotonic reasoning in Section 1, a possible world may be the one in which both Poisoned(food) and Eat(food) hold. The accessibility relation $R_{\mathcal{A}}$ provides the information concerning the knowledge of $\mathcal{A}$: $\mathcal{A}$ can access a world w' from a world w, if and only if $\mathcal{A}$ believes the state of affairs described by w' is possible when being in the state of affairs described by w. Employing this relation, the semantics of several epistemic operators can be defined. For instance, if Poisoned(food) holds in all the worlds that I can access from a world w, then KPoisoned(food) holds in w.

Given a Kripke structure, defining the semantics for a logic with both an operator for common knowledge and one for only knowing is rather straightforward. However, what is challenging is defining the set of such *possible worlds*. A naive first attempt could be to define a world as consisting of two components: 1) an *objective interpretation*, describing the state of affairs while disregarding any information about the knowledge of the agents; and 2) for each agent, a set of worlds that agent deems possible. Unfortunately, the circularity of this definition, which sees a world as a tuple made up of, among others, a set of worlds, breaks the rules of set theory. To avoid this type of issue, the most common approach is to *approximate* the knowledge of agents up to a certain level. In particular, one could define k+1-worlds as consisting, among others, of a set $\mathcal{A}^w$ of k-worlds for each agent $\mathcal{A}$, where a k-world encodes a state of affairs with the knowledge of the agents only up to level k. Unfortunately, this comes with two main issues. First, common knowledge is inherently infinite in nature, and it makes it necessary to have *infinitely deep* worlds. Fagin et al. [1991] and Belle and Lakemeyer [2015a] tried to achieve this by using infinite *precision-increasing* sequence of k-worlds. Nevertheless, there is a need to have worlds that are even deeper than this to evaluate certain formulas. Secondly, in such approximations, there is never enough information to conclude that this is "all we know": it might always be that by making the approximation more precise, more knowledge at some later level comes in.

Contributions. We solve the above problem by defining the notion of μ-biworld, where μ can be any countable ordinal. To design this novel concept, we draw some inspiration from the infinite sequences of k-worlds of Fagin et al. [1991] and Belle and Lakemeyer [2015a], and from the approximation techniques adopted in AFT. Intuitively, a μ-biworld encodes the knowledge of the agents only partially, up to depth μ, i.e., it approximates their knowledge; and it does so by providing two bounds: one on what an agent deems possible, and another on what they deem impossible. More precisely, with each μ+1-biworld w, we associate a set $\mathcal{A}^w$ of μ-biworlds representing the set of μ-biworlds of which the agent

deems some extension possible, and a set $\bar{\mathcal{A}}^w$ of μ-biworlds of which the agent deems some extension impossible[3].

Our contributions in this regard are contained in Chapter 5 and are published in [Cramer et al., 2023a,b]. They can be summarised as follows:

- We define the novel concept of μ-biworld for countable ordinals μ, and we show that such a definition satisfies a series of desirable properties (Theorem 5.1).
- We define the language $\mathcal{COL}$, extending propositional logic with the modal operators $K_{\mathcal{A}}$ and C_G among others[4], and a three-valued model semantics for it.
- We define a canonical Kripke structure over completed[5] ω^2+1-biworlds, and the two-valued semantics obtained from it is shown to coincide with the model semantics.
- We prove that our logic satisfies several desirable properties (Theorem 5.5). In particular, we show that for any formula φ, there is a unique state-of-mind of a given agent in which they only know φ.
- We investigate how our framework can be extended to satisfy truthfulness, positive introspection, and/or negative introspection. While positive introspection can be successfully integrated in our setup, truthfulness and negative introspection create some problems when paired with only knowing. This is not caused by the design of our framework, but it is inherent to the properties we want the concepts to satisfy.

4 Reading Guide

This dissertation includes the main topics I researched during my PhD, all of which share the usage of approximations to define semantics.

Chapter 2 introduces the main concepts, definitions, and results regarding Order Theory (OT) and AFT. The foundations of AFT are necessary to fully comprehend Chapters 3 and 4, whereas they are not required for the understanding of Chapter 5.

Chapters 3, 4, and 5 constitute the core of the thesis, containing the main results we have already briefly presented in the three 'Contributions" paragraphs in the sections above. They are all based on conference and/or journal papers that we co-authored, with slight modifications and additions. The former two concern AFT, while the latter regards multi-agent AEL.

[3]For limit ordinals, the situation is more complex, which makes the construction highly technical and mathematical.

[4]The language $\mathcal{COL}$ also includes formulas with the *knowing at most* operator $M_{\mathcal{A}}$ for an agent $\mathcal{A}$, and the *everybody knows* operator E_G for a set of agents G. The only-knowing operator $O_{\mathcal{A}}$ is syntactic sugar for the conjunction of $K_{\mathcal{A}}$ and $M_{\mathcal{A}}$.

[5]A *completed* biworld contains enough information to identify, for every other biworld of any other level, whether or not the agent believes it is possible, i.e., it characterizes complete knowledge.

We invite the reader interested in AFT to start with Chapter 3, which presents one of the several applications of AFT, namely DatalogMTL$^{\neg}$. The more practical approach of this chapter allows for a gentle familiarization with AFT before heading to Chapter 4, which is more theory-oriented and requires more time to be fully digested. The reader who is unfamiliar with CT is invited to visit Section 2.2 of Chapter 4, where all the category-theoretical concepts used in the chapter are formally presented.

To those intrigued by AEL and knowledge constructs, we suggest heading to Chapter 5. This chapter does not require an understanding of the previous two and can be read independently.

The last chapter (Chapter 6) summarizes the contributions and results of this dissertation and provides an outlook on possible future research directions.

Finally, a couple of brief editorial remarks for a smoother reading. When reading a chapter x, a reference to a section with no chapter specified always refers to chapter x. All the acronyms and symbols used throughout the thesis are itemized in the respective lists prior to the table of contents.

2

Background

In this chapter, we present the technical foundation of AFT, with the concepts and results that will later be used in Chapters 3 and 4. To do so, we first introduce the base concepts of Order Theory that sits at the core of AFT.

1 An Introduction to Order Theory

Ordering, in all its declinations, permeates almost any aspect of human life, and it determines several of our choices. From school grades to scores of athletic performances, from ratings of movies or restaurants to prices at supermarkets, from scores of the reviews of an academic paper to ones of a product on an online selling platform, we are used to comparing elements of a certain set, based on efficiency, expensiveness, beauty, quality, and so on. We are accustomed to considering different orderings over the same set: we can compare two dishes based on their healthiness, expense, or simply on our personal taste. Nevertheless, in any of these situations, it all boils down to a certain binary relation over a set of elements.
Order Theory (OT) is the branch of Mathematics that studies special binary relations called *orders*. In this section, we present the basic concepts drawn from OT that we use throughout this thesis. In particular, we introduce the definitions of a partial or total order, of a partially ordered set (poset), together with some special types of posets, and finally

of an *operator* over a poset, and its *fixpoints*. For further information and results on OT, we refer the reader to the book of Davey and Priestley [2002].

Definition 2.1

A binary relation $\leq$ between a set P and itself is a (*partial*) *order* on P if it is:

- *reflexive*, i.e., for each $a \in P$, $a \leq a$;
- *antisymmetric*, i.e., for each $a, b \in P$, if $a \leq b$ and $b \leq a$, then $a = b$; and
- *transitive*, i.e., for each $a, b, c \in P$, if $a \leq b$ and $b \leq c$, then $a \leq c$.

For any two elements $a, b \in P$, we use the standard notation $a < b$ to denote that $a \leq b$ and $a \neq b$; in this case, we say that a is *strictly* smaller than b, and b is *strictly* greater than a. A *partially ordered set* (poset) $\mathcal{P}$ is a set equipped with a partial order. We denote a poset by $\mathcal{P} = \langle P, \leq \rangle$, where P is the underlying set, and $\leq$ the order on it. By abuse of notation, when referring to a poset $\mathcal{P}$, we often use the notation for the underlying set P in place of the calligraphic one. We denote by $\mathcal{P}^{op}$ the poset with the same underlying set as $\mathcal{P} = \langle P, \leq \rangle$ but opposite order, i.e., $\mathcal{P}^{op} = \langle P, \geq \rangle$. It is trivial to show that $\mathcal{P}^{op}$ is indeed a poset for any given poset $\mathcal{P}$. Notice that for any poset $\langle P, \leq \rangle$ and any subset $S \subseteq P$, the restriction of the order $\leq$ on S, which we denote by $\leq \cap S^2$, is an order on S. It follows that S equipped with such an order relation is also a poset. We call $\langle S, \leq \cap S^2 \rangle$ a *subposet* of $\langle P, \leq \rangle$, and the order $\leq \cap S^2$ the *order induced by P on S*, or simply the *induced order*. By abuse of notation, we often denote the order $\leq \cap S^2$ by just $\leq$.

Example 2.1

Let S be a set. We denote by 2^S the *powerset of S*, i.e., the set of all subsets of S. The pair $\mathcal{P} := \langle 2^S, \subseteq \rangle$, where $\subseteq$ is the subset relation, can easily be shown to be a poset. Then, $\mathcal{P}^{op}$ is just the poset given by equipping the powerset 2^S with the superset relation $\supseteq$.

Example 2.2

Let $\mathbb{N}^* := \mathbb{N} \setminus \{0\}$ be the set of natural numbers without 0. We can define a binary relation $|$ on $\mathbb{N}^*$ as follows: for all $x, y \in \mathbb{N}^*$, $x \mid y$ if and only if y is divisible by x. It is immediate to see that $|$ constitutes a partial order on $\mathbb{N}^*$. Hence, $\langle \mathbb{N}^*, | \rangle$ is a poset.

Notice that in both examples above (Examples 2.1 and 2.2) there are some elements which are not comparable. For instance, if we consider the set $S := \{a, b, c\}$, clearly $\{a, b\} \not\subseteq \{b, c\}$ and $\{b, c\} \not\subseteq \{a, b\}$. Similarly, it holds that $2 \nmid 3$ and $3 \nmid 2$. In the special case when elements of a set are pair-wise comparable, we talk of a *total order*.

Definition 2.2

An order $\leq$ on a set P is *total* if for all $a, b \in P$, either $a \leq b$ or $b \leq a$. A set equipped with a total order is called a *totally ordered* set.

When considering a poset $\langle P, \leq \rangle$, it might turn out usefull to focus on certain portions of P. In particular, It may be of special interest to look for subsets of P over which $\leq$ is total. Such subsets are called *chains*.

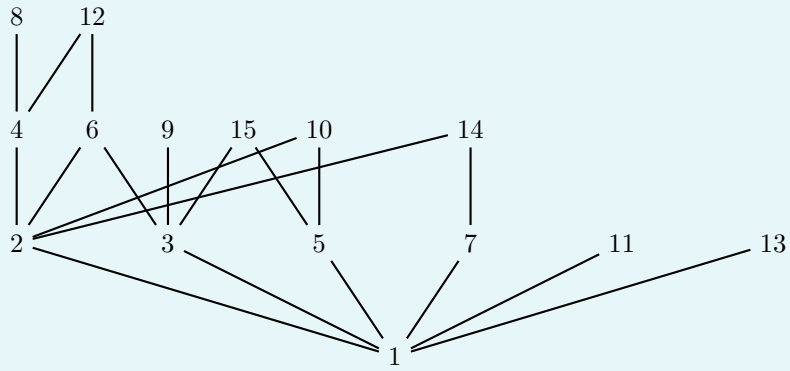
Definition 2.3

Let $\langle P, \leq \rangle$ be a poset. A subset $S \subseteq P$ is called a *chain* if $\langle S, \leq \cap S^2 \rangle$ is a totally ordered subposet of $\langle P, \leq \rangle$.

Observe that, by the definition above, the empty set is a chain for any poset.

Example 2.3

(Example 2.2 continued). We can consider the subset $\mathbb{N}^*_{15} := \{1, 2, \dots, 15\} \subseteq \mathbb{N}^*$ of the natural numbers from 1 to 15. We can visually represent the elements of $\langle \mathbb{N}^*_{15}, | \ \rangle$ with the Hasse diagram below.



Observe that $\langle \mathbb{N}^*_{15}, | \ \rangle$ is not totally ordered. On the contrary, the subset $\{1, 2, 4, 8\}$ made up of the powers of 2 up to 8, equipped with the order induced by $\mathbb{N}^*$, is indeed a totally ordered subposet of $\mathbb{N}^*$ (and of $\mathbb{N}^*_{15}$). As such, it is a chain of both $\mathbb{N}^*$ and $\mathbb{N}^*_{15}$.

When dealing with order relations, it is natural to define certain distinguished elements in a poset, as we do in what follows.

Definition 2.4

Let $\langle P, \leq \rangle$ be a poset, and $S \subseteq P$ be a subset of P. An element $x \in P$ is

- a *lower bound of* S if for all $y \in S$, $x \leq y$;
- an *upper bound of* S if for all $y \in S$, $y \leq x$;
- the *greatest lower bound (glb)* (or *infimum* or *meet*) *of* S if x is a lower bound of S and for all lower bounds y of S, $y \leq x$;
- the *least upper bound (lub)* (or *supremum* or *join*) *of* S if x is an upper bound of S and for all upper bounds y of S, $x \leq y$;
- the *bottom* (or *least element*) of S if x is a lower bound of S and $x \in S$;
- the *top* (or *greatest element*) of S if x is an upper bound of S and $x \in S$;

We denote by $\mathrm{glb}_P(S)$, $\mathrm{lub}_P(S)$, $\bot_S$, and $\top_S$, the greatest lower bound, least upper bound, bottom, and top of S, respectively, if they exist. For the sake of simplicity, we omit the subscript P when clear from the context. We sometimes also write $\mathrm{glb}_{\leq}(S)$ or $\mathrm{lub}_{\leq}(S)$ to specify which order we consider on S to compute the greatest lower bound or the least upper bound, respectively.

Example 2.4

(Example 2.3 continued). We can see that 1 is a lower bound of $\mathbb{N}^*_{15}$. It is actually the only lower bound, and thus, also the greatest lower bound. Since $1 \in \mathbb{N}^*_{15}$, $\mathrm{glb}(\mathbb{N}^*_{15}) = \bot_{\mathbb{N}^*_{15}} = 1$. On the contrary, there are infinitely many upper bounds of $\mathbb{N}^*_{15}$: any multiple of $2^3 \cdot 3^2 \cdot 5 \cdot 7 \cdot 11 \cdot 13 = 360360$ constitutes an upper bound. It follows easily that the least upper bound of $\mathbb{N}^*_{15}$ is 360360, and thus, $\mathbb{N}^*_{15}$ has no top.
Observe that, in general, for any subset $S \subseteq \mathbb{N}^*$, the greatest lower bound and the least upper bound of S correspond to the greatest common divisor and the least common multiple of the numbers in S, respectively.

As seen in Example 2.4, it is not always the case that all the distinguished elements defined in Definition 2.4 exist for a given (subset of a) poset. Depending on the existence of least upper bounds and greatest lower bounds, we differentiate between three types of posets.

Definition 2.5

A poset $\langle P, \leq \rangle$ is

- a *chain-complete poset (cpo)* if for every chain $S \subseteq P$, $\mathrm{lub}(S)$ exists;
- a *complete join semilattice* if for every subset $S \subseteq P$, $\mathrm{lub}(S)$ exists;
- a *complete lattice* if for every subset $S \subseteq P$, $\mathrm{lub}(S)$ and $\mathrm{glb}(S)$ exist.

Observe that, for any poset $P \neq \emptyset$, if $\mathrm{lub}_P(\emptyset)$ exists, then $\mathrm{lub}_P(\emptyset) = \bot_P$. Since the empty set $\emptyset$ is a chain, by the definition above, any cpo has a bottom element. Moreover, every complete lattice is also a complete join semilattice, and every complete join semilattice is a cpo.

Example 2.5

The poset $\langle \mathbb{N}^*, | \, \rangle$ defined in Example 2.2 is not a cpo. In fact, consider the set S composed by all the powers of 2, i.e., $S := \{2^k \mid k \in \mathbb{N}\}$. It is easy to see that, while S is totally ordered (and hence, a chain of $\mathbb{N}^*$), S does not have any upper bound.

Example 2.6

The poset $\langle \mathbb{N}^*_{15}, | \, \rangle$ defined in Example 2.3 is a cpo. This is trivial to show since every chain is finite. However, $\langle \mathbb{N}^*_{15}, | \, \rangle$ is not a complete join semilattice: for instance, the set $\{8, 12\} \subseteq \mathbb{N}^*_{15}$ does not have a least upper bound in $\mathbb{N}^*_{15}$.

Example 2.7

Observe that any subset of the poset $\langle \mathbb{N}^*_{15}, | \, \rangle$ defined in Example 2.3 has a greatest lower bound. Hence, the poset $\langle \mathbb{N}^*_{15}, | \, \rangle^{op}$, obtained by inverting the order of $\langle \mathbb{N}^*_{15}, | \, \rangle$, is a complete join semilattice. Nevertheless, $\langle \mathbb{N}^*_{15}, | \, \rangle^{op}$ is not a complete lattice for the same reasons why $\mathbb{N}^*_{15}$ is not a complete join semilattice.

Example 2.8

For any set S, the poset $\langle 2^S, \subseteq \rangle$ defined in Example 2.1 is a complete lattice: given a subset $A \subseteq S$, the least upper bound of a A is simply the union of the sets in A, and its greatest lower bound is their intersection.

Example 2.9: Logic Programming

For the sake of simplicity, in this example we focus on *propositional* logic programming. We consider a propositional vocabulary Σ, i.e., a collection of symbols called *atoms*. A *(two-valued) interpretation* I is a function sending a symbol of Σ to either true (denoted with $\mathbf{t}$), or false (denoted with $\mathbf{f}$). We denote by E_o the set of truth values $\{\mathbf{t}, \mathbf{f}\}$. Notice that E_o is a complete lattice when equipped with the order relation given by $\mathbf{f} \leq_o \mathbf{t}$. The set of all two-valued interpretations can be endowed

with the *pointwise order* induced by $\leq_o$, which we again denote by $\leq_o$: for all two valued-interpretations I and J, $I \leq_o J$ if and only if $I(p) \leq_o J(p)$ for all $p \in \Sigma$. The set of two-valued interpretations equipped with such a pointwise order is a complete lattice[a].

Analogously, we can identify a two-valued interpretation I with the set of symbols of Σ which are mapped to $\mathbf{t}$ via I, i.e., $I \mathrel{\widehat{=}} \{p \mid p \in \Sigma \text{ s.t. } I(p) = \mathbf{t}\}$. Then the set of all two-valued interpretation is simply the powerset 2^{Σ}, which is a complete lattice when endowed with the subset order relation (Example 2.8). It is easy to show that the two complete lattices that we have obtained are *isomorphic*, i.e., there exist a monotonic bijection between them and a monotonic inverse (see Definition 2.6). We will use these two views of interpretations, as functions and as sets, interchangeably.

[a] This follows immediately from Proposition 4.3 and **CLat** $\subseteq$ **POSet** (see Section 2.2 in Chapter 4 for more details).

1.1 Operators and Fixpoints

So far, we have seen different kinds of posets and elements of interest in them. In this subsection we focus on functions between posets, and, in particular, *operators* and their *fixpoints*.

Definition 2.6

A function $f : \langle P_1, \leq_1 \rangle \to \langle P_2, \leq_2 \rangle$ between posets is

- *monotonic* if for all $x, y \in P_1$ such that $x \leq_1 y$, it holds that $f(x) \leq_2 f(y)$;
- *antimonotonic* if for all $x, y \in P_1$ such that $x \leq_1 y$, it holds that $f(y) \leq_2 f(x)$.

Given two sets P_1, P_2, we denote by $(P_1 \to P_2)$ the set of all functions from $\mathcal{P}_1$ to $\mathcal{P}_2$. If $\langle P_2, \leq \rangle$ is a poset, we can easily define an order on $(P_1 \to P_2)$ (induced by $\leq$), which we again denote by $\leq$: for all $f, g \in (P_1 \to P_2)$, $f \leq g$ if and only if $f(x) \leq g(x)$ for all $x \in P_2$. We call such order on $(P_1 \to P_2)$ the *pointwise order* (induced by $\leq$). It is easy to show that $(P_1 \to P_2)$ equipped with its pointwise order is a poset, a cpo, a complete join semilattice, and a complete lattice, if P_2 is a poset, a cpo, a complete join semilattice, and a complete lattice, respectively.

We refer to functions between posets $O : \langle P, \leq \rangle \to \langle P, \leq \rangle$ with domain equal to the codomain as *operators*. For the sake of simplicity, we often use the notation $O : P \to P$ to denote an operator on the poset $\langle P, \leq \rangle$, instead of $O : \langle P, \leq \rangle \to \langle P, \leq \rangle$.

Definition 2.7

Let $O : \langle P, \leq \rangle \to \langle P, \leq \rangle$ be an operator. We call an element $x \in P$

- a *fixpoint* of O if $O(x) = x$;
- a *pre-fixpoint* of O if $O(x) \leq x$;
- a *post-fixpoint* of O if $x \leq O(x)$;
- the *greatest* fixpoint of O, denoted by $\mathrm{gfp}(O)$, if it is a fixpoint and for all fixpoints y of O, $y \leq x$;
- the *least* fixpoint of O, denoted by $\mathrm{lfp}(O)$, if it is a fixpoint and for all fixpoints y of O, $x \leq y$.

Example 2.10: Logic Programming

(Example 2.9 continued). Consider a propositional vocabulary Σ, and a logic program $\mathfrak{P}$, i.e., a set of rules of the form $p \leftarrow \varphi$, where p is an atom (the *head* of the rule), and φ is a propositional formula (the *body* of the rule). Given a two-valued interpretation I, the evaluation of a formula φ, denoted by φ^I, is provided by the standard truth tables of propositional logic. An example for an operator on the set of two-valued interpretaions 2^Σ is the *two-valued immediate consequence operator* $T^\mathfrak{P}$ [van Emden and Kowalski, 1976] that sends an interpretation I to $\{p \mid \exists p \leftarrow \varphi \in \mathfrak{P} \text{ s.t. } \varphi^I = \mathbf{t}\} \in 2^\Sigma$, i.e., the set containing every symbol which is in the head of a rule whose body evaluates to true in I.

Clearly, not all operators have fixpoints. However, this is not the case for monotonic operators between complete lattices, which always have a greatest fixpoint and a least fixpoint. This is guaranteed by one of the core theorems of OT, namely the Knaster-Tarski Theorem [Tarski, 1955].

Theorem 2.1: Knaster-Tarski Theorem

Let $\langle P, \leq \rangle$ be a complete lattice and $O : P \to P$ a monotonic operator. The set of fixpoints of O, equipped with $\leq$, forms a complete lattice. In particular, the least and greatest fixpoint coincide with the least pre-fixpoint and the greatest post-fixpoint, respectively.

A similar result was later shown to hold for cpo's by Markowsky [1976].

Theorem 2.2: Markowsky Theorem

Let $\langle P, \leq \rangle$ be a cpo and $O : P \to P$ a monotonic operator. The set of fixpoints of O, equipped with $\leq$, forms a cpo. Moreover, the least fixpoint of O coincides with its least pre-fixpoint.

While Theorems 2.1 and 2.2 ensure the existence of the least fixpoint of a monotonic operator, they do not provide any information on how to find it. However, looking at operators from a different perspective allows us to obtain a constructive characterization of the least fixpoint of a monotonic operator over a cpo. Given a cpo and a monotonic operator on it, we can define the concept of a *monotonic induction*.

Definition 2.8

Let $\langle P, \leq \rangle$ be a cpo, and $O : P \to P$ a monotonic operator. A *monotonic induction of* O is a (possibly transfinite) sequence $(x_\alpha)_{\alpha \leq \mu}$ such that

1. $x_0 = \bot_P$;
2. $x_\alpha < x_{\alpha+1} \leq O(x_\alpha)$ for every $\alpha < \mu$;
3. $x_\lambda = \text{lub}(\{x_\alpha \mid \alpha < \lambda\})$ for every limit ordinal $\lambda \leq \mu$.

A monotonic induction $(x_\alpha)_{\alpha \leq \mu}$ is *terminal* if it cannot be extended, i.e., there is no $x_{\mu+1}$ such that $(x_\alpha)_{\alpha \leq \mu+1}$ is a monotonic induction.

Notice that a monotonic induction of an operator O does not necessarily coincide with the iterative application of such an operator starting from the bottom element of the cpo over which O is defined. More precisely, the element $x_{\alpha+1}$ of a monotonic induction is bounded by the image of x_α via O, as stated in the second item of Definition 2.8. It follows that there might be multiple monotonic inductions for the same operator O.
Nevertheless, all terminal monotonic inductions of a monotonic operator O are *confluent*, i.e., they all have the same limit, and such limit coincides with the least fixpoint of O. In other words, if $(x_\alpha)_{\alpha \leq \mu}$ is a terminal monotonic induction of O, then $x_\mu = \text{lfp}(O)$. This was shown to hold for monotonic operators over complete lattices by Denecker and Vennekens [2007], and later generalised to monotonic operators over cpo's [Vanbesien, 2025].

Theorem 2.3

Let $\langle P, \leq \rangle$ be a cpo, and $O : P \to P$ a monotonic operator. The limit of each terminal monotonic induction of O is $\text{lfp}(O)$.

Observe that, while all terminal monotonic inductions of an operator O have the same limit, they reach such limit "at different speeds", and following different constructions. The "fastest" terminal monotonic induction, so to say, is the one whose terms are obtained by the iterative application of the operator O, starting from the bottom of the cpo over which O is defined. This repeated application of the monotonic operator provides one constructive way to obtain its least fixpoint.

We conclude this section with an example which illustrates all the concepts presented so far.

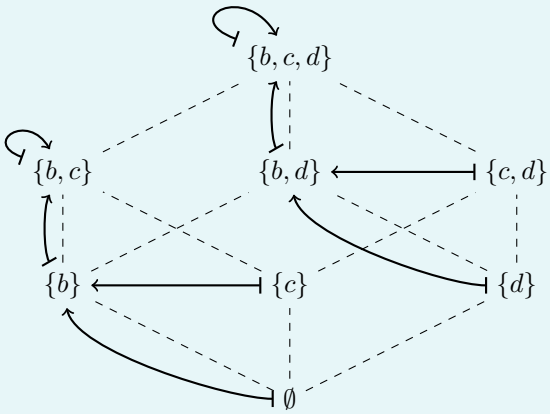
Example 2.11

Alice went on a trip to Italy and she fell in love with Italian food. She even took a cooking class with chef Bruno! Once back in Belgium, she decides to write an Italian cooking notebook and she asks Bruno to send her his favourite Italian recipes for it. To enrich her cooking notebook even more, Alice tells Bruno that anyone he trusts is allowed to contribute with more recipes. Moreover, anyone who already has the permission to add recipes to the notebook can grant permission to other people to do the same. The cooking notebook turns out so well, that Alice decides to publish it. But while collecting the recipes, she gets the impression that some people submitted recipes without being allowed to do so. To decide which recipes to publish, she needs to decide who got permission to contribute. For the sake of simplicity, we assume there are only three people other than Alice: Bruno (b), Cecilia (c), and Daniela (d). We consider the case in which Bruno grants permission to Cecilia, and Daniela grants permission to herself, in an attempt to cheat her way into the book. In the context of LP, we can encode this information with a binary predicate Permit: $\mathsf{Permit}(b, c)$ and $\mathsf{Permit}(d, d)$ hold. Then, we can write the following program $\mathfrak{P}$, defining the unary predicate Contributor in terms of Permit:

$$\mathsf{Contributor}(b)$$
$$\mathsf{Contributor}(x) \leftarrow \mathsf{Contributor}(y) \wedge \mathsf{Permit}(y, x)$$

The first rule of $\mathfrak{P}$ encodes the initial situation (base step), whereas the second one provides the induction step to construct the set of contributors.

Then, the two-valued immediate consequence operator $T^{\mathfrak{P}}$ can be described by the arrows in the Hasse diagram below. To make the picture less crammed, we use b, c, and d in place of $\mathsf{Contributor}(b)$, $\mathsf{Contributor}(c)$, and $\mathsf{Contributor}(d)$, respectively.



It is easy to check that $T^{\mathfrak{P}}$ is monotonic, and that it has two fixpoints, namely $F_1 := \{\mathsf{Contributor}(b), \mathsf{Contributor}(c)\}$ and $F_2 := \{\mathsf{Contributor}(b), \mathsf{Contributor}(c), \mathsf{Contributor}(d)\}$, of which the former is the least, as $F_1 \subseteq F_2$. This least fixpoint can also be constructed by iteratively applying $T^{\mathfrak{P}}$, starting from the bottom of the considered lattice:

$$\begin{aligned}\emptyset &\to T^{\mathfrak{P}}(\emptyset) = \{\mathsf{Contributor}(b)\} \\ &\to T^{\mathfrak{P}}(\{\mathsf{Contributor}(b)\}) = \{\mathsf{Contributor}(b), \mathsf{Contributor}(c)\} = F_1 = T^{\mathfrak{P}}(F_1).\end{aligned}$$

The obtained sequence $(\emptyset, \{\mathsf{Contributor}(b)\}, F_1)$ is a terminal monotonic induction of $T^{\mathfrak{P}}$ (the fastest one). Its limit, F_1, i.e., the least fixpoint of $T^{\mathfrak{P}}$, is the concept defined by the inductive definition that is the program $\mathfrak{P}$.

2 Approximation Fixpoint Theory

In many areas of KR, it is often fundamental to find the fixpoints of an operator, and in particular its least fixpoint.
As we have seen at the end of the previous section, Theorems 2.1 and 2.2 guarantee the existence of the least fixpoint of a monotonic operator over a complete lattice and a cpo, respectively. The condition on the monotonicity of the operator is indeed necessary for both Theorem 2.1 and 2.2 to hold, as we illustrate in the following example.

Example 2.12

(Example 2.11 continued). Alice's Italian cooking book is such an editorial success that she decides to print a second edition. This time she wants to print a shorter version, with only the recipes from the best chefs. Since she is not an expert, she trusts the chefs themselves to make a choice once again: someone who is a contributor can block other people from being one. The same permissions as in Example 2.11 are granted, but this time Daniela blocks Cecilia. We can encode this information with the predicated Permit and Block, saying that $\mathsf{Permit}(b,c)$, $\mathsf{Permit}(d,d)$, and $\mathsf{Block}(d,c)$ hold. To define the concept Contributor we use the auxiliary predicate Blocked, expressing which people are blocked. The following program $\mathfrak{P}$ defines Contributor and Blocked:

$$\mathsf{Contributor}(b)$$
$$\mathsf{Contributor}(x) \leftarrow \mathsf{Contributor}(y) \wedge \mathsf{Permit}(y,x) \wedge \neg\mathsf{Blocked}(x)$$
$$\mathsf{Blocked}(x) \leftarrow \mathsf{Contributor}(y) \wedge \mathsf{Block}(y,x)$$

Then, consider the two-valued immediate consequence operator $T^{\mathfrak{P}}$. For the sake of simplicity, we use b, c, and d in place of $\mathsf{Contributor}(b)$, $\mathsf{Contributor}(c)$, and $\mathsf{Contributor}(d)$, respectively; and $\not b, \not c$, and $\not d$ in place of $\mathsf{Blocked}(b)$, $\mathsf{Blocked}(c)$, and $\mathsf{Blocked}(d)$, respectively. It is easy to see that $T^{\mathfrak{P}}$ is not monotonic: $\{b,c\} \subseteq \{b,c,d,\not c\}$, but $T^{\mathfrak{P}}(\{b,c\}) = \{b,c\} \not\subseteq \{b,d,\not c\} = T^{\mathfrak{P}}(\{b,c,d,\not c\})$. Moreover, observe that $\{b,c\}$ and $\{b,d,\not c\}$ are both fixpoints of $T^{\mathfrak{P}}$, and they are lower bounds of the set of fixpoints of $T^{\mathfrak{P}}$. Since $\{b,c\} \not\subseteq \{b,d,\not c\}$ and $\{b,d,\not c\} \not\subseteq \{b,c\}$, $T^{\mathfrak{P}}$ has no least fixpoint.

Even when a non-monotonic operator has a least fixpoint, it may be difficult to find it. In fact, iteratively applying the operator starting from the bottom, like we did in Example 2.11 using the terminal monotonic induction principle, may not work as expected.

AFT generalizes Tarki's theory for monotonic operators by making use of a so-called *approximator*. In more detail, starting from a poset $\langle L, \leq \rangle$ and a (non-monotonic) operator $O : L \to L$, we define a poset $\mathcal{A}_L$, called an *approximation space*, and a new, monotonic operator A on $\mathcal{A}_L$, called an *approximator* (or *approximating operator*): each element of $\mathcal{A}_L$ "approximates" some elements of L, and the approximator A "approximates" the behaviour of O. By studying the fixpoints of A, we are able to obtain information on the fixpoints of O. In particular, AFT defines a few types of fixpoints (supported fixpoints, a Kripke-Kleene fixpoint, stable fixpoints, a well-founded fixpoint) of interest. In several fields of non-monotonic reasoning, it is relatively straightforward to define an approximator and it turns out that the different types of fixpoints then correspond to existing semantics.

During the years, several accounts of AFT have been developed, catering to different needs and applications [Denecker et al., 2000, 2004; Charalambidis et al., 2018; Heyninck et al., 2024; Vanbesien, 2025]. In the following subsections, we present in more detail two versions of AFT: first, the original framework, which we refer to as *standard* AFT [Denecker et al., 2000]; then, *consistent* AFT [Denecker et al., 2004], which restricts the approximation spaces taken into account.

2.1 Standard AFT

Standard AFT [Denecker et al., 2000] considers, for each complete lattice $\langle L, \leq \rangle$, its associated *square bilattice*.

Definition 2.9

The square bilattice of a complete lattice $\langle L, \leq \rangle$ is the structure $\langle L^2, \leq_{tr}, \leq_p \rangle$ where

1. $\leq_{tr}$ is the binary relation on L^2 defined by $(x_1, y_1) \leq_{tr} (x_2, y_2)$ iff $x_1 \leq x_2$ and $y_1 \leq y_2$, for all $(x_1, y_1), (x_2, y_2) \in L^2$,
2. $\leq_p$ is the binary relation on L^2 defined by $(x_1, y_1) \leq_p (x_2, y_2)$ iff $x_1 \leq x_2$ and $y_2 \leq y_1$, for all $(x_1, y_1), (x_2, y_2) \in L^2$.

The relations $\leq_{tr}$ and $\leq_p$ are called the *truth order* and the *precision order* of the square bilattice, respectively.

It can be shown easily that for a complete lattice $\langle L, \leq \rangle$, both $\leq_{tr}$ and $\leq_p$ are indeed partial orders, and that both $\langle L^2, \leq_{tr} \rangle$ and $\langle L^2, \leq_p \rangle$ are complete lattices[1].

Definition 2.10

The *approximation space* $\mathcal{A}^{\mathfrak{s}}_L$ for a complete lattice $\langle L, \leq \rangle$ is the complete lattice $\langle L^2, \leq_p \rangle$.

We will sometimes call such complete lattices $\langle L^2, \leq_p \rangle$ *square lattices*. The elements of an approximation space $\mathcal{A}^{\mathfrak{s}}_L$ can be viewed as approximations of the elements of $\langle L, \leq \rangle$: an element $(x, y) \in \mathcal{A}^{\mathfrak{s}}_L$ such that $x \leq y$ *approximates* all the values $z \in L$ such that $x \leq z \leq y$. In other words, pairs $(x, y) \in \mathcal{A}^{\mathfrak{s}}_L$ where $x \leq y$ can be thought as *intervals*. Following this analogy, stating that $(x_1, y_1) \leq_p (x_2, y_2)$, for some $(x_1, y_1), (x_2, y_2) \in \mathcal{A}^{\mathfrak{s}}_L$ such that

[1]The proof that $\langle L^2, \leq_{tr} \rangle$ is a complete lattice follows trivially from the fact that complete lattices form a Cartesian closed category. For $\langle L^2, \leq_p \rangle$ we refer to [Vanbesien, 2025].

$x_1 \leq y_1$ and $x_2 \leq y_2$, is equivalent to saying that the interval[2] $[x_1, y_1]$ contains the interval $[x_2, y_2]$. This also provides some insight on the reason why the order $\leq_p$ is called the *precision* order: if (x_2, y_2) is *more precise* than (x_1, y_1), then $[x_2, y_2] \subseteq [x_1, y_1]$. As such, all the elements approximated by (x_2, y_2) are approximated by (x_1, y_1) as well, meaning that (x_2, y_2) provides a *tighter estimate*, so to say.

Notice that this intuitive characterization of an element (x, y) in an approximation space $\mathcal{A}_L^{\mathfrak{s}}$ as the interval $[x, y]$ of approximated elements is only sensible when $x \leq y$, i.e., when $[x, y]$ is non-empty.

Definition 2.11

Let $\mathcal{A}_L^{\mathfrak{s}}$ be an approximation space for $\langle L, \leq \rangle$. An element $(x, y) \in \mathcal{A}_L^{\mathfrak{s}}$ is called

- *consistent* if $x \leq y$;
- *exact* if $x = y$;
- *inconsistent* if (x, y) is not consistent.

Example 2.13: Logic Programming

(Example 2.10 continued). First, notice that when we move to approximations in standard AFT, we obtain four pairs of truth values: $(\mathbf{f}, \mathbf{f})$, $(\mathbf{t}, \mathbf{t})$, $(\mathbf{f}, \mathbf{t})$, and $(\mathbf{t}, \mathbf{f})$. The former three are consistent, whereas the last one is inconsistent. It turns out that these four pairs match the four truth values of four-valued logic: $(\mathbf{f}, \mathbf{f})$ and $(\mathbf{t}, \mathbf{t})$ are exact and represent the values $\mathbf{f}$ and $\mathbf{t}$, respectively; $(\mathbf{f}, \mathbf{t})$ approximates both $\mathbf{f}$ and $\mathbf{t}$ and corresponds to *unkown* (denoted as $\mathbf{u}$); finally $(\mathbf{t}, \mathbf{f})$ represents inconsistency and matches with the truth value *inconsistent* (denoted as $\mathbf{i}$). Notice that the three consistent pairs correspond to the three truth values of three-valued logic. Moreover, the two orders on $\{(\mathbf{f}, \mathbf{f}), (\mathbf{t}, \mathbf{t}), (\mathbf{f}, \mathbf{t}), (\mathbf{t}, \mathbf{f})\}$ from Definition 2.9 induced by $\leq_o$ coincide, under the correspondence just presented, with the standard definitions of truth order and precision order on $\{\mathbf{f}, \mathbf{t}, \mathbf{u}, \mathbf{i}\}$ that we find in four-valued (and three-valued) logic. Hence, we will use the pairs from $\{(\mathbf{f}, \mathbf{f}), (\mathbf{t}, \mathbf{t}), (\mathbf{f}, \mathbf{t}), (\mathbf{t}, \mathbf{f})\}$ and the truth values in $\{\mathbf{f}, \mathbf{t}, \mathbf{u}, \mathbf{i}\}$ interchangeably.

We saw that, given a propositional vocabulary Σ, the two-valued approximations form a complete lattice 2^{Σ}. An element $\mathcal{I} = (I, J)$ of $\mathcal{A}_{2^{\Sigma}}^{\mathfrak{s}}$ can be seen as a *four-valued interpretation*: I represents what is *true* and J what is *possible*. A fact is then *true* (resp. *false*) if it is true (resp. false) in both I and J, *unknown* if it is true in J but not

[2]We adopt the standard notation "$[\cdot, \cdot]$" for closed intervals, i.e., $[x, y] := \{z \mid x \leq z \leq y\}$.

in I, and *inconsistent* if it is true in I but not in J. The four-valued interpretation (I, J) approximates all two-valued interpretations I' such that $I \subseteq I' \subseteq J$.

After having defined the approximation spaces, standard AFT considers as *approximators* a specific class of operators. In the following, we use the standard notation for the components of a function: given an operator $A : L^2 \to L^2$, we denote by $A_1 : L^2 \to L$ and $A_2 : L^2 \to L$ the *first* and the *second component* of A, respectively, i.e., $A(x, y) = (A_1(x, y), A_2(x, y))$ for all $(x, y) \in L^2$.

Definition 2.12

Let $\langle L, \leq \rangle$ be a complete lattice. An *approximator* $A : \mathcal{A}_L^s \to \mathcal{A}_L^s$ is a monotonic operator that is *symmetric*, i.e., for all $(x, y) \in \mathcal{A}_L^s$ it holds that $A_1(x, y) = A_2(y, x)$.

Definition 2.13

Let $\langle L, \leq \rangle$ be a complete lattice. An approximator $A : \mathcal{A}_L^s \to \mathcal{A}_L^s$ *approximates* an operator $O : L \to L$ if for all $x \in L$, $A(x, x) = (O(x), O(x))$. In this case, we say that A is an *approximator of* O.

Once an approximator A of an operator $O : L \to L$ is defined, we can gain information about the fixpoints of O by studying the fixpoints of A, and other operators derived from A:

- ***Supported fixpoints.*** The exact fixpoints of A are called the *supported fixpoints* (of A). Notice that, by Definition 2.13, $A(x, x) = (O(x), O(x))$ for all $x \in L$. Hence, for any $x \in L$, it holds that (x, x) is a fixpoint of A (i.e., a supported fixpoint) if and only if x is a fixpoint of O.
- ***Kripke-Kleene fixpoint.*** Since A is monotonic by definition, and $\mathcal{A}_L^s$ is a complete lattice for any complete lattice $\langle L, \leq \rangle$, by the Knaster-Tarski's theorem (Theorem 2.1) A has a least fixpoint. We call such least fixpoint the *Kripke-Kleene* fixpoint (of A). Notice that the Kripke-Kleene fixpoint approximates all supported fixpoints, i.e., all fixpoints of O.
- ***Well-founded fixpoint.*** Starting from A, we can define a new operator $\mathcal{S}_A$ over $\mathcal{A}_L^s$, called the *stable revision operator*, which takes a pair $(x, y) \in \mathcal{A}_L^s$ and sends it to $\mathcal{S}_A(x, y) := (\mathrm{lfp}(A_1(\cdot, y)), \mathrm{lfp}(A_2(x, \cdot))) \in \mathcal{A}_L^s$, where, for each $z \in L$, $A_1(\cdot, z) : L \to L$ (respectively, $A_2(z, \cdot) : L \to L$) is the operator sending w to $A_1(w, z)$ (respectively, $A_2(z, w)$) for all $w \in L$. Denecker et al. [2000] showed that, for each approximator A, $\mathcal{S}_A$ is well-defined and monotonic. Hence, by the Knaster-Tarski's theorem (Theorem 2.1), $\mathcal{S}_A$ has a least fixpoint. We call such least fixpoint the *well-founded* fixpoint (of A).

- ***Stable fixpoints.*** We call the *stable fixpoints* (of A) the exact fixpoints of S_A. Since the well-founded fixpoint is the least fixpoint of S_A, it follows that the well-founded fixpoint approximates all stable fixpoints.

Example 2.14: Logic Programming

(Example 2.13 continued). In the case of LP, we can take as approximator of the two-valued immediate consequence operator $T^{\mathfrak{P}}$ nothing else than Fitting's (2002) *four-valued immediate consequence operator* $\Psi^{\mathfrak{P}}$. This operator sends a four-valued interpretation $\mathcal{I}$ to $\Psi^{\mathfrak{P}}(\mathcal{I}) = (\Psi_1^{\mathfrak{P}}(\mathcal{I}), \Psi_2^{\mathfrak{P}}(\mathcal{I}))$ with

$$\Psi_1^{\mathfrak{P}}(\mathcal{I}) := \{p \mid \exists p \leftarrow \varphi \in \mathfrak{P} \text{ s.t. } \varphi^{\mathcal{I}} \in \{\mathbf{t}, \mathbf{i}\}\}$$
$$\Psi_2^{\mathfrak{P}}(\mathcal{I}) := \{p \mid \exists p \leftarrow \varphi \in \mathfrak{P} \text{ s.t. } \varphi^{\mathcal{I}} \in \{\mathbf{t}, \mathbf{u}\}\},$$

where the body of each rule is evaluated using Belnap's truth tables [Belnap, 1977]. For this approximator, the fixpoints defined by AFT correspond to the major homonymous semantics of LP [Denecker et al., 2000, 2012]: supported fixpoints are models of Clark's completion [Clark, 1977], stable fixpoints correspond to (partial) stable models [Gelfond and Lifschitz, 1988], the Kripke-Kleene fixpoint to the Kripke-Kleene model [Fitting, 1985] and the well-founded fixpoint is the well-founded model [Van Gelder et al., 1991].

Alternatively to what stated above, the Kripke-Kleene fixpoint of an approximator A can also be obtained constructively, via a monotonic induction of A, as described in Section 1.1. On the other hand, the well-founded fixpoint of A can be derived using another constructive process, namely a *well-founded induction* of A [Denecker and Vennekens, 2007]. While subsequent elements in the sequence of a monotonic induction are governed by Item 2 of Definition 2.8, for well-founded inductions of A we consider *A-refinements*. These are of two types: *application refinements* and *unfoundedness refinements*. While the formers are bounded by the application of A, the latters correspond to the elimination of an unfounded set.

Definition 2.14

Let $\langle L, \leq \rangle$ be a complete lattice, $(x, y) \in \mathcal{A}_L^{\mathfrak{s}}$, and A be an approximator over $\mathcal{A}_L^{\mathfrak{s}}$. A pair $(u, v) \in \mathcal{A}_L^{\mathfrak{s}}$ is:

- an *application A-refinement* of (x, y) if $(x, y) \leq_p (u, v) \leq_p A(x, y)$;
- an *unfoundedness A-refinement* of (x, y) if $x = u$ and $A_2(x, v) \leq v \leq y$;
- an *A-refinement* of (x, y) if it is either an application A-refinement of (x, y) or an unfoundedness A-refinement of (x, y).

Definition 2.15

Let $\langle L, \leq \rangle$ be a complete lattice, and A be an approximator over $\mathcal{A}_L^s$. A *well-founded induction of* A is a (possibly transfinite) sequence $(x_\alpha, y_\alpha)_{\alpha \leq \mu}$ of elements of $\mathcal{A}_L^s$ such that

1. $(x_0, y_0) = \bot_{\mathcal{A}_L^s}$;
2. $(x_{\alpha+1}, y_{\alpha+1})$ is an A-refinement of (x_α, y_α) for every $\alpha < \mu$;
3. $(x_\lambda, y_\lambda) = \mathrm{lub}_{\leq_p}(\{(x_\alpha, y_\alpha) \mid \alpha < \lambda\})$ for every limit ordinal $\lambda \leq \mu$.

A well-founded induction $(x_\alpha, y_\alpha)_{\alpha \leq \mu}$ is *terminal* if it cannot be extended, i.e., there is no A-refinement $(x_{\mu+1}, y_{\mu+1})$ of (x_μ, y_μ) strictly more precise than (x_μ, y_μ).

As for monotonic inductions of a monotonic operator, there may be multiple well-founded induction for an approximator A: Denecker and Vennekens [2007] proved that all such well-founded inductions have the same limit and this coincides with the well-founded fixpoint of A.

2.2 Consistent AFT

Denecker et al. [2004] showed that the same constructions and fixpoints obtained by standard AFT can be produced by considering a different class of approximation spaces, namely those comprising only the consistent elements; hence the name of this AFT account.

Definition 2.16

The *(consistent) approximation space* $\mathcal{A}_L^c$ for a complete lattice $\langle L, \leq \rangle$ is the subposet of $\mathcal{A}_L^s$ with underlying set $\{(x, y) \mid x, y \in L, x \leq y\}$.

It is not hard to show that $\mathcal{A}_L^c$ is a cpo for each complete lattice L [Denecker et al., 2004]. Notice that the lack of inconsistent elements allows for an even more sensible interpretation of the elements of an approximation space, as now they can all be viewed as intervals approximating the elements they contain. Since we do not have any inconsistent element, the symmetry condition on approximators does not make much sense anymore. Hence, we only ask for the approximators to be monotonic and to send any exact element to an exact element.

Example 2.15: Logic Programming

(Example 2.14 continued). For consistent AFT, the approximation space for two-valued interpretations is the cpo of pairs (I, J) of two-valued interpretations with $I \subseteq J$. One of such pairs is called a *three-valued interpretation*: it sends an element of the vocabulary to either **t**, **f**, or **u**. The approximator of choice, in this case, is Fitting's (2002) *three-valued immediate consequence operator*, which uses Kleene's strong three-valued logic [Kleene, 1952] to evaluate the body of rules.

Given an approximator $A : \mathcal{A}_L^c \to \mathcal{A}_L^c$, the stable revision operator can be defined as for standard AFT, however, its monotonicity is not guaranteed in general. Nevertheless, Denecker et al. [2004] showed that the stable revision operator is well-defined and monotonic on the cpo formed by the A-reliable and A-prudent pairs of $\mathcal{A}_L^c$.

Definition 2.17

Given an approximator $A : \mathcal{A}_L^c \to \mathcal{A}_L^c$, a pair $(x, y) \in \mathcal{A}_L^c$ is

- *A-reliable* if $(x, y) \leq_p A(x, y)$;
- *A-prudent* if $x \leq \mathrm{lfp}(A_1(\cdot, y))$.

Then, the four classes of fixpoints presented for standard AFT can be obtained for consistent AFT as well, in an analogous way.

The reader may now wonder whether there is a correspondence between the fixpoints obtained via standard and consistent AFT. Given an approximator $A : \mathcal{A}_L^c \to \mathcal{A}_L^c$ of $O : L \to L$, there may be multiple approximators extending[3] A to $\mathcal{A}_L^s$ (Theorem 4.3 in [Denecker et al., 2004]). Conversely, since any symmetric approximator over $\mathcal{A}_L^s$ is internal in $\mathcal{A}_L^c$ (i.e., it sends consistent elements to consistent elements), any approximator over $\mathcal{A}_L^s$ can be restricted[3] to obtain an approximator over $\mathcal{A}_L^c$ for the same operator $O : L \to L$. Denecker et al. [2004] proved that the consistent fixpoints of interest (Kripke-Kleene, supported, stable, and well-founded) of an approximator A over $\mathcal{A}_L^c$ are the same as the homonymous ones of any extension of A to $\mathcal{A}_L^s$.

Notice that, in general, there may be multiple approximators $A : \mathcal{A}_L^c \to \mathcal{A}_L^c$ of an operator $O : L \to L$. We can endow the set of all such approximators with the point-wise order $\leq_p$ induced by the precision order on $\mathcal{A}_L^c$: for all approximators $A : \mathcal{A}_L^c \to \mathcal{A}_L^c$ and $B : \mathcal{A}_L^c \to \mathcal{A}_L^c$ of $O : L \to L$, $A \leq_p B$ if and only if $A(x, y) \leq_p B(x, y)$ for all $(x, y) \in \mathcal{A}_L^c$. The $\leq_p$-maximal approximator of O is called an *ultimate approximator* of O, and it maps

[3] We use the standard notions of extention and restriction of a function. Let $f : X \to Y$ and $g : X' \to Y'$ be two functions, with $X \subseteq X'$. We say that f is the *restriction* of g to X and that g is an *extension* of f to X' if $f(x) = g(x)$ for all $x \in X$.

any tuple $(x, y) \in \mathcal{A}_L^{\mathsf{c}}$ to the pair $(\mathrm{glb}(O[x, y]), \mathrm{lub}(O[x, y]))$, where $O[x, y] := \{O(z) \mid z \in L \text{ s.t. } x \leq z \leq y\}$. We denote the ultimate approximator of an operator $O : L \to L$ by $\mathcal{U}_O$. Any exact stable fixpoint of an approximator $A : \mathcal{A}_L^{\mathsf{c}} \to \mathcal{A}_L^{\mathsf{c}}$ of an operator $O : L \to L$ is also a stable fixpoint of $\mathcal{U}_O$, and the well-founded fixpoint of $\mathcal{U}_O$ is more precise than the well-founded fixpoint of any approximator $A : \mathcal{A}_L^{\mathsf{c}} \to \mathcal{A}_L^{\mathsf{c}}$ of O.

3 Fixpoint Semantics for DatalogMTL with Negation

This chapter is based on the conference paper *Fixpoint Semantics for DatalogMTL with Negation* [Pollaci, 2025]. This work has been presented at the 41st International Conference on Logic Programming (ICLP) in Rende, Italy, and published in its proceedings. Part of the preliminaries of [Pollaci, 2025], were left out from this chapter since they are already covered by the content of Chapter 2. Moreover, the notation of [Pollaci, 2025] has been adapted in compliance with the one previously introduced in Chapter 2.

1 Introduction

Datalog [Abiteboul et al., 1995] is a declarative query language that extends relational algebra with recursion. The last few decades have witnessed the emergence of several variants of Datalog and their implementations [Ketsman and Koutris, 2022], aiming to solve data management tasks in various applications [Huang et al., 2011]. One of such variants is DatalogMTL [Brandt et al., 2018], which extends the Datalog language with operators from MTL [Koymans, 1990], and finds applications in ontology-based query answering [Brandt et al., 2018; Kalayci et al., 2018; Kikot et al., 2018; Koopmann, 2019], stream reasoning [Walega et al., 2019, 2023], and reasoning for the financial sector [Nissl and Sallinger, 2022; Mori et al., 2022; Colombo et al., 2023], amongst others.

For instance, we can use the following DatalogMTL rule to express that an adult has to wait at least 6 hours after the assumption of a paracetamol tablet, before taking another one:

$$\mathsf{NoMoreParacetamol}(x) \leftarrow \mathsf{Adult}(x) \wedge \diamondminus_{[0,6]} \mathsf{TakesParacetamol}(x),$$

where $\diamondminus_{[0,6]}\,\mathsf{TakesParacetamol}(x)$ is true at a timepoint t, if $\mathsf{TakesParacetamol}(x)$ was true at some point in the last 6 hours, i.e., at some time t' in the interval $[t-6, t]$. Other than $\diamondminus_{[0,6]}$, there are a few more metric temporal operators that are allowed in the expressions of DatalogMTL. If φ and ψ are two formulas and $[t_1, t_2]$ is an interval on the rational or integer timeline, then $\boxminus_{[t_1,t_2]}\varphi$ holds at time t if φ holds at all the timepoints of the interval $[t-t_2, t-t_1]$. Analogously, there are also the future-oriented counterparts of $\diamondminus$ and $\boxminus$, namely $\diamondplus$ and $\boxplus$, respectively: $\diamondplus_{[t_1,t_2]}\,\varphi$ holds at time t if φ holds at some timepoint of the interval $[t+t_1, t+t_2]$; and $\boxplus_{[t_1,t_2]}\varphi$ holds at time t if φ holds at all timepoints of the interval $[t+t_1, t+t_2]$. Finally, $\varphi\,\mathsf{S}_{[t_1,t_2]}\,\psi$ ("φ *since* ψ") holds at a time t if there exists a time $t' \in [t-t_2, t-t_1]$ at which ψ holds, and φ holds *since* then, i.e., φ holds at all the timepoints in (t', t); whereas $\varphi\,\mathsf{U}_{[t_1,t_2]}\,\psi$ ("φ *until* ψ") holds at a time t if there exists a time $t' \in [t+t_1, t+t_2]$ at which ψ holds, and φ holds *until* then, i.e., φ holds at all timepoints in (t, t'). A (DatalogMTL) *dataset* is then a finite set of facts, each paired with a time interval. For example, our dataset could include the fact $\mathsf{TakesParacetamol}(John)@[8,8]$, meaning that *John* took a paracetamol tablet at 8a.m (or, more precisely, at every timepoint of the interval $[8,8]$).

In 2021, DatalogMTL was extended with stratified negation as failure [Cucala et al., 2021], allowing negated atoms in the body of rules, but no recursion. Recently, Walega et al. [2024] further expanded the range of capabilities of DatalogMTL by equipping it with unstratifiable negation under the stable model semantics (DatalogMTL$^{\neg}$), extending, in this way, both DatalogMTL with stratified negation and Datalog with stable model negation. To obtain the stable models of a dataset and a program, Walega et al. [2024] employed interpretations similar to the ones used for the HT logic [Heyting, 1930; Pearce, 1996], namely pairs (I, J) of (two-valued) interpretations such that the atoms evaluated as true by I are true under J as well. Although used differently, this same framework with pairs of interpretations sits at the core of consistent AFT [Denecker et al., 2004], as introduced in Section 2.2 of Chapter 2: a pair (I, J) of two-valued interpretations with $I \subseteq J$ is viewed as a three-valued interpretation, approximating all the two-valued interpretations I' such that $I \subseteq I' \subseteq J$. This chapter demonstrates that consistent AFT is suitable to define a fixpoint semantics for DatalogMTL$^{\neg}$, showcasing once again the power and versatility of the AFT formalism.

We start by considering the set of pairs of two-valued interpretations already considered by Walega et al. [2024], we define an operator on it, and we prove that such operator is a consistent approximator. Then, consistent AFT allows us to obtain not only the *stable*

model semantics for DatalogMTL$^\neg$, but also the *well-founded*, the *Kripke-Kleene*, and the *supported* model semantics as well. Finally, we prove that the stable models obtained via the usage of AFT coincides with the stable models defined by Walega et al. [2024]. The advantages of employing the framework of AFT may be summarized as follows:

1. *Effortless.* The only thing required, is defining the right *approximator*; then the machinery of AFT lays out the stable, the well-founded, the Kripke-Kleene, and the supported model semantics for free. For most applications, the choice of an approximator is very natural; for DatalogMTL$^\neg$, it is no exception. The approximator, in fact, closely resembles Fitting's three-valued immediate consequence operator [Fitting, 2002], with some slight adjustments to account for time points and datasets.

2. *Solid mathematical background.* AFT provides a broad range of properties and general results that hold for all its applications; hence, there is no need to re-prove several statements when switching settings. For instance, the minimality and uniqueness result (Theorem 3.4 in [Walega et al., 2024]) leading to the definition of stable models for DatalogMTL$^\neg$ in [Walega et al., 2024], is already guaranteed by the AFT framework.

3. *Reliable.* The strong mathematical foundations of AFT make the obtained results more reliable. Even when semantics have already been defined, as in the case of the stable model semantics for DatalogMTL$^\neg$, AFT constitutes a valuable tool to screen the body of previous work for errors. For example, the employment of AFT in the context of Abstract Dialectical Frameworks allowed the detection of some bugs in the original semantics [Bogaerts, 2019].

4. *New Perspective.* The well-founded semantics obtained via AFT formally models the principle of (inductive) definitions. Hence, the well-founded semantics defined in this chapter provides a valuable insight and a novel perspective: DatalogMTL$^\neg$ can be viewed as a formal modelling of a particular class of inductive definitions with temporal conditions in the rules.

Outline of the Chapter. First, we provide some preliminary material[1] on the syntax and semantics of DatalogMTL$^\neg$ (Section 2). In Section 3, we define the base notions we need to apply AFT on DatalogMTL$^\neg$: the three-valued semantics, and the immediate consequence operators (two- and three-valued). Moreover, we prove there exists a correspondence between the pre-fixpoints of these operators, and the two-valued and three-valued models. In Section 4, we are finally able to apply AFT; we just need to prove that the three-valued immediate consequence operator defined in the previous section is indeed a consistent

[1] For preliminaries on AFT we refer back to Section 2 of Chapter 2.

approximator. Then, we are able to define the Kripke-Kleene, stable, well-founded, and supported models via AFT. Finally, in Section 5, we compare our notions of three-valued models and two-valued stable models, with the concepts from [Walega et al., 2024], and we show that the stable models obtained via AFT are the same as the ones from [Walega et al., 2024].

2 Preliminaries

In this section, we introduce the basic definitions and notation relative to timelines, the syntax of DatalogMTL$^\neg$, and the semantics of rule bodies, as they were presented by Walega et al. [2024].

A *timeline* $\mathcal{T}$ is either the ring of integers $\langle \mathbb{Z}, +, \cdot \rangle$, or the field of rationals $\langle \mathbb{Q}, +, \cdot \rangle$, equipped with the respective standard orders. A ($\mathcal{T}$-)*interval* δ is a non-empty subset of $\mathcal{T}$ such that the following conditions are satisfied:

1. δ is *convex*, i.e., if $t_1, t_2 \in \delta$, then $t \in \delta$ for all $t \in \mathcal{T}$ such that $t_1 < t < t_2$;
2. $\mathrm{glb}(\delta), \mathrm{lub}(\delta) \in \mathcal{T} \cup \{+\infty, -\infty\}$.

We call an interval δ *non-negative* if for all $t \in \delta$, it holds that $t \geq 0$. We represent an interval δ with the standard notation $\langle \mathrm{glb}(\delta), \mathrm{lub}(\delta) \rangle$, where the left bracket $\langle$ can either be [, if $\mathrm{glb}(\delta) \in \delta$, or (otherwise; and the right bracket $\rangle$ can either be], if $\mathrm{lub}(\delta) \in \delta$, or) otherwise. For any $t \in \mathcal{T}$, we denote the interval $[t, t]$ by simply t.

We now consider a function-free first-order vocabulary and a timeline $\mathcal{T}$. For any predicate P, and tuple $\mathbf{s}$ of constants and variables of the same arity of P, we call $\mathsf{P}(\mathbf{s})$ a *relational atom*. A *metric atom* is an expression given by

$$M ::= \top \mid \bot \mid \mathsf{P}(\mathbf{s}) \mid \diamondsuit_\delta M \mid \boxdot_\delta M \mid \boxminus_\delta M \mid \boxplus_\delta M \mid M \,\mathcal{S}_\delta\, M \mid M \,\mathcal{U}_\delta\, M,$$

where $\mathsf{P}(\mathbf{s})$ ranges over relational atoms, and δ ranges over non-negative intervals. A metric atom is *ground* if it does not contain any variable. A *head atom* is a metric atom specified by the grammar:

$$M ::= \top \mid \mathsf{P}(\mathbf{s}) \mid \boxminus_\delta M \mid \boxplus_\delta M, \tag{3.1}$$

where $\mathsf{P}(\mathbf{s})$ ranges over relational atom, and δ ranges over non-negative intervals. We denote by $\mathcal{M}$ the set of ground head atoms. A *metric fact* is an expression $M@\delta$, where M is a ground metric atom, and δ is an interval. A metric fact $M@\delta$ is also a *relational fact* if M is a ground relational atom. A *dataset* is a finite set of relational facts. A *(two-valued)*

interpretation I for DatalogMTL is a function which assigns to each $t \in \mathfrak{T}$ a set $I(t)$ of ground relational atoms. A two-valued interpretation I is a *(two-valued) model of a dataset* $\mathcal{D}$ if for all relational facts $\mathsf{P}(\mathbf{s})@\delta \in \mathcal{D}$, and for all $t \in \delta$, it holds that $\mathsf{P}(\mathbf{s}) \in I(t)$. Similarly to what happens for LP (Example 2.9 in Chapter 2), interpretations can be ordered with the point-wise order induced by the subset relation, i.e., $I \subseteq J$ if and only if for all $t \in \mathfrak{T}$, $I(t) \subseteq J(t)$, and it is easy to prove that the set of all two-valued interpretations together with the order given by the subset relation is a complete lattice. We denote such a lattice by $\mathfrak{I}$. A *rule* r is an expression of the form

$$M \leftarrow M_1 \wedge \ldots \wedge M_k \wedge \neg M_{k+1} \wedge \ldots \wedge \neg M_m, \tag{3.2}$$

where $m \geq k \geq 0$, each M_i is a metric atom, and M is a head atom. We call the atoms $M_1, \ldots, M_k$ *positive*, and $M_{k+1}, \ldots, M_m$ *negated*. The conjunction of positive and negated atoms of a rule is called its *body*, whereas the consequent M is its *head*. For brevity reasons, we will often denote a rule as $M \leftarrow B$, where B represents the conjunction in the body. A rule is *safe* if each variable it mentions in the head occurs in some positive body atom in a position other than a left operand of $\mathcal{S}$ or $\mathcal{U}$. A (DatalogMTL$^\neg$) *program* is a finite set of safe rules. A rule is *ground* if it has no variables. For a program $\mathcal{P}$, we denote by $\mathsf{ground}(\mathcal{P})$ the set of all the ground rules that can possibly be obtained from rules in $\mathcal{P}$ by replacing variables with constants.

Example 3.1

Belgian identity cards (IDs) last for ten years, after which they need to be renewed. Let NewID be a predicate of arity 1: the relational atom $\mathsf{NewID}(x)$ is meant to indicate that a person x gets a new ID, either by renewing an old one or by creating a new one from scratch. We can then write a very simple program $\mathcal{P}$ with just one rule:

$$\boxplus_{10}\mathsf{NewID}(x) \leftarrow \mathsf{NewID}(x) \wedge \neg \diamondplus_{(0,10)} \mathsf{NewID}(x).$$

The rule above should be interpreted as "*x gets a new identity card ten years from time t if x gets a new identity card at time t and x does not get a new identity card at any time between t and ten years from t*", for any time $t \in \mathfrak{T}$. Notice that, because of the recursion over negation, the program $\mathcal{P}$ is unstratifiable.

Now suppose Linde is a Belgian citizen who got her first identity card in 2009. Unfortunately, she lost her card in Milan right before her flight back to Belgium. This happened in 2014. We can encode this information in the following dataset using relational facts with the constant *Linde*:

$$\mathcal{D} := \{\mathsf{NewID}(\mathit{Linde})@2009, \mathsf{NewID}(\mathit{Linde})@2014\}.$$

Observe that, with this dataset, the only ground rule we shall consider is the one obtained by replacing the variable x with the constant *Linde* in the only rule of $\mathcal{P}$, i.e., $\boxplus_{10}\mathsf{NewID}(\mathit{Linde}) \leftarrow \mathsf{NewID}(\mathit{Linde}) \wedge \neg \diamondplus_{(0,10)} \mathsf{NewID}(\mathit{Linde})$.

We conclude the preliminaries with the definitions of the *two-valued semantics* of DatalogMTL$^\neg$, and the *two valued models* of a dataset and a program.

Definition 3.1

Let $\mathcal{P}$ be a program, I an interpretation, and $t \in \mathcal{T}$. The *two-valued semantics* of ground metric atoms and bodies is defined as follows:

1. $[\![\top]\!](I,t) = \mathbf{t}$,
2. $[\![\bot]\!](I,t) = \mathbf{f}$,
3. $[\![\mathsf{P}(\mathbf{s})]\!](I,t) = \begin{cases} \mathbf{t}, & \text{if } \mathsf{P}(\mathbf{s}) \in I(t) \\ \mathbf{f}, & \text{otherwise} \end{cases}$,
4. $[\![\diamondminus_\delta M]\!](I,t) = \mathrm{lub}_{\leq_o}(\{[\![M]\!](I,t') \mid t' \in \mathcal{T} \text{ such that } t - t' \in \delta\})$,
5. $[\![\diamondplus_\delta M]\!](I,t) = \mathrm{lub}_{\leq_o}(\{[\![M]\!](I,t') \mid t' \in \mathcal{T} \text{ such that } t' - t \in \delta\})$,
6. $[\![\boxminus_\delta M]\!](I,t) = \mathrm{glb}_{\leq_o}(\{[\![M]\!](I,t') \mid t' \in \mathcal{T} \text{ such that } t - t' \in \delta\})$,
7. $[\![\boxplus_\delta M]\!](I,t) = \mathrm{glb}_{\leq_o}(\{[\![M]\!](I,t') \mid t' \in \mathcal{T} \text{ such that } t' - t \in \delta\})$,
8. $[\![M_1 \,\mathcal{S}_\delta\, M_2]\!](I,t) = \begin{cases} \mathbf{t}, & \text{if } [\![M_2]\!](I,t') = \mathbf{t} \text{ for some } t' \text{ with } t - t' \in \delta \\ & \text{and } [\![M_1]\!](I,t'') = \mathbf{t} \text{ for all } t'' \in (t',t) \\ \mathbf{f}, & \text{otherwise} \end{cases}$,
9. $[\![M_1 \,\mathcal{U}_\delta\, M_2]\!](I,t) = \begin{cases} \mathbf{t}, & \text{if } [\![M_2]\!](I,t') = \mathbf{t} \text{ for some } t' \text{ with } t' - t \in \delta \\ & \text{and } [\![M_1]\!](I,t'') = \mathbf{t} \text{ for all } t'' \in (t,t') \\ \mathbf{f}, & \text{otherwise} \end{cases}$,
10. $[\![\neg M]\!](I,t) = [\![M]\!](I,t)^{-1}$,
11. $[\![E_1 \wedge \ldots \wedge E_m]\!](I,t) = \mathrm{glb}_{\leq_o}(\{[\![E_1]\!](I,t), \ldots, [\![E_m]\!](I,t)\})$,

where we define $\mathbf{f}^{-1} = \mathbf{t}$ and $\mathbf{t}^{-1} = \mathbf{f}$, and $\leq_o$ is the order given by $\mathbf{f} \leq_o \mathbf{t}$.

Definition 3.2

Let $\mathcal{P}$ be a program, $\mathcal{D}$ a dataset, and $I \in \mathcal{I}$ an interpretation. Then, I is a *(two-valued) model* of $\mathcal{P}$ iff for every rule $M \leftarrow B \in \mathsf{ground}(\mathcal{P})$, and for every $t \in \mathcal{T}$, $[\![B]\!](I,t) \leq_o [\![M]\!](I,t)$. I is a *(two-valued) model* of $\mathcal{P}$ and $\mathcal{D}$ iff I is a two-valued model of $\mathcal{P}$ and a model of $\mathcal{D}$.

3 The Three-Valued Semantics

In order to apply the machinery of AFT, we just need to define an approximation space and a suitable approximator. This turns out to be fairly straightforward: the approximation space is the space of *three-valued interpretations*, and the approximator is an adaptation of Fitting's *three valued immediate consequence operator* [Fitting, 2002]. In this section, we first define the three-valued interpretations and the three-valued semantics of rule bodies. Then, we define the two-valued, and the three-valued immediate consequence operators, and we show they characterize the two-valued, and three valued models, respectively.

Defining the three-valued semantics of rule bodies can easily be done by using three-valued interpretations. A *three-valued interpretation* $\boldsymbol{\mathcal{I}}$ associates to each $t \in \mathcal{T}$ a function $\boldsymbol{\mathcal{I}}(t)$ taking a ground relational atom and returning either $\mathbf{t}$, $\mathbf{f}$, or $\mathbf{u}$. For the sake of clarity, we will use the calligraphic font to differentiate three-valued interpretations from two-valued ones. Analogously to what happens for interpretations in LP (see Examples 2.13 and 2.15), a three-valued interpretation may also be regarded as a pair of two-valued interpretations (I,J), where $I,J \in \mathcal{I}$ such that $I \subseteq J$. Recall that then the truth values $\mathbf{t}$, $\mathbf{f}$, and $\mathbf{u}$ correspond to the pairs $(\mathbf{t},\mathbf{t})$, $(\mathbf{f},\mathbf{f})$, and $(\mathbf{f},\mathbf{t})$, respectively. For a three-valued interpretation $\boldsymbol{\mathcal{I}}$, we will use the notation I_1 and I_2 for the two two-valued interpretations forming the pair $\boldsymbol{\mathcal{I}}$ corresponds to[2], i.e., $\boldsymbol{\mathcal{I}} \mathrel{\hat{=}} (I_1, I_2)$. Recall that we can define two orders on three-valued interpretations: the *truth order* $\leq_{tr}$, i.e., $\boldsymbol{\mathcal{I}} \leq_{tr} \boldsymbol{\mathcal{J}}$ if and only if $I_1 \subseteq J_1$ and $I_2 \subseteq J_2$; and the *precision order* $\leq_p$, i.e., $\boldsymbol{\mathcal{I}} \leq_p \boldsymbol{\mathcal{J}}$ if and only if $I_1 \subseteq J_1$ and $J_2 \subseteq I_2$. We denote by $\mathcal{A}^c_{\mathcal{I}}$ the set of three-valued interpretations[3].

We can now define the three-valued semantics straightforwardly. Notice that we employ the same notation used for the two-valued semantics in Definition 3.1; the two semantics

[2] If we regard a three-valued interpretation $\boldsymbol{\mathcal{I}}$ as a function, then the two-valued interpretations forming the pair corresponding to $\boldsymbol{\mathcal{I}}$, that we denoted by I_1 and I_2, are just the components of $\boldsymbol{\mathcal{I}}$, i.e., $\boldsymbol{\mathcal{I}}_1$ and $\boldsymbol{\mathcal{I}}_2$, respectively, in accordance to the notation introduced in Section 2.1 of Chapter 2. Nevertheless, we adopt the notation I_1 and I_2 in this chapter, instead of the calligraphic one, to make it easier to discern three-valued interpretations from two-valued ones, just by looking at the font used.

[3] This is in agreement with the notation introduced in Section 2.2 of Chapter 2: the set of three-valued interpretations is indeed the underlying set of the (consistent) approximation space for $\mathcal{I}$, as we argue in Section 4 of this chapter.

are easy to distinguish by looking at the interpretation (two- or three-valued) in the argument.

Definition 3.3

Let $\mathcal{P}$ be a program, $\boldsymbol{\mathcal{I}} = (I_1, I_2)$ a three-valued interpretation, and $t \in \mathcal{T}$. The *three-valued semantics* of ground metric atoms and bodies is defined as follows:

1. $[\![M]\!](\boldsymbol{\mathcal{I}}, t) = ([\![M]\!](I_1, t), [\![M]\!](I_2, t))$,
2. $[\![\neg M]\!](\boldsymbol{\mathcal{I}}, t) = [\![M]\!](\boldsymbol{\mathcal{I}}, t)^{-1}$,
3. $[\![E_1 \wedge \ldots \wedge E_m]\!](\boldsymbol{\mathcal{I}}, t) = \mathrm{glb}_{\leq_{tr}}(\{[\![E_1]\!](\boldsymbol{\mathcal{I}}, t), \ldots, [\![E_m]\!](\boldsymbol{\mathcal{I}}, t)\})$,

where we define $\mathbf{f}^{-1} = \mathbf{t}$, $\mathbf{u}^{-1} = \mathbf{u}$, and $\mathbf{t}^{-1} = \mathbf{f}$, and $\leq_{tr}$ is the (three-valued) *truth order*, i.e., the order defined by $\mathbf{f} \leq_{tr} \mathbf{u} \leq_{tr} \mathbf{t}$.

Remark 3.1

In Definition 3.3, if $\boldsymbol{\mathcal{I}} = (I, I)$, then it holds that

$$\begin{aligned}[\![E_1 \wedge \ldots \wedge E_m]\!](\boldsymbol{\mathcal{I}}, t) &= \mathrm{glb}_{\leq_{tr}}(\{[\![E_1]\!](\boldsymbol{\mathcal{I}}, t), \ldots, [\![E_m]\!](\boldsymbol{\mathcal{I}}, t)\}) \\ &= ([\![E_1 \wedge \ldots \wedge E_m]\!](I, t), [\![E_1 \wedge \ldots \wedge E_m]\!](I, t)).\end{aligned}$$

Analogously to what we did in Section 2, we naturally obtain the definition of *three-valued model*.

Definition 3.4

Let $\mathcal{P}$ be a program, $\mathcal{D}$ a dataset, and $\boldsymbol{\mathcal{I}} \in \mathcal{A}^{\mathsf{c}}_{\mathfrak{I}}$ a three-valued interpretation. Then, $\boldsymbol{\mathcal{I}}$ is a *(three-valued) model* of $\mathcal{P}$ iff for every rule $M \leftarrow B \in \mathsf{ground}(\mathcal{P})$, and for every $t \in \mathcal{T}$, $[\![B]\!](\boldsymbol{\mathcal{I}}, t) \leq_{tr} [\![M]\!](\boldsymbol{\mathcal{I}}, t)$. $\boldsymbol{\mathcal{I}}$ is a *(three-valued) model* of $\mathcal{P}$ and $\mathcal{D}$ iff $\boldsymbol{\mathcal{I}}$ is a three-valued model of $\mathcal{P}$ and I_1 is a model of $\mathcal{D}$.

Before defining the two-valued and the three-valued immediate consequence operators, we build a function $\mathcal{F}$ to make explicit the link between a ground head atom and the time points it refers to. Recall that we allow the nesting of $\boxplus$ and $\boxminus$ operators in the head of rules.

For all $t \in \mathcal{T}$, and for all ground head atoms $M \in \mathcal{M}$, we define:

$$\mathcal{F}(M,t) := \begin{cases} \bigcup_{t' \in \mathcal{T} | t - t' \in \delta} \mathcal{F}(M', t') & \text{if } M = \boxminus_\delta M' \\ \bigcup_{t' \in \mathcal{T} | t' - t \in \delta} \mathcal{F}(M', t') & \text{if } M = \boxplus_\delta M' \\ \{(M,t)\} & \text{otherwise} \end{cases}.$$

Example 3.2

For instance, if we consider the ground rule from Example 3.1, we get that $\mathcal{F}(\boxplus_{10}\mathsf{NewID}(\mathit{Linde}), t) = \{(\mathsf{NewID}(\mathit{Linde}), t + 10)\}$.

From the definitions of the two-valued and the three-valued semantics, and the construction of $\mathcal{F}$, it is easy to see the relation between a ground head atom M at time t and the relational facts it refers to, as expressed more precisely in the following propositions.

Proposition 3.1

Let $M \in \mathcal{M}$, $I \in \mathcal{I}$, $\mathcal{I} \in \mathcal{A}^{\mathsf{c}}_{\mathcal{I}}$, and $t \in \mathcal{T}$. The following statements hold:

1. $[\![M]\!](I,t) = \mathbf{t}$ if and only if $[\![\mathsf{P}(\mathbf{s})]\!](I,t') = \mathbf{t}$ for all $(\mathsf{P}(\mathbf{s}), t') \in \mathcal{F}(M,t)$.
2. $[\![M]\!](\mathcal{I},t) = \mathrm{glb}_{\leq_{tr}}(\{[\![\mathsf{P}(\mathbf{s})]\!](\mathcal{I},t') \mid (\mathsf{P}(\mathbf{s}), t') \in \mathcal{F}(M,t)\})$.

Proof. Item 1 is obvious by the definition of $\mathcal{F}$. For Item 2, by Definition 3.3, we can write the following equalities: $[\![M]\!](\mathcal{I},t) = ([\![M]\!](I_1,t), [\![M]\!](I_2,t))$, and $[\![\mathsf{P}(\mathbf{s})]\!](\mathcal{I},t') = ([\![\mathsf{P}(\mathbf{s})]\!](I_1,t'), [\![\mathsf{P}(\mathbf{s})]\!](I_2,t'))$ for all $(\mathsf{P}(\mathbf{s}), t') \in \mathcal{F}(M,t)$. Then Item 2 follows easily from Item 1. □

Now, the definition of the two-valued, and three-valued immediate consequence operators for a database $\mathcal{D}$ and a program $\mathcal{P}$ naturally follow.

Definition 3.5

Let $\mathcal{P}$ be a program, and $\mathcal{D}$ be a dataset. The mapping $T^{\mathcal{D},\mathcal{P}} : \mathcal{I} \to \mathcal{I}$ is called the *two-valued immediate consequence operator for* $\mathcal{D}$ *and* $\mathcal{P}$, and it is defined for all $I \in \mathcal{I}$ and for all $t \in \mathcal{T}$ as follows

$$T^{\mathcal{D},\mathcal{P}}(I)(t) := \{\mathsf{P}(\mathbf{s}) \mid \exists \delta \text{ such that } \mathsf{P}(\mathbf{s})@\delta \in \mathcal{D} \text{ and } t \in \delta\} \cup \left\{ \mathsf{P}(\mathbf{s}) \;\middle|\; \begin{array}{l} \exists M \leftarrow B \in \mathsf{ground}(\mathcal{P}), \exists t' \in \mathcal{T} \\ \text{such that } [\![B]\!](I,t') = \mathbf{t} \text{ and } (\mathsf{P}(\mathbf{s}), t) \in \mathcal{F}(M,t') \end{array} \right\}$$

Definition 3.6

Let $\mathcal{P}$ be a program, and $\mathcal{D}$ be a dataset. The mapping $\Psi^{\mathcal{D},\mathcal{P}} : \mathcal{A}^{\mathsf{c}}_{\mathfrak{I}} \to \mathcal{A}^{\mathsf{c}}_{\mathfrak{I}}$ is called the *three-valued immediate consequence operator for* $\mathcal{D}$ *and* $\mathcal{P}$, and its components $\Psi^{\mathcal{D},\mathcal{P}}_1 : \mathcal{A}^{\mathsf{c}}_{\mathfrak{I}} \to \mathfrak{I}$, and $\Psi^{\mathcal{D},\mathcal{P}}_2 : \mathcal{A}^{\mathsf{c}}_{\mathfrak{I}} \to \mathfrak{I}$ are defined for all $\boldsymbol{\mathcal{I}} \in \mathcal{A}^{\mathsf{c}}_{\mathfrak{I}}$ and for all $t \in \mathfrak{T}$ as follows

$$\Psi^{\mathcal{D},\mathcal{P}}_1(\boldsymbol{\mathcal{I}})(t) := \{\mathsf{P}(\mathbf{s}) \mid \exists \delta \text{ such that } \mathsf{P}(\mathbf{s})@\delta \in \mathcal{D} \text{ and } t \in \delta\} \cup \left\{ \mathsf{P}(\mathbf{s}) \;\middle|\; \begin{array}{l} \exists M \leftarrow B \in \mathsf{ground}(\mathcal{P}), \exists t' \in \mathfrak{T} \\ \text{such that } [\![B]\!](\boldsymbol{\mathcal{I}}, t') = \mathbf{t} \text{ and } (\mathsf{P}(\mathbf{s}), t) \in \mathcal{F}(M, t') \end{array} \right\},$$

$$\Psi^{\mathcal{D},\mathcal{P}}_2(\boldsymbol{\mathcal{I}})(t) := \{\mathsf{P}(\mathbf{s}) \mid \exists \delta \text{ such that } \mathsf{P}(\mathbf{s})@\delta \in \mathcal{D} \text{ and } t \in \delta\} \cup \left\{ \mathsf{P}(\mathbf{s}) \;\middle|\; \begin{array}{l} \exists M \leftarrow B \in \mathsf{ground}(\mathcal{P}), \exists t' \in \mathfrak{T} \\ \text{such that } [\![B]\!](\boldsymbol{\mathcal{I}}, t') \neq \mathbf{f} \text{ and } (\mathsf{P}(\mathbf{s}), t) \in \mathcal{F}(M, t') \end{array} \right\}.$$

Notice that the operator in Definition 3.6 is just an adaptation of Fitting's three valued immediate consequence operator [Fitting, 2002]. Intuitively, the first set appearing in each component forces the ground relational atoms of the dataset $\mathcal{D}$ to be interpreted as true at the right timepoint; then, the second set in the first component $\Psi^{\mathcal{D},\mathcal{P}}_1(\boldsymbol{\mathcal{I}})(t)$ ensures that every supported ground relational atom is true at the right timepoint via $\mathcal{F}$; finally, the second set in the second component $\Psi^{\mathcal{D},\mathcal{P}}_2(\boldsymbol{\mathcal{I}})(t)$ leaves out the atoms that are not supported, i.e., each ground relational atom $\mathsf{P}(\mathbf{s})$ such that every rule with $\mathsf{P}(\mathbf{s})$ in the head has false body is interpreted as false (at the correct corresponding timepoint via $\mathcal{F}$).

As expected, $T^{\mathcal{D},\mathcal{P}}$ and $\Psi^{\mathcal{D},\mathcal{P}}$ characterize the two-valued, and the three-valued models, respectively.

Proposition 3.2

Let $\mathcal{P}$ be a program, $\mathcal{D}$ a database, and $I \in \mathfrak{I}$. The interpretation I is a two-valued model of $\mathcal{D}$ and $\mathcal{P}$ if and only if I is a pre-fixpoint of $T^{\mathcal{D},\mathcal{P}}$.

Proof. Suppose that I is a two-valued model of $\mathcal{D}$ and $\mathcal{P}$. We have to show that for all $t \in \mathfrak{T}$, $T^{\mathcal{D},\mathcal{P}}(I)(t) \subseteq I(t)$. Let $\mathsf{P}(\mathbf{s}) \in T^{\mathcal{D},\mathcal{P}}(I)(t)$. If there exists δ such that $t \in \delta$ and $\mathsf{P}(\mathbf{s})@\delta \in \mathcal{D}$, then $\mathsf{P}(\mathbf{s}) \in I(t)$ because I is, in particular, a model of $\mathcal{D}$. Let us now assume that such an interval δ does not exist. Then, it must be the case that there exists a ground rule $M \leftarrow B \in \mathsf{ground}(\mathcal{P})$ and a time point $t' \in \mathfrak{T}$ such that $[\![B]\!](I, t') = \mathbf{t}$ and $(\mathsf{P}(\mathbf{s}), t) \in \mathcal{F}(M, t')$. Since I is a two-valued model of $\mathcal{P}$, it holds that $[\![B]\!](I, t') \leq_o [\![M]\!](I, t')$; hence, it must hold that $[\![M]\!](I, t') = \mathbf{t}$. By Proposition 3.1, it follows that $[\![\mathsf{P}(\mathbf{s})]\!](I, t) = \mathbf{t}$, i.e., $\mathsf{P}(\mathbf{s}) \in I(t)$, as desired.

Conversely, assume that for all $t \in \mathcal{T}$, $T^{\mathcal{D},\mathcal{P}}(I)(t) \subseteq I(t)$. By definition of $T^{\mathcal{D},\mathcal{P}}$, for all ground relational facts $\mathsf{P}(\mathbf{s})@\delta \in \mathcal{D}$, and for all $t \in \delta$, it holds that $\mathsf{P}(\mathbf{s}) \in T^{\mathcal{D},\mathcal{P}}(I)(t) \subseteq I(t)$. Hence, I is a model of $\mathcal{D}$. We now prove that I is also a two-valued model of $\mathcal{P}$. Let $M \leftarrow B \in \mathsf{ground}(\mathcal{P})$, and $t \in \mathcal{T}$, and suppose $[\![B]\!](I,t) = \mathbf{t}$. It suffices to show that $[\![M]\!](I,t) = \mathbf{t}$. If $M = \top$, this is trivially verified. Suppose now that $M \neq \top$. Observe that, by definition of $T^{\mathcal{D},\mathcal{P}}$, for all $(\mathsf{P}(\mathbf{s}),t') \in \mathcal{F}(M,t)$, $\mathsf{P}(\mathbf{s}) \in T^{\mathcal{D},\mathcal{P}}(I)(t') \subseteq I(t')$. In particular, $[\![\mathsf{P}(\mathbf{s})]\!](I,t') = \mathbf{t}$ for all $(\mathsf{P}(\mathbf{s}),t') \in \mathcal{F}(M,t)$. Hence, by Proposition 3.1, $[\![M]\!](I,t) = \mathbf{t}$, as desired. □

Proposition 3.3

Let $\mathcal{P}$ be a program, $\mathcal{D}$ a database, and $\mathcal{I} \in \mathcal{A}^{\mathsf{tj}}_{\mathsf{j}}$. The interpretation $\mathcal{I}$ is a three-valued model of $\mathcal{D}$ and $\mathcal{P}$ if and only if $\mathcal{I}$ is a pre-fixpoint of $\Psi^{\mathcal{D},\mathcal{P}}$, i.e., $\Psi^{\mathcal{D},\mathcal{P}}(\mathcal{I}) \leq_{tr} \mathcal{I}$.

Proof. Suppose that $\mathcal{I} = (I_1, I_2)$ is a three-valued model of $\mathcal{D}$ and $\mathcal{P}$. We have to show that $\Psi^{\mathcal{D},\mathcal{P}}(\mathcal{I}) \leq_{tr} \mathcal{I}$. Let $t \in \mathcal{T}$, and $\mathsf{P}(\mathbf{s})$ be a ground relational atom such that $\Psi^{\mathcal{D},\mathcal{P}}(\mathcal{I})(t)(\mathsf{P}(\mathbf{s})) = \mathbf{t}$. By Definition 3.6, either there exists an interval δ such that $t \in \delta$ and $\mathsf{P}(\mathbf{s})@\delta \in \mathcal{D}$, in which case $\mathcal{I}(t)(\mathsf{P}(\mathbf{s})) = \mathbf{t}$ (as $\mathcal{I}$ is a model of $\mathcal{D}$ by hypothesis); or there exists a ground rule $M \leftarrow B \in \mathsf{ground}(\mathcal{P})$ and a time point $t' \in \mathcal{T}$ such that $[\![B]\!](\mathcal{I},t') = \mathbf{t}$ and $(\mathsf{P}(\mathbf{s}),t) \in \mathcal{F}(M,t')$. Since $\mathcal{I}$ is a three-valued model of $\mathcal{P}$, it holds that $[\![B]\!](\mathcal{I},t') \leq_{tr} [\![M]\!](\mathcal{I},t')$; hence, it must hold that $[\![M]\!](\mathcal{I},t') = \mathbf{t}$. By Proposition 3.1, it follows that $\mathcal{I}(t)(\mathsf{P}(\mathbf{s})) = (I_1(t)(\mathsf{P}(\mathbf{s})), I_2(t)(\mathsf{P}(\mathbf{s})) = ([\![\mathsf{P}(\mathbf{s})]\!](I_1,t), [\![\mathsf{P}(\mathbf{s})]\!](I_2,t)) = [\![\mathsf{P}(\mathbf{s})]\!](\mathcal{I},t) = \mathbf{t}$, as desired.

Now let $\mathsf{P}(\mathbf{s})$ be a ground relational atom such that $\Psi^{\mathcal{D},\mathcal{P}}(\mathcal{I})(t)(\mathsf{P}(\mathbf{s})) = \mathbf{u}$. Hence, by Definition 3.6, there exists a ground rule $M \leftarrow B \in \mathsf{ground}(\mathcal{P})$ and a time point $t' \in \mathcal{T}$ such that $[\![B]\!](\mathcal{I},t') \neq \mathbf{f}$ and $(\mathsf{P}(\mathbf{s}),t) \in \mathcal{F}(M,t')$. Since $\mathcal{I}$ is a three-valued model of $\mathcal{P}$, it holds that $\mathbf{u} \leq_{tr} [\![B]\!](\mathcal{I},t') \leq_{tr} [\![M]\!](\mathcal{I},t')$. In particular, we have that $[\![M]\!](\mathcal{I},t') \neq \mathbf{f}$. By Proposition 3.1, it follows that $\mathcal{I}(t)(\mathsf{P}(\mathbf{s})) = [\![\mathsf{P}(\mathbf{s})]\!](\mathcal{I},t) \neq \mathbf{f}$, concluding the first half of the proof.

Conversely, assume that for all $t \in \mathcal{T}$, $\Psi^{\mathcal{D},\mathcal{P}}(\mathcal{I})(t) \leq_{tr} \mathcal{I}(t)$. By definition of $\Psi^{\mathcal{D},\mathcal{P}}$, for all ground relational facts $\mathsf{P}(\mathbf{s})@\delta \in \mathcal{D}$, and for all $t \in \delta$, it holds that $\mathbf{t} = \Psi^{\mathcal{D},\mathcal{P}}(\mathcal{I})(t)(\mathsf{P}(\mathbf{s})) \leq_{tr} \mathcal{I}(t)(\mathsf{P}(\mathbf{s}))$. Hence, $\mathcal{I}$ is a model of $\mathcal{D}$. We now prove that $\mathcal{I}$ is also a three-valued model of $\mathcal{P}$. Let $M \leftarrow B \in \mathsf{ground}(\mathcal{P})$, and $t \in \mathcal{T}$. We have to show that $[\![B]\!](\mathcal{I},t) \leq_{tr} [\![M]\!](\mathcal{I},t)$. If $M = \top$, this is trivially verified, so assume that $M \neq \top$. We first suppose $[\![B]\!](\mathcal{I},t) = \mathbf{t}$ and show that $[\![M]\!](\mathcal{I},t) = \mathbf{t}$. Observe that, by definition of $\Psi^{\mathcal{D},\mathcal{P}}$, for all $(\mathsf{Q}(\mathbf{s}),t') \in \mathcal{F}(M,t)$, it holds that $\mathbf{t} = \Psi^{\mathcal{D},\mathcal{P}}(\mathcal{I})(t')(\mathsf{Q}(\mathbf{s})) \leq_{tr} \mathcal{I}(t')(\mathsf{Q}(\mathbf{s}))$. In particular, $[\![\mathsf{Q}(\mathbf{s})]\!](\mathcal{I},t') = \mathbf{t}$ for all $(\mathsf{Q}(\mathbf{s}),t') \in \mathcal{F}(M,t)$. Hence, by Proposition 3.1, $[\![M]\!](\mathcal{I},t) = \mathbf{t}$, as desired. Now, we assume $[\![B]\!](\mathcal{I},t) = \mathbf{u}$ and show that $[\![M]\!](\mathcal{I},t) \neq \mathbf{f}$. Observe that, by definition of $\Psi^{\mathcal{D},\mathcal{P}}$, for all $(\mathsf{Q}(\mathbf{s}),t') \in \mathcal{F}(M,t)$, it holds that $\mathbf{u} \leq_{tr} \Psi^{\mathcal{D},\mathcal{P}}(\mathcal{I})(t')(\mathsf{Q}(\mathbf{s})) \leq_{tr} \mathcal{I}(t')(\mathsf{Q}(\mathbf{s}))$. In particular, $[\![\mathsf{Q}(\mathbf{s})]\!](\mathcal{I},t') \geq_{tr} \mathbf{u}$

for all $(\mathbf{Q}(\mathbf{s}), t') \in \mathcal{F}(M, t)$. Hence, by Proposition 3.1, $[\![M]\!](\mathcal{I}, t) \geq_{tr} \mathbf{u}$, concluding the proof. □

4 Approximation Fixpoint Theory for DatalogMTL$^\neg$

After having defined the three-valued semantics and the three-valued immediate consequence operator in the previous section, it only remains to show that the three-valued immediate consequence operator $\Psi^{\mathcal{D},\mathcal{P}}$ is a consistent approximator of $T^{\mathcal{D},\mathcal{P}}$ (Theorem 3.1). Then, the machinery of AFT allows for the definition of four kinds of models of a database $\mathcal{D}$ and a program $\mathcal{P}$: *supported*, *stable*, *Kripke-Kleene*, and *well-founded*.

We start by proving that the space of two-valued interpretations $\mathfrak{I}$ is a complete lattice, which follows straightforwardly from results of order theory.

Proposition 3.4

$\langle \mathfrak{I}, \subseteq \rangle$ is a complete lattice.

Proof. Let $\mathcal{G}$ be the set of ground relational atoms. Then $\mathfrak{I}$ is the set of functions with signature $\mathcal{T} \to 2^{\mathcal{G}}$. The power set $2^{\mathcal{G}}$ ordered with set inclusion $\subseteq$ is clearly a complete lattice (see Example 2.8 in Chapter 2).
Let $S \subseteq \mathfrak{I}$, and let $I_{\mathrm{glb}(S)}, I_{\mathrm{lub}(S)} : \mathcal{T} \to 2^{\mathcal{G}}$ be the interpretations defined by $I_{\mathrm{glb}(S)}(t) := \mathrm{glb}(\{J(t) \mid J \in S\})$ and $I_{\mathrm{lub}(S)}(t) := \mathrm{lub}(\{J(t) \mid J \in S\})$, for all $t \in \mathcal{T}$. By the definition of $I_{\mathrm{glb}(S)}$ and $I_{\mathrm{lub}(S)}$, it holds that $I_{\mathrm{glb}(S)}(t) \subseteq J(t) \subseteq I_{\mathrm{lub}(S)}(t)$ for all $J \in S$ and for all $t \in \mathcal{T}$. Hence, $I_{\mathrm{glb}(S)} \subseteq J \subseteq I_{\mathrm{lub}(S)}$ for all $J \in S$, i.e., $I_{\mathrm{glb}(S)}$ is a lower bound of S and $I_{\mathrm{lub}(S)}$ is an upper bound of S.
Now, we show that $I_{\mathrm{glb}(S)}$ is the greatest lower bound of S. Let $L \in \mathfrak{I}$ be a lower bound of S, i.e., $L(t) \subseteq J(t)$ for all $J \in S$ and for all $t \in \mathcal{T}$. It follows that $L(t) \subseteq \mathrm{glb}(\{J(t) \mid J \in S\}) = I_{\mathrm{glb}(S)}(t)$ for all $t \in \mathcal{T}$, i.e., $L \subseteq I_{\mathrm{glb}(S)}$. Hence, $I_{\mathrm{glb}(S)} = \mathrm{glb}(S)$.
The analogous can be done to prove that $I_{\mathrm{lub}(S)}$ is the least upper bound of S, concluding the proof. □

Using the terminology of consistent AFT (Section 2.2 of Chapter 2), the set of three-valued interpretations $\mathcal{A}^{\mathrm{c}}_{\mathfrak{I}}$ equipped with the precision order $\leq_p$ ($\mathcal{I} \leq_p \mathcal{J}$ iff $I_1 \subseteq J_1 \subseteq J_2 \subseteq I_2$) is an *approximation space* for $\mathfrak{I}$. In order to prove Theorem 3.1, we make use of the following two lemmata, which ensure the evaluation of ground metric atoms preserves the order relation in both the two-valued and the three-valued case.

Lemma 3.1

Let $I_1, I_2 \in \mathfrak{I}$ be two-valued interpretations, and M be a ground atom. If $I_1 \subseteq I_2$, then for all $t \in \mathcal{T}$, $[\![M]\!](I_1, t) \leq_o [\![M]\!](I_2, t)$.

Proof. Let $I_1 \subseteq I_2$, i.e., $I_1(t) \subseteq I_2(t)$ for all $t \in \mathcal{T}$. We proceed by induction on the structure of M.
If $M \in \{\bot, \top\}$, then clearly $[\![M]\!](I_1,t) = [\![M]\!](I_2,t)$. If $M = \mathsf{P}(\mathbf{s})$, then $[\![M]\!](I_1,t) \leq_o [\![M]\!](I_2,t)$ follows from Definition 3.1 and the assumption that $I_1(t) \subseteq I_2(t)$.
Now, let $M = \diamondsuit_\delta M'$, and assume $[\![M']\!](I_1,t) \leq_o [\![M']\!](I_2,t)$ for all $t \in \mathcal{T}$. If $[\![M]\!](I_1,t) = \mathbf{t}$ for some $t \in \mathcal{T}$, then by Definition 3.1, there exists $t' \in \mathcal{T}$ such that $t - t' \in \delta$ for which $[\![M']\!](I_1,t') = \mathbf{t}$. By induction hypothesis, it must hold that $[\![M']\!](I_2,t') = \mathbf{t}$, as desired. We can proceed analogously for the cases $M = \boxplus_\delta M'$, $M = \boxminus_\delta M'$, and $M = \boxplus_\delta M'$.
Let $M = M_1 \mathcal{S}_\delta M_2$, and assume $[\![M_1]\!](I_1,t) \leq_o [\![M_1]\!](I_2,t)$, and $[\![M_2]\!](I_1,t) \leq_o [\![M_2]\!](I_2,t)$ for all $t \in \mathcal{T}$. If $[\![M]\!](I_1,t) = \mathbf{t}$ for some $t \in \mathcal{T}$, then by Definition 3.1, there exists $t' \in \mathcal{T}$ such that $t - t' \in \delta$, $[\![M_2]\!](I_1,t') = \mathbf{t}$, and $[\![M_1]\!](I_1,t'') = \mathbf{t}$ for all $t'' \in (t',t)$. By induction hypothesis, it must hold that $[\![M_2]\!](I_2,t') = \mathbf{t}$, and $[\![M_1]\!](I_2,t'') = \mathbf{t}$ for all $t'' \in (t',t)$. It follows that $[\![M]\!](I_2,t) = \mathbf{t}$, as desired. We can proceed analogously for the case $M = M_1 \mathcal{U}_\delta M_2$. □

Lemma 3.2

Let $\mathcal{I}, \mathcal{J} \in \mathcal{A}^{\mathsf{c}}_{\mathsf{J}}$ be three-valued interpretations, and E be a ground expression. If $\mathcal{I} \leq_p \mathcal{J}$, then for all $t \in \mathcal{T}$, $[\![E]\!](\mathcal{I},t) \leq_p [\![E]\!](\mathcal{J},t)$.

Proof. Assume that $(I_1,I_2) = \mathcal{I} \leq_p \mathcal{J} = (J_1,J_2)$, i.e., $I_1(t) \subseteq J_1(t) \subseteq J_2(t) \subseteq I_2(t)$ for all $t \in \mathcal{T}$. We proceed by induction on the structure of E.
If E is a ground metric atom, then by Lemma 3.1, we have $[\![E]\!](I_1,t) \leq_o [\![E]\!](J_1,t) \leq_o [\![E]\!](J_2,t) \leq_o [\![E]\!](I_2,t)$. By Definition 3.3, we obtain $[\![E]\!](\mathcal{I},t) \leq_p [\![E]\!](\mathcal{J},t)$.
Now assume $E = \neg M$ for some ground metric atom M, and that $[\![M]\!](\mathcal{I},t) \leq_p [\![M]\!](\mathcal{J},t)$. If $[\![M]\!](\mathcal{I},t) = [\![M]\!](\mathcal{J},t)$, then clearly $[\![E]\!](\mathcal{I},t) = [\![E]\!](\mathcal{J},t)$. Now assume $[\![M]\!](\mathcal{I},t) <_p [\![M]\!](\mathcal{J},t)$. It must be that $[\![M]\!](\mathcal{I},t) = \mathbf{u}$ and $[\![M]\!](\mathcal{J},t) \in \{\mathbf{t},\mathbf{f}\}$. By Definition 3.3, it clearly follows that $[\![E]\!](\mathcal{I},t) = \mathbf{u}$ and $[\![E]\!](\mathcal{J},t) \in \{\mathbf{t},\mathbf{f}\}$; hence $[\![E]\!](\mathcal{I},t) \leq_p [\![E]\!](\mathcal{J},t)$.
Finally, for the case $E = E_1 \wedge \ldots \wedge E_m$, we can easily use the induction hypothesis on $E_1, \ldots, E_m$ to obtain that $[\![E]\!](\mathcal{I},t) \leq_p [\![E]\!](\mathcal{J},t)$. □

Theorem 3.1

Let $\mathcal{P}$ be a program, and $\mathcal{D}$ be a dataset. The operator $\Psi^{\mathcal{D},\mathcal{P}}$ is an approximator of $T^{\mathcal{D},\mathcal{P}}$.

Proof. We have to prove three things:

1. $\Psi^{\mathcal{D},\mathcal{P}}$ *is consistent*, i.e., for all $\mathcal{I} \in \mathcal{A}^{\mathsf{c}}_{\mathsf{J}}$ that are exact, $\Psi^{\mathcal{D},\mathcal{P}}(\mathcal{I})$ is also exact.
2. $\Psi^{\mathcal{D},\mathcal{P}}$ *is* $\leq_p$*-monotonic*, i.e., for all $\mathcal{I}, \mathcal{J} \in \mathcal{A}^{\mathsf{c}}_{\mathsf{J}}$ such that $\mathcal{I} \leq_p \mathcal{J}$, it holds that $\Psi^{\mathcal{D},\mathcal{P}}(\mathcal{I}) \leq_p \Psi^{\mathcal{D},\mathcal{P}}(\mathcal{J})$.

3. $\Psi^{\mathcal{D},\mathcal{P}}$ *approximates* $T^{\mathcal{D},\mathcal{P}}$, i.e., for all $I \in \mathfrak{I}$, $\Psi^{\mathcal{D},\mathcal{P}}(I,I) = (T^{\mathcal{D},\mathcal{P}}(I), T^{\mathcal{D},\mathcal{P}}(I))$.

Item 1 follows from Item 3, which we prove below, after Item 2.
Now, to prove Item 2, let $\boldsymbol{\mathcal{I}} \leq_p \boldsymbol{\mathcal{J}}$ be two elements of $\mathcal{A}^{\mathsf{c}}_{\mathfrak{I}}$, and $t \in \mathfrak{T}$. We need to show that $\Psi_1^{\mathcal{D},\mathcal{P}}(\boldsymbol{\mathcal{I}})(t) \subseteq \Psi_1^{\mathcal{D},\mathcal{P}}(\boldsymbol{\mathcal{J}})(t)$ and $\Psi_2^{\mathcal{D},\mathcal{P}}(\boldsymbol{\mathcal{J}})(t) \subseteq \Psi_2^{\mathcal{D},\mathcal{P}}(\boldsymbol{\mathcal{I}})(t)$. Let $\mathsf{P}(\mathbf{s}) \in \Psi_1^{\mathcal{D},\mathcal{P}}(\boldsymbol{\mathcal{I}})(t)$ (if there is no such $\mathsf{P}(\mathbf{s})$, then we already have $\Psi_1^{\mathcal{D},\mathcal{P}}(\boldsymbol{\mathcal{I}})(t) \subseteq \Psi_1^{\mathcal{D},\mathcal{P}}(\boldsymbol{\mathcal{J}})(t)$). If there exists δ such that $t \in \delta$ and $\mathsf{P}(\mathbf{s})@\delta \in \mathcal{D}$, then $\mathsf{P}(\mathbf{s}) \in \Psi_1^{\mathcal{D},\mathcal{P}}(\boldsymbol{\mathcal{J}})(t)$ too. If this is not the case, then there exists a ground rule $M \leftarrow B \in \mathsf{ground}(\mathcal{P})$ and $t' \in \mathfrak{T}$ such that $[\![B]\!](\boldsymbol{\mathcal{I}}, t') = \mathbf{t}$ and $(\mathsf{P}(\mathbf{s}), t) \in \mathcal{F}(M, t')$. By Lemma 3.2, it holds that $[\![B]\!](\boldsymbol{\mathcal{J}}, t') = \mathbf{t}$, hence $\mathsf{P}(\mathbf{s}) \in \Psi_1^{\mathcal{D},\mathcal{P}}(\boldsymbol{\mathcal{J}})(t)$. We have so far proved the first inclusion, i.e., $\Psi_1^{\mathcal{D},\mathcal{P}}(\boldsymbol{\mathcal{I}})(t) \subseteq \Psi_1^{\mathcal{D},\mathcal{P}}(\boldsymbol{\mathcal{J}})(t)$. Showing that the second inclusion holds can be performed analogously.
Finally, for Item 3, let B be the body of a ground rule in $\mathsf{ground}(\mathcal{P})$, and $t \in \mathfrak{T}$. It is easy to prove by induction on B, and using Definition 3.3 and Remark 3.1, that $[\![B]\!]((I,I),t) = ([\![B]\!](I,t), [\![B]\!](I,t))$. In particular, $[\![B]\!]((I,I),t) \neq \mathbf{u}$, and $[\![B]\!]((I,I),t) = \mathbf{t}$ if and only if $[\![B]\!]((I,I),t) \neq \mathbf{f}$. Hence, by Definitions 3.5 and 3.6, for all $t \in \mathfrak{T}$, $\Psi_1^{\mathcal{D},\mathcal{P}}(I,I)(t) = \Psi_2^{\mathcal{D},\mathcal{P}}(I,I)(t) = T^{\mathcal{D},\mathcal{P}}(I)(t)$, as desired. □

Thanks to Theorem 3.1, and the machinery of AFT, we are able to define the following models of interest.

Definition 3.7

Let $\mathcal{D}$ be a dataset, $\mathcal{P}$ be a program, and $\boldsymbol{\mathcal{I}} \in \mathcal{A}^{\mathsf{c}}_{\mathfrak{I}}$.

- $\boldsymbol{\mathcal{I}}$ is a *three-valued supported model of* $\mathcal{D}$ *and* $\mathcal{P}$ if it is a fixpoint of $\Psi^{\mathcal{D},\mathcal{P}}$;
- $\boldsymbol{\mathcal{I}}$ is the *Kripke-Kleene model of* $\mathcal{D}$ *and* $\mathcal{P}$ if it is the $\leq_p$-least fixpoint of $\Psi^{\mathcal{D},\mathcal{P}}$;
- $\boldsymbol{\mathcal{I}}$ is a *three-valued stable model of* $\mathcal{D}$ *and* $\mathcal{P}$ if it is a stable fixpoint of $\Psi^{\mathcal{D},\mathcal{P}}$; that is, if $I_1 = \mathrm{lfp}(\Psi_1^{\mathcal{D},\mathcal{P}}(\cdot, I_2))$ and $I_2 = \mathrm{lfp}(\Psi_2^{\mathcal{D},\mathcal{P}}(I_1, \cdot))$, where $\Psi_1^{\mathcal{D},\mathcal{P}}(\cdot, I_2)$ is the function that maps an interpretation $J \subseteq I_2$ to the first component of $\Psi^{\mathcal{D},\mathcal{P}}(J, I_2)$, and analogously for $\Psi_2^{\mathcal{D},\mathcal{P}}(I_1, \cdot)$;
- $\boldsymbol{\mathcal{I}}$ is the *well-founded model of* $\mathcal{D}$ *and* $\mathcal{P}$ if it is the well-founded fixpoint of $\Psi^{\mathcal{D},\mathcal{P}}$, i.e., if it is the $\leq_p$-least three-valued stable model.

We call a two-valued model I of $\mathcal{D}$ and $\mathcal{P}$ a *two-valued stable* (respectively, *supported*) *model of* $\mathcal{D}$ *and* $\mathcal{P}$ if (I,I) is a three-valued stable (respectively, supported) model of $\mathcal{D}$ and $\mathcal{P}$.

Remark 3.2

Recall that the upper bound of the three-valued immediate consequence operator $\Psi^{\mathcal{D},\mathcal{P}}$ removes atoms that are not supported in $\boldsymbol{\mathcal{I}}$, making them false in the resulting

interpretation $\Psi^{\mathcal{D},\mathcal{P}}(\boldsymbol{\mathcal{I}})$. However, the upper bound of the *stable revision operator* $\mathcal{S}_{\Psi^{\mathcal{D},\mathcal{P}}}(I_1, I_2) := (\mathrm{lfp}(\Psi_1^{\mathcal{D},\mathcal{P}}(\cdot, I_2)), \mathrm{lfp}(\Psi_2^{\mathcal{D},\mathcal{P}}(I_1, \cdot)))$, used for the definition of stable models and well-founded model above, removes atoms belonging to unfounded sets with respect to (I_1, I_2), making them false in the resulting interpretation $\mathcal{S}_{\Psi^{\mathcal{D},\mathcal{P}}}(I_1, I_2)$.

Proposition 3.5

Let $\mathcal{D}$ be a dataset, $\mathcal{P}$ be a program, and $(I_1, I_2) \in \mathcal{A}_{\mathcal{J}}^{c}$. If (I_1, I_2) is a supported, Kripke-Kleene, stable, or well-founded model, then I_1 is a model of $\mathcal{D}$.

Proof. If (I_1, I_2) is a supported or the Kripke-Kleene model, then (I_1, I_2) is a fixpoint of $\Psi^{\mathcal{D},\mathcal{P}}$. In particular, it holds that $I_1 = \Psi_1^{\mathcal{D},\mathcal{P}}(I_1, I_2)$. Hence, by definition of $\Psi^{\mathcal{D},\mathcal{P}}$ (Definition 3.6), I_1 is a model of $\mathcal{D}$.
If (I_1, I_2) is a stable or the well-founded model, then (I_1, I_2) is a fixpoint of the stable revision operator of $\Psi^{\mathcal{D},\mathcal{P}}$. In particular, it holds that $I_1 = \mathrm{lfp}(\Psi_1^{\mathcal{D},\mathcal{P}}(\cdot, I_2))$. Hence, we have that $I_1 = \Psi_1^{\mathcal{D},\mathcal{P}}(I_1, I_2)$, and we can again conclude that I_1 is a model of $\mathcal{D}$. □

Example 3.3

Astrid and Dieter live together and they alternate house chores. We use the 0-arity predicate `dirty` to indicate that the house is dirty and in need of a general cleaning. Moreover, the 0-arity predicate `Acleans` (respectively, `Dcleans`) is true if Astrid (respectively, Dieter) is cleaning. We assume that the house needs a deep cleaning at least once a week (Line 2 in Listing 3.1), and that Astrid and Dieter alternate doing the cleaning. To keep track of whether Astrid cleaned the last time, we use a 0-arity predicate `Alast` (Lines 7 and 8 in Listing 3.1). We write the rules defining all the aforementioned predicates in a program $\mathcal{P}$, contained in Listing 3.1.

Listing 3.1: House cleaning.

```
1  % The house is dirty on the seventh day with no cleaning
2  dirty ← ¬⟐[1,6]Acleans ∧ ¬⟐[1,6]Bcleans.
3
4  % Astrid and Dieter alternate the cleaning.
5  Acleans ← dirty ∧ ¬Alast.
6  Bcleans ← dirty ∧ Alast.
7  Alast ← ⊟1Acleans.
8  Alast ← ⊟1Alast ∧ ¬⊟1Dcleans.
```

Let $\mathcal{D} = \emptyset$ be the empty dataset, and $\mathcal{T} = \mathbb{Z}$. It is easy to see that both the Kripke-Kleene and the well-founded model coincide with the three-valued interpretation that leaves every predicate undefined, for each timepoint in the timeline $\mathcal{T}$. The set

of two-valued stable models coincides with the set of two-valued supported models: each model corresponds to a specific cleaning schedule. In particular, there exists 14 two-valued stable (or suported) models. We can write them compactly as follows: for each $a \in [1, 14]$ we define the two-valued interpretation I_a for each $t \in [a-6, a+7]$, and for each $k \in \mathcal{T}$ as

$$I_a(t + 14k) := \begin{cases} \emptyset & \text{if } t \in [a-6, a-1] \\ \{\mathsf{dirty}, \mathsf{Acleans}\} & \text{if } t = a \\ \{\mathsf{Alast}\} & \text{if } t \in [a+1, a+6] \\ \{\mathsf{dirty}, \mathsf{Alast}, \mathsf{Dcleans}\} & \text{if } t = a+7 \end{cases}.$$

5 Comparison with Previous Work

In this section, we compare the definitions of *HT-models* and stable models of Walega et al. [2024] with ours. In particular, we first show that any three-valued model is an HT-model from [Walega et al., 2024], while the converse does not hold in general. Finally, we prove that indeed the notion of *stable model* from [Walega et al., 2024] coincides with our definition of two-valued stable model. For the sake of clarity, we rewrite the definitions of HT-model (Definition 3.8) and stable model (Definition 3.9) from [Walega et al., 2024] with the notation used in this thesis. To avoid ambiguity, we will rename the stable models from [Walega et al., 2024] "stable HT-models".

Definition 3.8

Let $\mathcal{D}$ be a dataset, $\mathcal{P}$ be a program, and $\boldsymbol{\mathcal{I}} \in \mathcal{A}^{\mathsf{c}}_{\mathsf{J}}$. $\boldsymbol{\mathcal{I}}$ is an *HT-model of* $\mathcal{D}$ *and* $\mathcal{P}$ if I_1 is a model of $\mathcal{D}$ and for all rules $M \leftarrow M_1 \wedge \ldots \wedge M_k \wedge \neg M_{k+1} \wedge \ldots \wedge \neg M_m \in \mathsf{ground}(\mathcal{P})$, for all $t \in \mathcal{T}$, the following conditions hold

1. if $[\![M_i]\!](I_1, t) = \mathbf{t}$ for all $i \in \{1, \ldots, k\}$ and $[\![M_j]\!](I_2, t) = \mathbf{f}$ for all $j \in \{k+1, \ldots, m\}$, then $[\![M]\!](I_1, t) = \mathbf{t}$;
2. if $[\![M_i]\!](I_2, t) = \mathbf{t}$ for all $i \in \{1, \ldots, k\}$ and $[\![M_j]\!](I_2, t) = \mathbf{f}$ for all $j \in \{k+1, \ldots, m\}$, then $[\![M]\!](I_2, t) = \mathbf{t}$.

Proposition 3.6

Let $\mathcal{D}$ be a dataset, $\mathcal{P}$ be a program, and $\boldsymbol{\mathcal{I}} \in \mathcal{A}^{\mathsf{c}}_{\mathsf{J}}$. If $\boldsymbol{\mathcal{I}}$ is a three-valued model of $\mathcal{D}$ and $\mathcal{P}$, then $\boldsymbol{\mathcal{I}}$ is an HT-model of $\mathcal{D}$ and $\mathcal{P}$.

Proof. Clearly, I_1 is a model for $\mathcal{D}$ by hypothesis. We now show that Item 1 from Definition 3.8 holds. Let $M \leftarrow B \in \mathsf{ground}(\mathcal{P})$ be a ground rule with $B = M_1 \wedge \ldots \wedge M_k \wedge \neg M_{k+1} \wedge \ldots \wedge \neg M_m$, and $t \in \mathcal{T}$ such that $[\![M_i]\!](I_1, t) = \mathbf{t}$ for all $i \in \{1, \ldots, k\}$ and $[\![M_j]\!](I_2, t) = \mathbf{f}$ for all $j \in \{k+1, \ldots, m\}$. This implies that $[\![M_i]\!](\mathcal{I}, t) = \mathbf{t}$ for all $i \in \{1, \ldots, k\}$ and $[\![M_j]\!](\mathcal{I}, t) = \mathbf{f}$ for all $j \in \{k+1, \ldots, m\}$. In particular, by Definition 3.3 we have that $[\![B]\!](\mathcal{I}, t) = \mathbf{t}$. Since $\mathcal{I}$ is a three-valued model by hypothesis, it must be that $[\![M]\!](\mathcal{I}, t) = \mathbf{t}$ too. Hence, $[\![M]\!](I_1, t) = \mathbf{t}$, as desired.
Now, we prove that Item 2 from Definition 3.8 holds as well. Let $M \leftarrow B \in \mathsf{ground}(\mathcal{P})$ be a ground rule with $B = M_1 \wedge \ldots \wedge M_k \wedge \neg M_{k+1} \wedge \ldots \wedge \neg M_m$, and $t \in \mathcal{T}$ such that $[\![M_i]\!](I_2, t) = \mathbf{t}$ for all $i \in \{1, \ldots, k\}$ and $[\![M_j]\!](I_2, t) = \mathbf{f}$ for all $j \in \{k+1, \ldots, m\}$. This implies that $[\![M_i]\!](\mathcal{I}, t) \neq \mathbf{f}$ for all $i \in \{1, \ldots, k\}$ and $[\![M_j]\!](\mathcal{I}, t) = \mathbf{f}$ for all $j \in \{k+1, \ldots, m\}$. In particular, by Definition 3.3 we have that $[\![B]\!](\mathcal{I}, t) \neq \mathbf{f}$. Since $\mathcal{I}$ is a three-valued model, it must be that $[\![M]\!](\mathcal{I}, t) \neq \mathbf{f}$. Hence, $[\![M]\!](I_2, t) = \mathbf{t}$, concluding the proof. □

The converse of Proposition 3.6 does not hold, i.e., not all HT-models are three-valued models. In fact, by Definition 3.4, $\mathcal{I} \in \mathcal{A}^{\mathsf{c}}_{\mathsf{j}}$ is a three-valued model of $\mathcal{P}$ iff for every rule $M \leftarrow B \in \mathsf{ground}(\mathcal{P})$, and for every $t \in \mathcal{T}$, it holds that $[\![B]\!](\mathcal{I}, t) \leq_{tr} [\![M]\!](\mathcal{I}, t)$, i.e., it holds that

1. if $[\![M_i]\!](I_1, t) = \mathbf{t}$ for all $i \in \{1, \ldots, k\}$ and $[\![M_j]\!](I_2, t) = \mathbf{f}$ for all $j \in \{k+1, \ldots, m\}$, then $[\![M]\!](I_1, t) = \mathbf{t}$,
2. if $[\![M_i]\!](I_2, t) = \mathbf{t}$ for all $i \in \{1, \ldots, k\}$ and $[\![M_j]\!](I_1, t) = \mathbf{f}$ for all $j \in \{k+1, \ldots, m\}$, then $[\![M]\!](I_2, t) = \mathbf{t}$,

where $B := M_1 \wedge \ldots \wedge M_k \wedge \neg M_{k+1} \wedge \ldots \wedge \neg M_m$. Notice that, while the first condition above is the same as the first one in Definition 3.8, the second condition is more strict than the second condition of Definition 3.8: it applies even when $[\![M_h]\!](I_2, t) = \mathbf{t}$ for some $h \in \{k+1, \ldots, m\}$, provided that $[\![M_j]\!](I_1, t) = \mathbf{f}$ for all $j \in \{k+1, \ldots, m\}$.

We show this discrepancy more concretely in the example below.

Example 3.4

Consider an empty dataset $\mathcal{D}$ and a program $\mathcal{P}$ with just one rule being $\mathsf{P} \leftarrow \neg\mathsf{Q}$, where P and Q are predicates of arity 0. Let (I_1, I_2) be the three-valued interpretation with $I_1(t) = \emptyset$ and $I_2(t) = \{\mathsf{Q}\}$ for all $t \in \mathcal{T}$. Clearly, I_1 is a model for the empty dataset $\mathcal{D}$. Moreover, since the antecedent in both conditions of Definition 3.8 are not satisfied, (I_1, I_2) is an HT-model for $\mathcal{D}$ and $\mathcal{P}$. However, (I_1, I_2) is not a three-valued model:

$$[\![\neg\mathsf{Q}]\!]((I_1, I_2), t) = ([\![\mathsf{Q}]\!](I_1, t), [\![\mathsf{Q}]\!](I_2, t))^{-1} = \mathbf{u}^{-1} = \mathbf{u} \not\leq_{tr} \mathbf{f} = [\![\mathsf{P}]\!]((I_1, I_2), t),$$

for all $t \in \mathcal{T}$.

We now focus on *stable HT-models*, and show that they correspond to our two-valued stable models.

Definition 3.9

Let $\mathcal{D}$ be a dataset, $\mathcal{P}$ be a program, and $I \in \mathcal{I}$. I is a *stable HT-model of* $\mathcal{D}$ *and* $\mathcal{P}$ if (I, I) is an HT-model of $\mathcal{D}$ and $\mathcal{P}$ and

$$I = \text{glb}_{\subseteq}(\{J \mid (J, I) \text{ is an HT-model of } \mathcal{D} \text{ and } \mathcal{P}\}). \tag{3.3}$$

Lemma 3.3

Let $\mathcal{D}$ be a dataset, $\mathcal{P}$ be a program, and $I, J \in \mathcal{I}$. If (I, J) is an HT-model, then I is a prefixpoint of $\Psi_1^{\mathcal{D},\mathcal{P}}(\cdot, J)$.

Proof. We have to show that $\Psi_1^{\mathcal{D},\mathcal{P}}(I, J) \subseteq I$. Let $t \in \mathcal{T}$, and $\mathsf{P}(\mathbf{s}) \in \Psi_1^{\mathcal{D},\mathcal{P}}(I, J)(t)$. Then, there are two possible cases:

1. There exists an interval δ such that $t \in \delta$ and $\mathsf{P}(\mathbf{s})@\delta \in \mathcal{D}$. In this case, it must hold that $\mathsf{P}(\mathbf{s}) \in I(t)$, because (I, J) is an HT-model of $\mathcal{D}$ and $\mathcal{P}$.

2. There exist $M \leftarrow B \in \mathsf{ground}(\mathcal{P})$ and $t' \in \mathcal{T}$ such that $[\![B]\!]((I, J), t') = \mathbf{t}$ and $(\mathsf{P}(\mathbf{s}), t) \in \mathcal{F}(M, t')$. By Definition 3.3, it is easy to see that $[\![B]\!]((I, J), t') = \mathbf{t}$ satisfies the antecedent of Item 1 of Definition 3.8. Since (I, J) is an HT-model, we can derive that $[\![M]\!](I, t') = \mathbf{t}$. By Proposition 3.1, we have that $[\![\mathsf{P}(\mathbf{s})]\!](I, t) = \mathbf{t}$, i.e., $\mathsf{P}(\mathbf{s}) \in I(t)$.

In both cases above we can conclude that $\mathsf{P}(\mathbf{s}) \in I(t)$. By the arbitrarity of $\mathsf{P}(\mathbf{s})$ and t, it holds that $\Psi_1^{\mathcal{D},\mathcal{P}}(I, J) \subseteq I$, i.e., I is a prefixpoint of $\Psi_1^{\mathcal{D},\mathcal{P}}(\cdot, J)$, as desired. □

Theorem 3.2

Let $\mathcal{D}$ be a dataset, $\mathcal{P}$ be a program, and $I \in \mathcal{I}$. I is a stable HT-model of $\mathcal{D}$ and $\mathcal{P}$ if and only if I is a two-valued stable model of $\mathcal{D}$ and $\mathcal{P}$.

Proof. Suppose I is a stable HT-model. To show that I is a two-valued stable model of $\mathcal{D}$ and $\mathcal{P}$, we have to prove that (I, I) is a stable fixpoint of $\Psi^{\mathcal{D},\mathcal{P}}$, i.e., $I = \text{lfp}(\Psi_1^{\mathcal{D},\mathcal{P}}(\cdot, I)) = \text{lfp}(\Psi_2^{\mathcal{D},\mathcal{P}}(I, \cdot))$. By Theorem 3.1, $\Psi_1^{\mathcal{D},\mathcal{P}}(I, I) = I$ if and only if $\Psi_2^{\mathcal{D},\mathcal{P}}(I, I) = I$, i.e., I is a fixpoint of $\Psi_1^{\mathcal{D},\mathcal{P}}(\cdot, I)$ if and only if I is a fixpoint of $\Psi_2^{\mathcal{D},\mathcal{P}}(I, \cdot)$. Since we consider only consistent pairs, i.e., pairs (J_1, J_2) with $J_1 \subseteq J_2$, it is sufficient to prove that $I =$

$\mathrm{lfp}(\Psi_1^{\mathcal{D},\mathcal{P}}(\cdot, I))$. We start by proving that I is a fixpoint of $\Psi_1^{\mathcal{D},\mathcal{P}}(\cdot, I)$, i.e., that $I = \Psi_1^{\mathcal{D},\mathcal{P}}(I, I)$. By Lemma 3.3, I is a pre-fixpoint of $\Psi_1^{\mathcal{D},\mathcal{P}}(\cdot, I)$, i.e., $\Psi_1^{\mathcal{D},\mathcal{P}}(I, I) \subseteq I$. We now show that $I \subseteq \Psi_1^{\mathcal{D},\mathcal{P}}(I, I)$. Since I is a stable HT-model, it holds that $I \subseteq J$ for all $J \in \mathcal{I}$ such that (J, I) is an HT-model of $\mathcal{D}$ and $\mathcal{P}$. Hence, it suffices to prove that $(\Psi_1^{\mathcal{D},\mathcal{P}}(I, I), I)$ is an HT-model of $\mathcal{D}$ and $\mathcal{P}$. Clearly, $\Psi_1^{\mathcal{D},\mathcal{P}}(I, I)$ is a model of $\mathcal{D}$ by Definition 3.6. It remains to verify the two conditions listed in Definition 3.8. Let $B = M_1 \wedge \ldots \wedge M_k \wedge \neg M_{k+1} \wedge \ldots \wedge \neg M_m$ such that $M \leftarrow B \in \mathsf{ground}(\mathcal{P})$, and $t \in \mathcal{T}$:

1. Suppose $[\![M_i]\!](\Psi_1^{\mathcal{D},\mathcal{P}}(I, I), t) = \mathbf{t}$ for all $i \in \{1, \ldots, k\}$ and $[\![M_j]\!](I, t) = \mathbf{f}$ for all $j \in \{k+1, \ldots, m\}$. Since $\Psi_1^{\mathcal{D},\mathcal{P}}(I, I) \subseteq I$, by Lemma 3.1, also $[\![M_i]\!](I, t)$ must be true for all $i \in \{1, \ldots, k\}$. Clearly, since all the M_i's do not contain negation, we also have that $[\![M_i]\!]((I, I), t) = \mathbf{t}$ for all $i \in \{1, \ldots, k\}$ and $[\![M_j]\!]((I, I), t) = \mathbf{f}$ for all $j \in \{k+1, \ldots, m\}$. Hence, we have that $[\![B]\!]((I, I), t) = \mathbf{t}$. By Definition 3.6, $\mathsf{P}(\mathbf{s}) \in \Psi_1^{\mathcal{D},\mathcal{P}}(I, I)(t')$ for all $t' \in \mathcal{T}$ and $\mathsf{P}(\mathbf{s})$ ground relational atoms such that $(\mathsf{P}(\mathbf{s}), t') \in \mathcal{F}(M, t)$. By Proposition 3.1, it follows that $[\![M]\!](\Psi_1^{\mathcal{D},\mathcal{P}}(I, I), t) = \mathbf{t}$, as desired.

2. Suppose $[\![M_i]\!](I, t) = \mathbf{t}$ for all $i \in \{1, \ldots, k\}$ and $[\![M_j]\!](I, t) = \mathbf{f}$ for all $j \in \{k+1, \ldots, m\}$. By the same reasoning performed at the point above, it follows that $[\![M]\!](\Psi_1^{\mathcal{D},\mathcal{P}}(I, I), t) = \mathbf{t}$. Since $\Psi_1^{\mathcal{D},\mathcal{P}}(I, I) \subseteq I$, by Lemma 3.1, we get that $[\![M]\!](I, t) = \mathbf{t}$, as desired.

Thus, we can conclude that $(\Psi_1^{\mathcal{D},\mathcal{P}}(I, I), I)$ is an HT-model. Since I is a stable HT-model, it follows that $\Psi_1^{\mathcal{D},\mathcal{P}}(I, I) = I$, i.e., I is a fixpoint of $\Psi_1^{\mathcal{D},\mathcal{P}}(\cdot, I)$.
We prove now that I is the *least* fixpoint of $\Psi_1^{\mathcal{D},\mathcal{P}}(\cdot, I)$. Let $J \subseteq I$ be a fixpoint of $\Psi_1^{\mathcal{D},\mathcal{P}}(\cdot, I)$. It suffices to show that $I \subseteq J$. If we show that (J, I) is an HT-model, we can again conclude by the minimality of I. Hence, we proceed to show that (J, I) is indeed an HT-model of $\mathcal{D}$ and $\mathcal{P}$. First, observe that since $J = \Psi_1^{\mathcal{D},\mathcal{P}}(J, I)$, J is a model of $\mathcal{D}$ by Definition 3.6. It remains to verify the two conditions listed in Definition 3.8. Let $B = M_1 \wedge \ldots \wedge M_k \wedge \neg M_{k+1} \wedge \ldots \wedge \neg M_m$ such that $M \leftarrow B \in \mathsf{ground}(\mathcal{P})$, and $t \in \mathcal{T}$:

1. Suppose $[\![M_i]\!](J, t) = \mathbf{t}$ for all $i \in \{1, \ldots, k\}$ and $[\![M_j]\!](I, t) = \mathbf{f}$ for all $j \in \{k+1, \ldots, m\}$. By Lemma 3.1, it holds that $[\![M_i]\!](I, t) = \mathbf{t}$ for all $i \in \{1, \ldots, k\}$, and $[\![M_j]\!](J, t) = \mathbf{f}$ for all $j \in \{k+1, \ldots, m\}$. Hence, we have that $[\![M_i]\!]((J, I), t) = \mathbf{t}$ for all $i \in \{1, \ldots, k\}$ and $[\![M_j]\!]((J, I), t) = \mathbf{f}$ for all $j \in \{k+1, \ldots, m\}$. It follows that $[\![B]\!]((J, I), t) = \mathbf{t}$. By Definition 3.6, $\mathsf{P}(\mathbf{s}) \in \Psi_1^{\mathcal{D},\mathcal{P}}(J, I)(t') = J(t')$ for all $t' \in \mathcal{T}$ and $\mathsf{P}(\mathbf{s})$ ground relational atoms such that $(\mathsf{P}(\mathbf{s}), t') \in \mathcal{F}(M, t)$. By Proposition 3.1, it follows that $[\![M]\!](J, t) = \mathbf{t}$, as desired.

2. Suppose $[\![M_i]\!](I, t) = \mathbf{t}$ for all $i \in \{1, \ldots, k\}$ and $[\![M_j]\!](I, t) = \mathbf{f}$ for all $j \in \{k+1, \ldots, m\}$. By the same reasoning performed at the point above, it follows that $[\![M]\!](J, t) = \mathbf{t}$. By Lemma 3.1, we get that $[\![M]\!](I, t) = \mathbf{t}$, as desired.

Thus, we can conclude that (J, I) is an HT-model. Since I is a stable HT-model, it follows that $J = I$, i.e., I is the least fixpoint of $\Psi_1^{\mathcal{D},\mathcal{P}}(\cdot, I)$.
Assume now that (I, I) is a three-valued stable model of $\mathcal{D}$ and $\mathcal{P}$. By Definition 3.7, it holds that I is a fixpoint of both $\Psi_1^{\mathcal{D},\mathcal{P}}(\cdot, I)$ and $\Psi_2^{\mathcal{D},\mathcal{P}}(I, \cdot)$, i.e., $\Psi^{\mathcal{D},\mathcal{P}}(I, I) = (I, I)$. By Proposition 3.6, (I, I) is an HT-model of $\mathcal{D}$ and $\mathcal{P}$. It remains to show that the equality (3.3) in Definition 3.9 holds. Let $J \subseteq I$ such that (J, I) is an HT-model of $\mathcal{D}$ and $\mathcal{P}$. We show that indeed J must be equal to I. It suffices to prove that $I \subseteq J$.
By Proposition 3.3 of Denecker et al. [2004], $\Psi_1^{\mathcal{D},\mathcal{P}}(\cdot, I)$ is a monotonic operator on the complete lattice $\langle \{H \mid H \subseteq I\}, \subseteq \rangle$. Hence, by the Tarski–Knaster Theorem Tarski [1955], the least fixpoint of $\Psi_1^{\mathcal{D},\mathcal{P}}(\cdot, I)$ coincide with its least pre-fixpoint. Since $\mathrm{lfp}(\Psi_1^{\mathcal{D},\mathcal{P}}(\cdot, I)) = I$ by hypothesis, it follows that I is also the least pre-fixpoint of $\Psi_1^{\mathcal{D},\mathcal{P}}(\cdot, I)$. By Lemma 3.3, J is a prefixpoint of $\Psi_1^{\mathcal{D},\mathcal{P}}(\cdot, I)$. Thus, it must be the case that $I \subseteq J$, as desired. □

Notice that rules of the form $\bot \leftarrow B$ are allowed in [Walega et al., 2024] to enforce the falsity of B in every (two-valued) stable model. However, this type of rules is not permitted in this chapter, as $\bot$ is not a head atom. To constraint B to be false in every (two-valued) stable model, we can instead perform a standard translation of the rule $\bot \leftarrow B$: we create a new atom P and replace $\bot \leftarrow B$ by $\mathsf{P} \leftarrow B \wedge \neg\mathsf{P}$. It is not difficult to show that the rule translation from $\bot \leftarrow B$ to $\mathsf{P} \leftarrow B \wedge \neg\mathsf{P}$ does not affect stable HT-models. Let us show this formally.

We call $\bot$-*rule* an expression of the form $\bot \leftarrow M_1 \wedge \ldots \wedge M_k \wedge \neg M_{k+1} \wedge \ldots \wedge \neg M_m$, where $m \geq k \geq 0$, and each M_i is a metric atom. A $\bot$-*program* is a finite set of safe rules and $\bot$-rules. We denote by $\mathcal{R}_{\mathcal{P}}^{\bot}$ the set of all $\bot$-rules of a $\bot$-program $\mathcal{P}$. We define a mapping $\cdot^{\bot}$ sending a $\bot$-program $\mathcal{P}$ to the program $\mathcal{P}^{\bot}$ defined by the following algorithm:

1. If $\mathcal{R}_{\mathcal{P}}^{\bot} = \emptyset$, then $\mathcal{P}^{\bot} := \mathcal{P}$;
2. Let P be a 0-arity relational atom that does not appear in $\mathcal{P}$;
3. Let $\bot \leftarrow B \in \mathcal{R}_{\mathcal{P}}^{\bot}$, and replace $\mathcal{P}$ by $\mathcal{P} \setminus \{\bot \leftarrow B\} \cup \{\mathsf{P} \leftarrow B \wedge \neg\mathsf{P}\}$;
4. Go back to step 1.

Lemma 3.4

Let $\mathcal{P}$ be a $\bot$-program, $\mathcal{D}$ a dataset, and $\boldsymbol{\mathcal{I}} \in \mathcal{A}_{\mathbb{J}}^{\mathsf{c}}$. If $\boldsymbol{\mathcal{I}}$ is an HT-model of $\mathcal{D}$ and $\mathcal{P}$, then $\boldsymbol{\mathcal{I}}$ is an HT-model of $\mathcal{D}$ and $\mathcal{P}^{\bot}$.

Proof. Let $\boldsymbol{\mathcal{I}}$ be an HT-model of $\mathcal{D}$ and $\mathcal{P}$. We only have to show that for each $r \in \mathsf{ground}(\mathcal{P}^{\bot})$, for all $t \in \mathcal{T}$, the conditions in Definition 3.8 hold. Since $\boldsymbol{\mathcal{I}}$ is an HT-model of $\mathcal{D}$ and $\mathcal{P}$, it is sufficient to check the conditions for every rule in $\mathsf{ground}(\mathcal{P}^{\bot}) \setminus \mathsf{ground}(\mathcal{P})$. Let $r \in \mathsf{ground}(\mathcal{P}^{\bot}) \setminus \mathsf{ground}(\mathcal{P})$. By definition of $\mathcal{P}^{\bot}$, $r = \mathsf{P} \leftarrow B \wedge \neg\mathsf{P}$ for some relational

atom P and B is a conjunction of ground atoms of the form $M_1 \wedge \ldots \wedge M_k \wedge \neg M_{k+1} \wedge \ldots \wedge \neg M_m$. Moreover, there exists a rule $r' \in \mathsf{ground}(\mathcal{P})$ such that $r' = \bot \leftarrow B$. Since $\mathcal{I}$ be an HT-model of $\mathcal{D}$ and $\mathcal{P}$, r' must satisfy the two conditions of Definition 3.8. Since the head of r' is evaluated to false by any interpretation at any time, the antecedents of both conditions in Definition 3.8 cannot hold for r'. Since the body of r is the conjunction of the body of r' with $\neg\mathsf{P}$, it is easy to see that the antecedents of both conditions in Definition 3.8 do not hold for r either. In other words, r trivially satisfies both conditions in Definition 3.8 at any $t \in \mathcal{T}$. Hence, $\mathcal{I}$ is an HT-model of $\mathcal{D}$ and $\mathcal{P}^\bot$, as desired. □

Theorem 3.3

Let $\mathcal{D}$ be a dataset, $\mathcal{P}$ be a $\bot$-program, and $I \in \mathcal{I}$. I is a stable HT-model of $\mathcal{D}$ and $\mathcal{P}$ if and only if I is a stable HT-model of $\mathcal{D}$ and $\mathcal{P}^\bot$.

Proof. Let

$$A := \{J \mid (J, I) \text{ is an HT-model of } \mathcal{D} \text{ and } \mathcal{P}\},$$
$$A^\bot := \{J \mid (J, I) \text{ is an HT-model of } \mathcal{D} \text{ and } \mathcal{P}^\bot\}.$$

($\Longleftarrow$) Suppose I is a stable HT-model of $\mathcal{D}$ and $\mathcal{P}^\bot$, i.e., $I = \mathrm{glb}_\subseteq(A^\bot)$ and $\{I\} = A^\bot$. By Lemma 3.4, $A \subseteq A^\bot$. If (I, I) is an HT-model of $\mathcal{D}$ and $\mathcal{P}$, then $I \in A \subseteq A^\bot = \{I\}$, i.e., $\{I\} = A = A^\bot$. Hence, we can reduce to showing that (I, I) is an HT-model of $\mathcal{D}$ and $\mathcal{P}$. I is a model of $\mathcal{D}$ by hypothesis, so we only have to prove that each rule in $\mathsf{ground}(\mathcal{P})$ satisfies the conditions in Definition 3.8 for all $t \in \mathcal{T}$. We proceed by contradiction. Suppose there exists a rule $r \in \mathsf{ground}(\mathcal{P})$ that breaks one of the conditions in Definition 3.8. Since (I, I) is an HT model of $\mathcal{D}$ and $\mathcal{P}^\bot$ by hypothesis, r can only be a $\bot$-rule, i.e., $r = \bot \leftarrow B$ where B is some conjunction of ground atoms of the form $M_1 \wedge \ldots \wedge M_k \wedge \neg M_{k+1} \wedge \ldots \wedge \neg M_m$. Hence, there exists some $t' \in \mathcal{T}$ such that the antecedent of one of the conditions in Definition 3.8 holds true, i.e., one of the following holds

1. $[\![M_i]\!](I, t') = \mathbf{t}$ for all $i \in \{1, \ldots, k\}$ and $[\![M_j]\!](I, t') = \mathbf{f}$ for all $j \in \{k+1, \ldots, m\}$;
2. $[\![M_i]\!](I, t') = \mathbf{t}$ for all $i \in \{1, \ldots, k\}$ and $[\![M_j]\!](I, t') = \mathbf{f}$ for all $j \in \{k+1, \ldots, m\}$.

By definition of $\mathcal{P}^\bot$, there exists $r' \in \mathsf{ground}(\mathcal{P}^\bot)$ such that $r' = \mathsf{P} \leftarrow B \wedge \neg\mathsf{P}$ for some ground relational atom P. Since (I, I) is a two-valued HT-model of $\mathcal{D}$ and $\mathcal{P}^\bot$, if Item 1 above holds, then the first condition in Definition 3.8 for r' forces $\mathsf{P} \in I(t')$. Similarly, if Item 2 above holds, then the second condition in Definition 3.8 for r' forces $\mathsf{P} \in I(t')$. However, since by construction of $\mathcal{P}^\bot$ the atom P appears only in r', it is easy to see that (J, I), where $J(t') = I(t') \setminus \{\mathsf{P}\}$ and $J(t) = I(t)$ for all $t \in \mathcal{T} \setminus \{t'\}$, is an HT-model of

$\mathcal{D}$ and $\mathcal{P}^{\perp}$. Since $J \subseteq I$ but $J \neq I$, I is not a lower bound of $A^{\perp}$, a contradiction. Hence, (I, I) is an HT-model of $\mathcal{D}$ and $\mathcal{P}$ and I is a stable HT-model of $\mathcal{D}$ and $\mathcal{P}$, as desired.

($\implies$) Suppose I is a stable HT-model of $\mathcal{D}$ and $\mathcal{P}$, i.e., $I = \text{glb}_{\subseteq}(A)$ and $\{I\} = A$. By Lemma 3.4, (I, I) is also an HT-model of $\mathcal{D}$ and $\mathcal{P}^{\perp}$. Hence, it suffices to show that there is no $J \neq I$ such that (J, I) is an HT-model of $\mathcal{D}$ and $\mathcal{P}^{\perp}$. We proceed by contradiction. Suppose there exists J strictly contained in I such that (J, I) is an HT model of $\mathcal{D}$ and $\mathcal{P}^{\perp}$. Since J is strictly contained in I, by the minimality of I as stable HT-model of $\mathcal{D}$ and $\mathcal{P}$, there must exist a rule $r \in \mathsf{ground}(\mathcal{P})$ and $t' \in \mathcal{T}$ such that (J, I) do not satisfy condition i in Definition 3.8 for r and t', for some $i \in \{1, 2\}$. Since (J, I) is an HT of $\mathcal{D}$ and $\mathcal{P}^{\perp}$, (J, I) satisfies both conditions of Definition 3.8 for every rule in $\mathsf{ground}(\mathcal{P}^{\perp})$ and every time $t \in \mathcal{T}$. Hence, r must be $\perp$-rule of the for $\perp \leftarrow B$. By definition of $\mathcal{P}^{\perp}$, there exist $r' \in \mathcal{P}^{\perp}$ of the form $\mathsf{P} \leftarrow B \wedge \neg \mathsf{P}$. Since r does not satisfy condition i of Definition 3.8 at time t' for (J, I), it holds that B satisfies the antecedent of condition i. Since (J, I) is an HT of $\mathcal{D}$ and $\mathcal{P}^{\perp}$, r' satisfies both conditions of Definition 3.8 at any $t \in \mathcal{T}$ for (J, I). In particular, r' satisfies condition i of Definition 3.8 at t' for (J, I). We then distinguish two possible cases regarding condition i of Definition 3.8 for r', t', and (J, I):

1. If $[\![\mathsf{P}]\!](I, t') = \mathit{true}$, the antecedent of condition i of Definition 3.8 is false, i.e., the condition itself is satisfied. However, P is not an atom of $\mathcal{P}$ by definition of $\mathcal{P}^{\perp}$. Hence, $\mathsf{P} \in I(t')$ contradicts the fact that I is a stable HT-model of $\mathcal{D}$ and $\mathcal{P}$.

2. If $[\![\mathsf{P}]\!](I, t') = \mathit{false}$, the antecedent of condition i of Definition 3.8 holds true. Then, since we get that $[\![\mathsf{P}]\!](J, t') = \mathit{true}$ or $[\![\mathsf{P}]\!](I, t') = \mathit{true}$, if $i = 1$ or $i = 2$, respectively. Both options contradicts $[\![\mathsf{P}]\!](I, t') = \mathit{false}$.

Since we have reached a contradiction for any possible value of $[\![\mathsf{P}]\!](I, t')$, we can conclude that there is no J strictly contained in I such that (J, I) is an HT model of $\mathcal{D}$ and $\mathcal{P}^{\perp}$. Hence, I is a stable HT-model of $\mathcal{D}$ and $\mathcal{P}^{\perp}$, as desired. □

Corollary 3.1

Let $\mathcal{D}$ be a dataset, $\mathcal{P}$ be a $\perp$-program, and $I \in \mathcal{I}$. I is a stable HT-model of $\mathcal{D}$ and $\mathcal{P}$ if and only if I is a two-valued stable model of $\mathcal{D}$ and $\mathcal{P}^{\perp}$.

Proof. Follows from Theorems 3.2 and 3.3. □

6 Conclusion

Using the solid framework of Approximation Fixpoint Theory, we have defined several semantics of interest for DatalogMTL$^{\neg}$: the Kripke-Kleene, well-founded, stable, and

supported model semantics. This was possible by defining a *consistent approximator*, i.e., an operator on the space of partial, three-valued interpretations. As we have shown, such an operator does not require any far-fetched choice; on the contrary, it is the most natural adaptation of Fitting's three-valued immediate consequence operator to the setting of DatalogMTL$^\neg$, showcasing, yet again, the simplicity, versatility, and power of the AFT formalism. In the final section, we have also proved that the stable models obtained via AFT are equivalent to the ones defined by Walega et al. [2024] through interpretations stemming from the logic of HT.

Since pairs of interpretations are used both in AFT and for the logic of HT, although in a different way, it is often the case that when the logic of HT is employed to define some *stable models*, AFT can provide an easier route to achieve the same results, while also giving access to other types of models for free. This might be true for the *metric stable models* defined by Becker et al. [2024], used to obtain the non-monotonic Metric Equilibrium Logic (MEL) starting from the Metric logic of HT, a temporal extension of the logic of HT. If this is indeed the case, AFT could provide a solid and reliable ground for a comparison between MEL developed by Becker et al. [2024], and the extension of Datalog MTL with negation-as-failure of Walega et al. [2024], discussed in this chapter.

It is also worth mentioning that AFT grants access to a wide body of theoretical results that hold for its applications. A line of work focused on stratification results [Vennekens et al., 2006; Bogaerts and Cruz-Filipe, 2021]. In particular, a notion of *stratifiable operator* was defined, allowing for the computation of the Kripke-Kleene, supported, well-founded, and stable models stratum by stratum. This seems to be equivalent to what was done for stratifiable DatalogMTL$^\neg$, i.e., for DatalogMTL$^\neg$ programs with no cyclic dependencies via negation [Cucala et al., 2021].

4 Category-Theory for Higher-Order AFT

This chapter is based on the conference paper *A Category-Theoretic Perspective on Higher-Order Approximation Fixpoint Theory* [Pollaci et al., 2024], and its journal extension [Pollaci et al., 2025]. This work has been presented at the 17th International Conference on Logic Programming and Non-monotonic Reasoning (LPNMR) in Dallas, Texas, USA. Parts of the preliminaries of [Pollaci et al., 2024] and [Pollaci et al., 2025], were left out from this chapter since they are already covered by the content of Chapter 2. Moreover, the notation has been adapted in compliance with the one previously introduced in Chapter 2. Section 6 was not included in the aforementioned publications.

1 Introduction

This chapter is motivated by a need to apply AFT to *higher-order* logic programming that arose in several contexts [Dasseville et al., 2015, 2016; Charalambidis et al., 2018]. An important issue that arises in this context is that using pairs of interpretations no longer allows for an obvious way to evaluate formulas in an approximation. Let us illustrate this with a brief example (for more detailed ones, we invite the reader to look at Examples 4.4, and 4.5). Consider a logic program in which a first-order predicate p and a second-order predicate Q are defined. Now assume that in the body of a rule, the atom $\mathsf{Q}(\mathsf{p})$ occurs. A

tuple (I, J) of interpretations in this case tells us whether $\mathsf{Q}(S)$ is true, false, unknown, or inconsistent, for any given set S. However, the interpretation of p via (I, J) is not a set, but a partially defined set, making it hard to evaluate expressions of the form $\mathsf{Q}(\mathsf{p})$. In other words, an approximation of the interpretation of Q has to take as argument not only sets, i.e., *exact* elements, but also partially defined sets, i.e., *approximate* elements, like the interpretation of p in this example. Thus, there is a need for a richer space of approximations where approximate objects can be applied to other approximate objects.

The above example and considerations suggest that spaces of approximations of higher-order objects should be defined inductively from lower-order ones, following the type hierarchy: we start by assigning a *base approximation space* to each type at the bottom of the hierarchy, and then, for each composite type $\tau_1 \to \tau_2$, we define its approximation space as *a certain class of functions* from the approximation space for τ_1 to the approximation space for τ_2, and so on. This method was heavily inspired by the approach used by Charalambidis et al. [2018] to obtain a generalization of the well-founded semantics for higher-order logic programs with negation. Notice that there are two major points in the construction above which are yet not defined: the base approximation spaces, and the class of functions we consider. The main question of this chapter is how to define them in a generic way that works in all applications of AFT.

We want to apply the same AFT techniques on approximation spaces at any hierarchy level, i.e., on base approximation spaces and the aforementioned sets of functions, which should thus have the same algebraic structure. In Category Theory (CT), the notion of *Cartesian closed category* captures this behavior. A category consists of a collection of *objects* and a collection of *morphisms*, i.e., relations between objects. For example, we can define the *category of complete lattices* as the one having complete lattices as objects, and monotonic functions as morphisms. The objects of a Cartesian closed category $\mathbf{C}$ satisfy a property that can be intuitively understood as follows: *if A and B are two objects of $\mathbf{C}$, then the set of morphisms from A to B is also an object of $\mathbf{C}$*. Hence, if the base approximation spaces are objects of a Cartesian closed category, then the category contains the full hierarchy of spaces we are aiming for. We call such a Cartesian closed category an *approximation category* and denote it by **Approx**.

In this category-theoretic framework, the questions on the nature of the base approximation spaces and the class of functions reduce to defining the objects and the morphisms of **Approx**. Clearly, this depends on the application we want to use AFT for. Different applications imply different higher-order languages, with different types, and possibly different versions of AFT (standard AFT [Denecker et al., 2000], consistent AFT [Denecker et al., 2004], or other extensions [Charalambidis et al., 2018]). To formalize this, and unify different AFT accounts, we develop the notion of an *approximation system*. Once a language and the semantics of its types are fixed, we can choose an approximation system that consists, among other things, of a Cartesian closed category **Approx**, equipped

with a function App associating the semantics of a type to an approximation space in **Approx**. The approximation system also determines which elements of the approximation spaces are *exact*, i.e., which elements approximate exactly one element of the semantics of a type, and, for every type, it provides a projection from the exact elements to the objects they represent in the corresponding semantics. This is non-trivial for higher-order approximation spaces, and it is indeed fundamental to obtain a sensible account for AFT for higher-order definitions.

In recent work, a stable semantics for higher-order logic programs was defined building on consistent AFT [Bogaerts et al., 2024]. In that work, the approach taken to evaluate an expression of the form Q(p), instead of applying an approximate interpretation for Q to an approximate interpretation for p, is to apply the approximate interpretation for Q to *all* exact interpretations for p that are still possible, and returning the least precise approximation of all the results. What this means in effect is that some sort of *ultimate construction* [Denecker et al., 2004] is used; this has also been done in other extensions of logic programming [Pelov et al., 2007; Dasseville et al., 2016]. We expand on this in Section 6. Bogaerts et al. [2024] also pointed out a rather counterintuitive behaviour of the well-founded semantics defined in the work of Charalambidis et al. [2018], namely that even for simple non-recursive programs, the well-founded model might not assume expected values (leaving all atoms unknown). It is important to mention, though, that this counterintuitive behaviour is caused solely by the treatment of (existentially) quantified variables and not by the algebraic theory (which is the focus of the current chapter). Finally, it is interesting to mention that another paper Charalambidis and Rondogiannis [2023] joined CT and AFT, albeit with different perspective and aim: while we treat the whole set of possible approximation spaces as a category to apply any account of AFT to higher-order definitions, Charalambidis and Rondogiannis [2023] view the approximation spaces themselves as categories to provide a novel version of standard AFT.

In short, the main contributions of this chapter are as follows:

1. We generalize the work of Charalambidis et al. [2018] to a category-theoretic setting. In doing so, we shed light on the general principles underlying their constructions for higher-order logic programing and make their construction applicable to arbitrary current and future non-monotonic reasoning formalism captured by AFT.

2. We improve the work of Charalambidis et al. [2018]. In particular, we define a new approximator, which provides the expected well-founded semantics; and we study the concept of *exactness*, previously missing, allowing the use of the theory to define exact *stable models* instead of focusing purely on the well-founded model.

It is also worth remarking that the generality of the CT environment allows to cover various accounts of AFT, like the ones of Denecker et al. [2000], Denecker et al. [2004],

and Charalambidis et al. [2018], and possibly others. Different versions of AFT are suitable for various situations and cater to specific applications. For instance, consistent AFT [Denecker et al., 2004] utilizes a three-valued logic and provides a rather intuitive and easily applicable notion of approximation. On the other hand, standard AFT [Denecker et al., 2000] employs a four-valued approach, with *inconsistent* elements, as presented earlier in this introduction. This can sometimes be more difficult to use in applications, but shows several advantages from the formal, mathematical standpoint: having a square lattice, composed of both consistent and inconsistent elements, provides symmetry and allows for duality results to be derived, simplifying the proofs of the fundamental theorems at the core of this version of AFT. It is hence valuable to obtain a framework that covers as many accounts of AFT as possible.

The rest of this chapter is structured as follows. In Section 2, we provide an overview of the elements of [Charalambidis et al., 2018] that we aim at modifying, and the fundamental concepts from CT that we use[1]. Section 3 presents the novel definitions of approximation system, with the category **Approx**, and of exact elements of an approximation space. In Section 4, we show that the square lattices form a Cartesian closed category that can be chosen as **Approx** for standard AFT. With a suitable choice of App and exact elements, depending on the application at hand, we obtain an approximation system that recovers the framework of standard AFT and extends it to higher-order objects. This section can be skipped by the reader interested uniquely in the AFT version of Charalambidis et al. [2018], which is addressed in the following section. In Section 5, we apply the novel categorical framework to Charalambidis et al. [2018]. First, in Subsection 5.1, we show that the approximation spaces from Charalambidis et al. [2018] form a Cartesian closed category. Second, in Subsection 5.2 we define an approximation system that enables us to reconstruct in a simple way (using the general principles outlined above) the semantic elements defined ad-hoc by Charalambidis et al. [2018]. At the same time, the definition of such approximation system also resolves a question that was left open in that work. Namely, what we get now is a clear definition for exact higher-order elements, and, in particular, this allows to determine when a model of a program is two-valued (see Example 4.3). We proceed with Subsection 5.3, where we present the new approximator that adjusts the behaviour of the well-founded semantics of Charalambidis et al. [2018] for programs with existential quantifiers in the body of rules. We close the subsection with two examples of logic programs in which we need to apply an approximate object on another approximate object. In Section 6, we compare the approaches taken in [Charalambidis et al., 2018] and [Bogaerts et al., 2024] to extend consistent AFT for higher-order LP. Finally, we draw our conclusions in Section 7.

[1]For preliminaries on AFT we refer back to Section 2 of Chapter 2 of this thesis.

2 Preliminaries

In this section, we provide an introduction to the formal concepts we utilize throughout this chapter[2]. We divide the content into two subsections. In the first (Section 2.1), we briefly present the parts of the work of Charalambidis et al. [2018] that we aim to modify in Section 5. In the second subsection (Section 2.2), we present the notions of Category Theory (CT) we need, with the definition of Cartesian closed category being the key concept. For further information on CT, we refer to the book by Riehl [2017].

2.1 Consistent AFT for Higher-Order LP

Charalambidis et al. [2018] already proposed a way to deal with higher-order logic programs via an extension of consistent AFT. However, as already highlighted by Bogaerts et al. [2024], the work of Charalambidis et al. [2018] had some hidden problematic features. They can be summarised as follows:

1. *The Approximator:* the well-founded semantics obtained via the approximator defined by Charalambidis et al. [2018] does not behave as expected when an existential quantifier occurs in the body of a rule. Take for instance the logic program with just the simple rule $\mathsf{p} \leftarrow \mathsf{R} \ \wedge \neg\mathsf{R}$, where R is a variable ranging over the booleans $\{\mathbf{f}, \mathbf{t}\}$. If we naively ground such program, we obtain $\mathsf{p} \leftarrow \mathbf{f} \wedge \mathbf{t}$ and $\mathsf{p} \leftarrow \mathbf{t} \wedge \mathbf{f}$, and p would clearly be evaluated as false. However, the approach adopted by Charalambidis et al. [2018] uses *approximated* elements. In particular, variables of type boolean range over $\{\mathbf{f}, \mathbf{t}, \mathbf{u}\}$. In more detail, for the logic program $\mathsf{p} \leftarrow \mathsf{R} \ \wedge \neg\mathsf{R}$, the approximator assignes to p the least upper bound of the body $\mathsf{R} \ \wedge \neg\mathsf{R}$, with R ranging over $\{\mathbf{f}, \mathbf{t}, \mathbf{u}\}$. Since such least upper bound is computed with respect to the *truth order* $\mathbf{f} \leq \mathbf{u} \leq \mathbf{t}$, the predicate p is assigned the value $\mathrm{lub}(\{\mathbf{t} \wedge \mathbf{f}, \mathbf{f} \wedge \mathbf{t}, \mathbf{u} \wedge \mathbf{u}\}) = \mathbf{u}$ under the well-founded semantics. This contradicts the more intuitive and standard two-valued approach via grounding, which assigns $\mathbf{f}$ to p. In a way, allowing the existentially quantified variable to vary over all approximated elements seems to unnecessarily increase (w.r.t. the truth order) the value of the defined predicate, when evaluated under the well-founded semantics.

2. *The Notion of Exactness:* the work of Charalambidis et al. [2018] lacks the notion of exactness for higher-order objects, which is fundamental in the context of AFT and rather non-trivial in the higher-order setting: exactness allows to recognize whether an approximated object, i.e., a pair (x, y) in the square lattice, represents just one *real* element of the lattice, and, in particular, when a model is two-valued. In other

[2]For preliminaries on AFT we refer back to Section 2 of Chapter 2 of this thesis.

words, having such concept makes it possible to study not just the well-founded models, but also the stable ones.

In Section 5, we will show how we can use our novel concepts in the framework of Charalambidis et al. [2018] to solve the issues listed above.

2.2 Category Theory

CT studies mathematical structures and the relations between them, through the notion of a *category*. Intuitively, a category $\mathbf{C}$ consists of a collection $\mathrm{Ob}(\mathbf{C})$ of *objects* and a collection $\mathrm{Mor}(\mathbf{C})$ of relations, called *morphisms*, between objects, satisfying some basic properties: every morphism f has a *domain* $s(f)$ and a *codomain* $t(f)$, morphisms can be composed, and so on.

Definition 4.1

A *category* $\mathbf{C}$ consists of

- a collection of *objects* $\mathrm{Ob}(\mathbf{C})$,
- a collection of *morphisms* $\mathrm{Mor}(\mathbf{C})$,
- for every morphism $f \in \mathrm{Mor}(\mathbf{C})$, an object $s(f)$ called the *source* (or *domain*) of f, and an object $t(f)$ called the *target* (or *codomain*) of f,
- for every object $X \in \mathrm{Ob}(\mathbf{C})$, a morphism id_X called the *identity morphism*,
- for every two morphisms $f, g \in \mathrm{Mor}(\mathbf{C})$ with $t(f) = s(g)$, a morphism $g \circ f$, called their *composite*,

such that

- for all $f, g \in \mathrm{Mor}(\mathbf{C})$ such that $t(f) = s(g)$, $s(g \circ f) = s(f)$
- for all $f, g \in \mathrm{Mor}(\mathbf{C})$ such that $t(f) = s(g)$, $t(g \circ f) = t(g)$,
- for all $X \in \mathrm{Ob}(\mathbf{C})$, $s(id_X) = t(id_X) = X$,
- for all $f, g, h \in \mathrm{Mor}(\mathbf{C})$ such that $t(f) = s(g)$ and $t(g) = s(h)$, $(h \circ g) \circ f = h \circ (g \circ f)$,
- for all $X, Y \in \mathrm{Ob}(\mathbf{C})$ and for all $f \in \mathrm{Mor}(\mathbf{C})$ such that $s(f) = X$ and $t(f) = Y$, $f \circ id_X = f$ and $id_Y \circ f = f$.

Given two morphisms $f, g \in \mathrm{Mor}(\mathbf{C})$ with $t(f) = s(g)$, we say f is *composable* with g. Given a morphism $f \in \mathrm{Mor}(\mathbf{C})$, we say a morphism $g \in \mathrm{Mor}(\mathbf{C})$ is a *left-inverse* (resp. *right-inverse*) of f if $g \circ f = id_{s(f)}$ (resp. $f \circ g = id_{t(f)}$); we say g is a *2-sided inverse* of f if g is both a right- and a left-inverse of f. A morphism is called an *isomorphism* if

there exists a 2-sided inverse of it. Two objects $X, Y \in \mathrm{Ob}(\mathbf{C})$ are *isomorphic* (denoted as $X \cong Y$) if there exists an isomorphism $f : X \to Y$.

In this chapter, objects will always be certain ordered sets, and morphisms will be monotonic functions. In the same way as morphisms between objects encode relations within a category, a morphism of categories, called a *functor*, describes the relation between two categories.

Definition 4.2

Let $\mathbf{C}, \mathbf{D}$ be two categories. A *functor* $F : \mathbf{C} \to \mathbf{D}$ consists of a function $F_0 : \mathrm{Ob}(\mathbf{C}) \to \mathrm{Ob}(\mathbf{D})$ between the classes of objects, and a function $F_1 : \mathrm{Mor}(\mathbf{C}) \to \mathrm{Mor}(\mathbf{D})$, such that:

- it respects source and target of morphisms, i.e., $s(F_1(f)) = F_0(s(f))$ and $t(F_1(f)) = F_0(t(f))$ for all $f \in \mathrm{Mor}(\mathbf{C})$;
- it respects identity morphisms, i.e., $F_1(id_X) = id_{F_0(X)}$ for all $X \in \mathrm{Ob}(\mathbf{C})$;
- it respects composition, i.e., $F_1(g \circ f) = F_1(g) \circ F_1(f)$ for all $f, g \in \mathrm{Mor}(\mathbf{C})$.

Let **Set** be the category of sets, i.e., objects are sets, and morphisms are functions between sets. For each $X, Y \in \mathrm{Ob}(\mathbf{C})$, we denote by $\hom_{\mathbf{C}}(X, Y)$ the set of morphisms of $\mathbf{C}$ with domain X and codomain Y. When $\mathbf{C} = \mathbf{Set}$, we write $(X \to Y)$ in place of $\hom_{\mathbf{Set}}(X, Y)$, i.e., $(X \to Y)$ is the set of all functions from X to Y.

Definition 4.3

A functor $F : \mathbf{C} \to \mathbf{D}$ is

- *full* if for each $X, Y \in \mathrm{Ob}(\mathbf{C})$, the map $F_1|_{\hom_{\mathbf{C}}(X,Y)} : \hom_{\mathbf{C}}(X, Y) \to \hom_{\mathbf{D}}(F_0(X), F_0(Y))$ is surjective,
- *faithful* if for each $X, Y \in \mathrm{Ob}(\mathbf{C})$, the map $F_1|_{\hom_{\mathbf{C}}(X,Y)} : \hom_{\mathbf{C}}(X, Y) \to \hom_{\mathbf{D}}(F_0(X), F_0(Y))$ is injective,
- *embedding* if F is faithful and F_0 is injective.

The domain of a full embedding $F : \mathbf{C} \to \mathbf{D}$ is called a *full subcategory* of the codomain (denoted as $\mathbf{C} \subseteq \mathbf{D}$).

It is easy to see that we can define a category **POSet** with objects the posets, and as morphisms the monotonic functions between posets. We denote by **CPO**, **CJSLat**, and **CLat** the full subcategories of **POSet** with objects the cpo's, the complete join semilattices, and the complete lattices, respectively. Clearly, it also holds that **CLat** $\subseteq$ **CJSLat** $\subseteq$ **CPO** $\subseteq$ **POSet** $\subseteq$ **Set**.

We are interested in inductively building *approximation spaces* for higher-order concepts starting from *base* ones. To be able to perform this construction, we need the approximation spaces to belong to a *Cartesian closed* category, i.e., a category **C** with a *terminal object*, *products*, and *exponentials*.

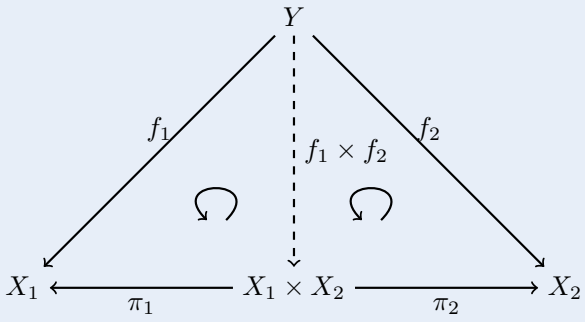
Definition 4.4

$T \in \mathrm{Ob}(\mathbf{C})$ is *terminal* if for each $X \in \mathrm{Ob}(\mathbf{C})$ there exists a unique morphism $f : X \to T$.

For instance, the poset with one element and trivial order is the terminal object of **POSet**, **CPO**, **CJSLat**, and **CLat**.

Definition 4.5

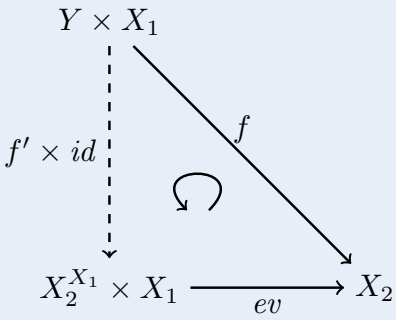
Let $X_1, X_2 \in \mathrm{Ob}(\mathbf{C})$. A *product* of X_1 and X_2 is an object of **C**, denoted by $X_1 \times X_2$, equipped with two morphisms $\pi_1 : X_1 \times X_2 \to X_1$ and $\pi_2 : X_1 \times X_2 \to X_2$, called *first*, and *second projection* respectively, such that for any $Y \in \mathrm{Ob}(\mathbf{C})$ and any morphisms $f_1 : Y \to X_1$ and $f_2 : Y \to X_2$, there exists a unique morphism $f_1 \times f_2 : Y \to X_1 \times X_2$ such that $\pi_1 \circ (f_1 \times f_2) = f_1$ and $\pi_2 \circ (f_1 \times f_2) = f_2$.



In **POSet**, and analogously for **CPO**, **CJSLat**, and **CLat**, the product of two objects $\mathcal{P}_1$ and $\mathcal{P}_2$ is the Cartesian product of P_1 and P_2 equipped with the *product order*, i.e., for all $(x_1, y_1), (x_2, y_2) \in P_1 \times P_2$, $(x_1, y_1) \leq (x_2, y_2)$ if and only if $x_1 \leq x_2$ and $y_1 \leq y_2$. Note that, by abuse of notation, we often denote orders on different sets by simply $\leq$: whenever we write $x \leq y$ for some x and y in a poset $\mathcal{P}$, $\leq$ refers to the order relation defined on $\mathcal{P}$. The projections π_1 and π_2 are given by the usual Cartesian projections.

Definition 4.6

Let $X_1, X_2 \in \mathrm{Ob}(\mathbf{C})$. An *exponential* of X_1 and X_2 is an object of $\mathbf{C}$, denoted by $X_2^{X_1}$, equipped with a morphism $ev : X_2^{X_1} \times X_1 \to X_2$, called the *evaluation*, such that for any $Y \in \mathrm{Ob}(\mathbf{C})$ and any morphism $f : Y \times X_1 \to X_2$, there exists a unique morphism $f' : Y \to X_2^{X_1}$ such that $ev \circ (f' \times id) = f$.



In **POSet**, and analogously for **CPO**, **CJSLat**, and **CLat**, the exponential $\mathcal{P}_2^{P_1}$ is the set of monotonic functions from P_1 to P_2 equipped with the pointwise order induced by the order on P_2, i.e., for all monotonic functions $f_1, f_2 \in (P_1 \to P_2)$, $f_1 \leq f_2$ if and only if for all $x \in P_1$, $f_1(x) \leq f_2(x)$. The evaluation ev is given by the usual function evaluation, i.e., $ev(f, x) = f(x)$.

Definition 4.7

A category $\mathbf{C}$ is *Cartesian closed* if it has a terminal object, and for each $X_1, X_2 \in \mathrm{Ob}(\mathbf{C})$, there exist a product $X_1 \times X_2 \in \mathrm{Ob}(\mathbf{C})$ and an exponential $X_2^{X_1} \in \mathrm{Ob}(\mathbf{C})$.

By our prior observations, it follows that **POSet**, **CPO**, **CJSLat**, and **CLat** are all Cartesian closed.

In the context of AFT, we are often interested in the space of interpretations over a possibly infinite vocabulary of a logic program. In order to include this, we need another notion from CT, namely a generalized version of the categorical product for (possibly infinite) families of objects.

Definition 4.8

Let $\{X_i\}_{i\in I}$ be a family of objects of a category $\mathbf{C}$ indexed by I. The *(generalized) product* of the family $\{X_i\}_{i\in I}$ is an object of $\mathbf{C}$, denoted by $\Pi_{i\in I}X_i$, equipped with morphisms $\pi_i : \Pi_{i\in I}X_i \to X_i$, called the *i-th projection*, such that for all $Y \in \mathrm{Ob}(\mathbf{C})$ and for all families of morphisms $\{\varphi_i : Y \to X_i\}_{i\in I}$ indexed by I, there exists a unique $\psi : Y \to \Pi_{i\in I}X_i$ such that for all $i \in I$ it holds that $\varphi_i = \pi_i \circ \psi$.

We say that a category $\mathbf{C}$ *has generalized products* if for all families $\{X_i\}_{i\in I}$ of objects of $\mathbf{C}$ the product $\Pi_{i\in I}X_i \in \mathrm{Ob}(\mathbf{C})$ exists. Clearly, if $\mathbf{C}$ is Cartesian closed and the index I is finite, this product always exists; but this is not always the case for infinite I. However, we will show in the remainder of this subsection that every full subcategory of **POSet** has generalized products (Proposition 4.2). We wish to warn the reader that the following category-theoretic notions are only meant to support the proofs of Propositions 4.2 and 4.3, and will not be used in the next sections of this chapter.

The generalized product is a special case of a very common construction in CT, called the *limiting cone*, or simply *limit*. We proceed with the definitions leading to the concept of limit.

Definition 4.9

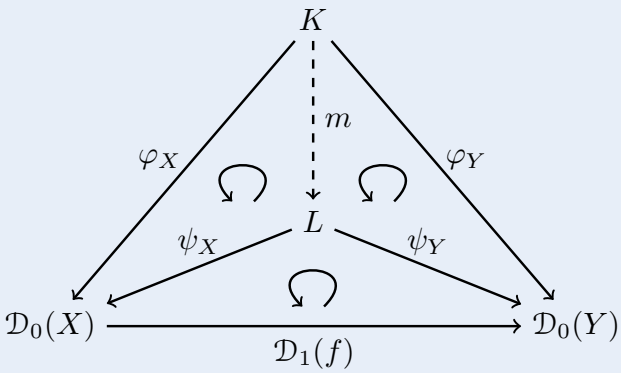
Let $\mathbf{C}$ and $\mathbf{S}$ be categories. A *diagram* $\mathcal{D}$ of shape $\mathbf{S}$ in $\mathbf{C}$ is a functor from $\mathbf{S}$ to $\mathbf{C}$.

Definition 4.10

Let $\mathcal{D} : \mathbf{S} \to \mathbf{C}$ be a diagram of shape $\mathbf{S}$ in a category $\mathbf{C}$. A *cone* to $\mathcal{D}$ is an object $C \in \mathrm{Ob}(\mathbf{C})$ together with a family $\Psi := \{\psi_X : C \to \mathcal{D}_0(X)\}_{X\in\mathrm{Ob}(\mathbf{S})}$ of morphisms indexed by the objects of $\mathbf{S}$ such that $\mathcal{D}_1(f) \circ \psi_X = \psi_Y$ for all morphisms $f : X \to Y \in \mathrm{Mor}(\mathbf{C})$.

Definition 4.11

Let $\mathcal{D} : \mathbf{S} \to \mathbf{C}$ be a diagram of shape $\mathbf{S}$ in a category $\mathbf{C}$. A *limiting cone* or *limit* of $\mathcal{D}$ is a cone (L, Ψ) to $\mathcal{D}$ such that for every cone (K, Φ) to $\mathcal{D}$ there exists a unique morphism $m : K \to L$, called the *mediating morphism*, such that $\psi_X \circ m = \varphi_X$ for all $X \in \mathrm{Ob}(\mathbf{S})$.



A diagram $\mathcal{D} : \emptyset \to \mathbf{C}$ may admit no limit. If a limit exists, then it is unique up to isomorphism. We may then refer to it as *the* limit of $\mathcal{D}$.

Example 4.1

We denote by $\mathbf{O}$ the *empty category*, i.e., the category with $\mathrm{Ob}(\mathbf{O}) = \mathrm{Mor}(\mathbf{O}) = \emptyset$. Notice that, for each category $\mathbf{C}$, there exists a unique functor from $\mathbf{O}$ to $\mathbf{C}$. If the limit of the functor $\mathcal{D} : \mathbf{O} \to \mathbf{C}$ exists, its object is the terminal object of $\mathbf{C}$.

Definition 4.12

A functor $F : \mathbf{C} \to \mathbf{D}$ *reflects all limits* if for all diagrams $\mathcal{D}$ in $\mathbf{C}$, and for all cones (C, Ψ) to $\mathcal{D}$ such that $(F_0(C), F_1(\Psi))$ is a limiting cone to $F \circ \mathcal{D}$, (C, Ψ) is a limiting cone of $\mathcal{D}$.

Proposition 4.1

A full and faithful functor reflects all limits.

Proof. Lemma 3.3.5 in [Riehl, 2017]. □

We are finally able to prove that every full subcategory of **POSet** has generalized products.

Proposition 4.2

If $\mathbf{C} \subseteq \mathbf{POSet}$, then $\mathbf{C}$ has generalized products.

Proof. By Definition 4.3, there exists a full embedding $F : \mathbf{C} \to \mathbf{POSet}$. By Proposition 4.1, F reflects all limits. Moreover, it is clear that **POSet** has generalized products. Since generalized products are a type of limit, we conclude that $\mathbf{C}$ has generalized products. □

In a full subcategory of **POSet** we can rewrite a generalized product as a poset of functions.

Proposition 4.3

Let $\mathbf{C}$ be a full subcategory of **POSet**, $X \in \mathrm{Ob}(\mathbf{Set})$, and $Y \in \mathrm{Ob}(\mathbf{C})$. Then there exists an isomorphism $(X \to Y) \cong \Pi_{x \in X} Y$ in $\mathbf{C}$.

Proof. By Proposition 4.2, $\Pi_{x \in X} Y \in \mathrm{Ob}(\mathbf{C})$. Moreover, there exists a bijection $\varphi : (X \to Y) \to \Pi_{x \in X} Y$, defined by $\varphi(f) = (f(a))_{a \in X}$, with inverse $\varphi^{-1} : \Pi_{x \in X} Y \to (X \to Y)$, defined by, for all $a \in X$, $\varphi^{-1}((y_x)_{x \in X})(a) = y_a$. Because of the definition of pointwise order and product order, it is immediate to show that both φ and φ^{-1} preserve the orders, i.e., they are monotonic. □

3 The Approximation System

In this section, we introduce the notions of *approximation category* and of *approximation system*, which constitute the core of the theoretical framework for AFT we developed.

Let $\mathfrak{L}$ be a higher-order language based on a hierarchy of types $\mathbb{H}$ comprising of *base types* τ, and two kinds of composite types: *product types* $\Pi_{i \in I} \tau_i$, and *morphism types* $\tau_1 \to \tau_2$. For instance, a base type could be the boolean type o or the type ι of individuals, whereas in the composite types we may find the type $\iota \to o$, which is the type of unary first-order predicates. We denote by $\mathcal{B}_{\mathbb{H}}$ the set of base types of $\mathbb{H}$. For the sake of simplicity, we omit the subscript of $\mathcal{B}$ when it is clear from the context of use.

We associate to each type τ of $\mathcal{B}_{\mathbb{H}}$, an object $E_\tau \in \mathrm{Ob}(\mathbf{POSet})$, and we define inductively for all $\{\tau_i\}_{i \in I} \subseteq \mathbb{H}$, $E_{\Pi_{i \in I} \tau_i} = \Pi_{i \in I} E_{\tau_i}$, and for all $\tau_1, \tau_2 \in \mathbb{H}$, $E_{\tau_1 \to \tau_2} = (E_{\tau_1} \to E_{\tau_2})$. The object E_τ is called the *semantics of* τ. For example, if the semantics of the boolean type o is chosen to be $E_o := \{\mathbf{f}, \mathbf{t}\}$ with the standard truth ordering, then the semantics for type $o \to o$ is the poset of functions from E_o to E_o.

In many applications of AFT, we are ultimately interested in the space of interpretations, which associate to each symbol of a vocabulary, an element of the semantics of the type of such symbol. It follows that an interpretation can be seen as a tuple of elements of different semantics. In more detail, given a vocabulary V, we can consider the product type $\tau = \Pi_{s \in V} \tau(s)$, where $\tau(s)$ is the type of the symbol s. Then, the space of interpretations for the vocabulary V coincides with the semantics $E_\tau = \Pi_{s \in V} E_{\tau(s)}$.

We have so far defined the semantics of all the base types and the composite ones constructed from them. Notice that, it is often not necessary to define the spaces of approximations for all such semantics E_τ, which are infinitely many. Because of the nature of our formalism, we can easily restrict the set of types we take into account: we can fix a subset $\mathbb{K} \subseteq \mathbb{H}$ of types, and focus our attention onto the set $S_\mathbb{K}$ of semantics defined as follows:

- for all $\tau \in \mathbb{K}$, $E_\tau \in S_\mathbb{K}$,
- if $E_{\tau_1 \to \tau_2} \in S_\mathbb{K}$, then $E_{\tau_2} \in S_\mathbb{K}$,
- if $E_{\Pi_{i \in I} \tau_i} \in S_\mathbb{K}$, then $E_{\tau_i} \in S_\mathbb{K}$ for all $i \in I$.

We will dive deeper into this matter in Section 5 where we present applications of our framework. We denote by $\mathcal{B}_\mathbb{K}$ the set of base types of $\mathbb{H}$ belonging to $\mathbb{K}$.

The notion of *approximation system* (Definition 4.13) together with what follows in this section, provide a general framework in which the techniques of AFT can be applied on higher-order languages. Before stating the, rather lengthy, definition of an approximation system, we provide an intuitive understanding of its components.

For each $E_\tau \in S_\mathbb{K}$, we shall consider a corresponding space $\mathrm{App}(E_\tau)$, called the *approximation space* of E_τ, whose elements approximate the elements of E_τ. Hence, we define a Cartesian closed full subcategory of **CPO**, denoted by **Approx**, and a map $\mathrm{App} : S_\mathbb{K} \to \mathrm{Ob}(\mathbf{Approx})$ encoding such correspondence. The fact that $\mathbf{Approx} \subseteq \mathbf{CPO}$ allows us to apply the Knaster-Tarski theorem on the approximation spaces, and guarantees the existence of generalized products (Proposition 4.2). Notice that, even though we fixed a mapping App between the set $S_\mathbb{K}$ and the objects of **Approx**, there is, so far, no relation between the elements of E_τ and those of $\mathrm{App}(E_\tau)$. The approximation space $\mathrm{App}(E_\tau)$ is meant to approximate the elements of E_τ. In particular, we want the order $\leq$ on $\mathrm{App}(E_\tau)$, which we call a *precision order*, to encode the approximating nature of $\mathrm{App}(E_\tau)$ for E_τ: intuitively, for all $a, b \in E_\tau$, $a \leq b$ if a is *less precise* than b, i.e., if an element $e \in E_\tau$ is approximated by b, then e is also approximated by a. In the context of AFT, of particular interest are the elements of $\mathrm{App}(E_\tau)$ which approximate just one element, called the *exact* elements. Thus, in the definition of approximation system that we are about to give, for every base type $\tau \in \mathcal{B}_\mathbb{K}$, we fix a set $\mathcal{E}_\tau$ of exact elements of $\mathrm{App}(E_\tau)$, and a function $\natural^0_\tau : \mathcal{E}_\tau \to E_\tau$, which associates each exact element to the unique element of E_τ it represents.

To obtain a sensible framework, it is fundamental to carefully define the sets of exact elements and a projection that associates each exact element to the object it represents. Hence, we impose conditions on the possible choices of the sets $\mathcal{E}_\tau$ and the functions $\mathfrak{p}^0_\tau$, for $\tau \in \mathcal{B}_\mathbb{K}$. Since an exact element of $\mathrm{App}(E_\tau)$ approximates a single element of the semantics E_τ, if both a and b are exact and one is more precise than the other, then they should represent the same element, i.e., $\mathfrak{p}^0_\tau(a) = \mathfrak{p}^0_\tau(b)$ (Item 4b in Definition 4.13). This requirement also hints at a very important fact: the definition of approximation system allows for the existence of multiple exact elements of $\mathrm{App}(E_\tau)$ representing the *same* element of E_τ. Because of this possible multitude of exact representatives, we want to have, for each element $e \in E_\tau$, a natural choice for a representative in the approximation space $\mathrm{App}(E_\tau)$. This is why, for each element $e \in E_\tau$, we require that the greatest lower bound of all the exact elements representing e exists, is exact, and represents e (Item 4c in Definition 4.13). Lastly, we add one more condition on exact elements to accommodate several existing versions of AFT. In consistent AFT [Denecker et al., 2004], exact elements are maximal, while in standard AFT, this is not the case, and there are elements beyond exact ones. We require that either the exact elements are maximal, or we can take arbitrary joins in the approximation spaces (Item 3b in Definition 4.13). This last condition will later allow for a generalization of both $\mathcal{E}_\tau$ and $\mathfrak{p}^0_\tau$ to any type τ of $\mathbb{H}$, satisfying properties analogous to the ones required for the base types counterparts (Propositions 4.5 and 4.6).

We are now ready to state the definition of an approximation system. We write $f^{-1}(y)$ for the *preimage* of an element $y \in Y$ via a function $f : X \to Y$, i.e., $f^{-1}(y) = \{x \mid x \in X, f(x) = y\} \subseteq X$. Recall that given two posets $\mathcal{P}_1, \mathcal{P}_2 \in \mathrm{Ob}(\mathbf{POSet})$, we denote by $(\mathcal{P}_1 \to \mathcal{P}_2) \in \mathrm{Ob}(\mathbf{POSet})$ the poset of (possibly non-monotonic) functions from $\mathcal{P}_1$ to $\mathcal{P}_2$ ordered with the pointwise order.

Definition 4.13

An *approximation system (for $S_{\mathbb{K}}$)* is a tuple $(\mathbf{Approx}, \mathrm{App}, \{\mathcal{E}_\tau\}_{\tau\in\mathcal{B}}, \{\mathfrak{p}^0_\tau\}_{\tau\in\mathcal{B}})$ where

1. **Approx** is a Cartesian closed full subcategory of **CPO**, called an *approximation category*. The objects of **Approx** are called *approximation spaces*.
2. $\mathrm{App} : S_{\mathbb{K}} \to \mathrm{Ob}(\mathbf{Approx})$ is a function, called an *approximating function*, such that for all $E_\tau \in S_{\mathbb{K}}$
   (a) if $\tau = \Pi_{i\in I}\tau_i$ is a product type, then $\mathrm{App}(E_\tau) = \Pi_{i\in I}\,\mathrm{App}(E_i)$,
   (b) if $\tau = \tau_1 \to \tau_2$ and $E_{\tau_1} \notin S_{\mathbb{K}}$, then $\mathrm{App}(E_{\tau_1\to\tau_2}) = (E_{\tau_1} \to \mathrm{App}(E_{\tau_2}))$,
   (c) if $\tau = \tau_1 \to \tau_2$ and $E_{\tau_1} \in S_{\mathbb{K}}$, then $\mathrm{App}(E_{\tau_1\to\tau_2}) = \mathrm{App}(E_{\tau_2})^{\mathrm{App}(E_{\tau_1})}$.
3. $\{\mathcal{E}_\tau\}_{\tau\in\mathcal{B}}$ is a family of sets such that the following hold:
   (a) for each base type $\tau \in \mathcal{B}$, $\mathcal{E}_\tau \subseteq \mathrm{App}(E_\tau)$,
   (b) either $\mathrm{App}(E_\tau) \in \mathrm{Ob}(\mathbf{CJSLat})$ for all $\tau \in \mathcal{B}$, or for all $\tau \in \mathcal{B}$, if $a \in \mathcal{E}_\tau$ and $b \in \mathrm{App}(E_\tau)$ such that $a \le b$, then also $b \in \mathcal{E}_\tau$.
4. $\{\mathfrak{p}^0_\tau\}_{\tau\in\mathcal{B}}$ is a family of functions such that for each base type $\tau \in \mathcal{B}$:
   (a) $\mathfrak{p}^0_\tau : \mathcal{E}_\tau \to E_\tau$,
   (b) for all $a, b \in \mathcal{E}_\tau$, if $a \le b$, then $\mathfrak{p}^0_\tau(a) = \mathfrak{p}^0_\tau(b)$,
   (c) for all $e \in E_\tau$, there exists $\mathrm{glb}((\mathfrak{p}^0_\tau)^{-1}(e)) \in \mathcal{E}_\tau$ and $\mathfrak{p}^0_\tau(\mathrm{glb}((\mathfrak{p}^0_\tau)^{-1}(e))) = e$, i.e., the following diagram commutes

Note that, by Proposition 4.3, the object $(E_{\tau_1} \to \mathrm{App}(E_{\tau_2}))$ in Item 2b of Definition 4.13 is indeed an object of the approximation category **Approx**. Morover, again by Proposition 4.3, it holds that $E_{\tau_1\to\tau_2} = (E_{\tau_1} \to E_{\tau_2}) \cong \Pi_{i\in E_{\tau_1}} E_{\tau_2} = E_{\Pi_{i\in E_{\tau_1}}\tau_2}$. However, in Item 2c of the above definition, we have $\mathrm{App}(E_{\tau_1\to\tau_2}) = \mathrm{App}(E_{\tau_2})^{\mathrm{App}(E_{\tau_1})} \ncong \Pi_{i\in E_{\tau_1}}\mathrm{App}(E_{\tau_2}) = \mathrm{App}(E_{\Pi_{i\in E_{\tau_1}}\tau_2})$. Hence, while the map App, in a way, respects the structure given by the type hierarchy $\mathbb{H}$, it does not commute with isomorphisms of posets.

Finally, it is important to notice that, while the approximation system depends on the application at hand, i.e., on the language, the semantics, and so on, the approximation category depends only on the version of AFT.

Proposition 4.4

Let $(\mathbf{Approx}, \mathrm{App}, \{\mathcal{E}_\tau\}_{\tau\in\mathcal{B}}, \{\mathfrak{p}^0_\tau\}_{\tau\in\mathcal{B}})$ be an approximation system. The function $\mathfrak{p}^0_\tau$ is surjective for every $\tau \in \mathcal{B}$.

Proof. Let $\tau \in \mathcal{B}$, and define $\mathfrak{b}^0_\tau := \mathrm{glb} \circ (\mathfrak{p}^0_\tau)^{-1}$. By Item 4c in Definition 4.13, $\mathfrak{b}^0_\tau$ is a well-defined right inverse of $\mathfrak{p}^0_\tau$. □

We now fix an approximation system $(\mathbf{Approx}, \mathrm{App}, \{\mathcal{E}_\tau\}_{\tau\in\mathcal{B}}, \{\mathfrak{p}^0_\tau\}_{\tau\in\mathcal{B}})$ for $S_\mathbb{K}$, and extend the notion of exactness to all approximation spaces.

Definition 4.14

Let $E_\tau \in S_\mathbb{K}$. An element $e \in \mathrm{App}(E_\tau)$ is *exact* if one of the following conditions holds:

1. $\tau \in \mathcal{B}_\mathbb{K}$ and $e \in \mathcal{E}_\tau$,
2. $\tau = \Pi_{i\in I}\tau_i$ and for each $i \in I$, the i-th component $\pi_i(e)$ of e is exact,
3. $\tau = \tau_1 \to \tau_2$, $E_{\tau_1} \notin S_\mathbb{K}$, and for all $e_1 \in E_{\tau_1}$, $e(e_1) \in \mathrm{App}(E_{\tau_2})$ is exact.
4. $\tau = \tau_1 \to \tau_2$, $E_{\tau_1} \in S_\mathbb{K}$, and for all $e_1 \in \mathrm{App}(E_{\tau_1})$ exact, $e(e_1) \in \mathrm{App}(E_{\tau_2})$ is exact.

For $\tau \in \mathbb{K}$, we denote by $\mathcal{E}_\tau$ the subset of $\mathrm{App}(E_\tau)$ of exact elements of type τ.

The reader may wonder why there are two cases for a morphism type $\tau_1 \to \tau_2$ in Definition 4.14, depending whether E_{τ_1} is in $S_\mathbb{K}$ or not, i.e., whether we approximate the elements of the semantics of τ_1 or not. Recall that an exact element in the approximation space is meant to represent one and only one element of the semantics of the same type. Intuitively, for a morphism type $\tau_1 \to \tau_2$, a function in $\mathrm{App}(E_{\tau_1\to\tau_2})$ is exact when the image of any exact element is exact. This is indeed sufficient and we do not need to consider the image of non-exact elements of the domain: we will prove in Proposition 4.6 that there is an exact element in the approximation space for each element of the semantics of the corresponding type. Now, considering *exact elements of the domain* (and looking at their image) makes sense only when the domain is an approximation space, i.e., when $E_{\tau_1} \in S_\mathbb{K}$; in the other case, when $E_{\tau_1} \notin S_\mathbb{K}$ we can directly consider the elements of the semantics, as they do not get approximated.

The following proposition shows that the condition 3b of Definition 4.13 holds for any $E_\tau \in S_\mathbb{K}$.

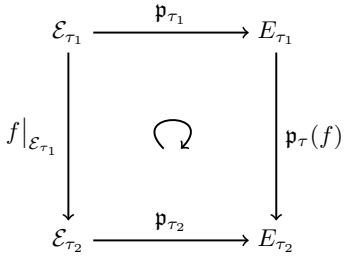
Proposition 4.5

Either for all $E_\tau \in S_{\mathbb{K}}$ it holds that $\mathrm{App}(E_\tau) \in \mathrm{Ob}(\mathbf{CJSLat})$, or for all $E_\tau \in S_{\mathbb{K}}$, for all $b \in \mathrm{App}(E_\tau)$, and for all $e \in \mathcal{E}_\tau$, if $e \leq b$, then $b \in \mathcal{E}_\tau$.

Proof. Suppose there exists $E_{\tau'} \in S_{\mathbb{K}}$ such that $\mathrm{App}(E_{\tau'}) \notin \mathrm{Ob}(\mathbf{CJSLat})$. We have to show that for all $E_\tau \in S_{\mathbb{K}}$, for all $b \in \mathrm{App}(E_\tau)$, and for all $e \in \mathcal{E}_\tau$, if $e \leq b$, then $b \in \mathcal{E}_\tau$. We proceed by induction on τ.
Let τ be a base type. Since we assumed that $\mathrm{App}(E_{\tau'}) \notin \mathrm{Ob}(\mathbf{CJSLat})$ for some $E_{\tau'} \in S_{\mathbb{K}}$, and **CJSLat** is Cartesian closed, and has generalized products by Proposition 4.2, then there must exists a $\sigma \in \mathcal{B}$ such that $\mathrm{App}(E_\sigma) \notin \mathrm{Ob}(\mathbf{CJSLat})$. Thus, by condition 3b of Definition 4.13 we can conclude the base step of the induction.
Now let $\tau = \Pi_{i\in I}\tau_i$ and suppose the proposition hold for E_{τ_i} for all $i \in I$. Let $(b_i) \in \mathrm{App}(E_\tau)$ such that $e := (e_i) \leq (b_i)$. By the definition of the product order, Definition 4.14, and the induction hypothesis, we get that $b_i \in \mathcal{E}_{\tau_i}$ for all $i \in I$, i.e., $(b_i) \in \mathcal{E}_\tau$, as desired.
Let $\tau = \tau_1 \to \tau_2$ with $E_{\tau_1} \notin S_{\mathbb{K}}$, and suppose the proposition hold for E_{τ_2}. By Proposition 4.3, it holds that $\mathrm{App}(E_\tau) \cong \mathrm{App}(\Pi_{i\in E_{\tau_1}} E_{\tau_2})$, thus, we can reduce to the previous case.
Let $\tau = \tau_1 \to \tau_2$ with $E_{\tau_1} \in S_{\mathbb{K}}$, and suppose the proposition hold for E_{τ_1} and E_{τ_2}. Let $f \in \mathrm{App}(E_\tau)$ such that $e \leq f$. For f to be exact, it must send exact elements to exact elements. Let $a \in \mathcal{E}_{\tau_1}$. By the definition of the order on morphisms, and Defintion 4.14, it holds that $f(a) \geq_{\mathrm{App}(E_{\tau_2})} e(a) \in \mathcal{E}_{\tau_2}$. By induction hypothesis, it follows that $f(a) \in \mathcal{E}_{\tau_2}$. Hence, $f \in \mathcal{E}_\tau$, as desired. □

Now that we have defined the exact elements for any semantics in $S_{\mathbb{K}}$, we extend the family $\{\mathfrak{p}^0_\tau\}_{\tau\in\mathcal{B}}$ to have a map for each $E_\tau \in S_{\mathbb{K}}$. We can do this inductively, by defining a new family of functions $\{\mathfrak{p}_\tau : \mathcal{E}_\tau \to E_\tau\}_{E_\tau\in S_{\mathbb{K}}}$ as follows:

1. if $\tau \in \mathcal{B}$, then $\mathfrak{p}_\tau := \mathfrak{p}^0_\tau$,
2. if $\tau = \Pi_{i\in I}\tau_i$, then for all $(e_i)_{i\in I} \in \mathcal{E}_\tau$, $\mathfrak{p}_\tau((e_i)_{i\in I}) := (\mathfrak{p}_{\tau_i}(e_i))_{i\in I}$,
3. if $\tau = \tau_1 \to \tau_2$, and $E_{\tau_1} \notin S_{\mathbb{K}}$, then for all $f \in \mathcal{E}_\tau$, and for all $e \in E_{\tau_1}$, $\mathfrak{p}_\tau(f)(e) := \mathfrak{p}_{\tau_2}(f(e))$.
4. if $\tau = \tau_1 \to \tau_2$, and $E_{\tau_1} \in S_{\mathbb{K}}$, then for all $f \in \mathcal{E}_\tau$, $\mathfrak{p}_\tau(f) : E_{\tau_1} \to E_{\tau_2}$ is the unique function that makes the following diagram commute



In the following proposition, we prove that for each $E_\tau \in S_{\mathbb{K}}$, the function $\mathfrak{p}_\tau$ is well-defined, surjective, and satisfies properties analogous to 4b and 4c of Definition 4.13.

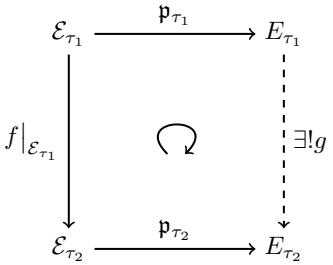
Proposition 4.6

Let $E_\tau \in S_{\mathbb{K}}$, $e_1, e_2 \in \mathcal{E}_\tau$, and $e \in E_\tau$. The following statements hold:

1. $\mathfrak{p}_\tau$ is well-defined.
2. $\mathfrak{p}_\tau$ is surjective.
3. if $e_1 \leq e_2$, then $\mathfrak{p}_\tau(e_1) = \mathfrak{p}_\tau(e_2)$.
4. there exists $\mathrm{glb}(\mathfrak{p}_\tau^{-1}(e)) \in \mathcal{E}_\tau$ and $\mathfrak{p}_\tau(\mathrm{glb}(\mathfrak{p}_\tau^{-1}(e))) = e$.

Proof. We proceed by induction on τ. Let $\tau \in \mathcal{B}_{\mathbb{K}}$. Then $\mathfrak{p}_\tau = \mathfrak{p}_\tau^0$ and the Items 1, 2, 3, and 4 hold by definition of $\mathfrak{p}_\tau^0$.
Now suppose $\tau = \Pi_{i\in I}\tau_i$, and assume that Items 1, 2, 3, and 4 hold for τ_i for all $i \in I$. Since $\mathfrak{p}_{\tau_i}$ is well-defined and surjective by hypothesis for all $i \in I$, it is clear by definition that also $\mathfrak{p}_\tau$ is well-defined and surjective. Let $(a_i), (b_i) \in \mathcal{E}_\tau$ such that $(a_i) \leq (b_i)$. By the product order on $\mathrm{App}(E_\tau)$ we have $a_i \leq b_i$ for all $i \in I$. By definition 4.14, $a_i, b_i \in \mathcal{E}_{\tau_i}$ for all $i \in I$. Hence, by hypothesis it follows that $\mathfrak{p}_{\tau_i}(a_i) = \mathfrak{p}_{\tau_i}(b_i)$ for all $i \in I$. By definition of $\mathfrak{p}_\tau$, we get that $\mathfrak{p}_\tau((a_i)) = \mathfrak{p}_\tau((b_i))$. Thus, Item 3 hold. Now let $(a_i) \in E_\tau$. By the definition of $\mathfrak{p}_\tau$, it is easy to see that $\mathfrak{p}_\tau^{-1}((a_i)) = \Pi_{i\in I}\mathfrak{p}_{\tau_i}^{-1}(a_i)$. Hence, by induction hypothesis for Item 4, we have that $\mathrm{glb}(\mathfrak{p}_\tau^{-1}((a_i))) = \mathrm{glb}(\Pi_{i\in I}\mathfrak{p}_{\tau_i}^{-1}(a_i)) = \Pi_{i\in I}\,\mathrm{glb}(\mathfrak{p}_{\tau_i}^{-1}(a_i)) \in \Pi_{i\in I}\mathfrak{p}_{\tau_i}^{-1}(a_i) = \mathfrak{p}_\tau^{-1}((a_i))$, where the second equality holds because of the definition of the product order on $\mathrm{App}(E_\tau)$. Thus, also Item 4 hold for $\mathfrak{p}_\tau$.
Suppose $\tau = \tau_1 \to \tau_2$ with $E_{\tau_1} \notin S_{\mathbb{K}}$, and assume that Items 1, 2, 3, and 4 hold for and τ_2. By Proposition 4.3 and Definition 4.13, we have $E_{\tau_1\to\tau_2} = (E_{\tau_1} \to E_{\tau_2}) \cong \Pi_{i\in E_{\tau_1}} E_{\tau_2}$ and $\mathrm{App}(E_{\tau_1\to\tau_2}) \cong \mathrm{App}(\Pi_{i\in E_{\tau_1}} E_{\tau_2})$. It is easy to see that we can reduce to the previous case with $I := E_{\tau_1}$.

Finally, suppose $\tau = \tau_1 \to \tau_2$ with $E_{\tau_1} \in S_{\mathbb{K}}$, and assume that Items 1, 2, 3, and 4 hold for τ_1 and τ_2. We first show that $\mathfrak{p}_\tau$ is well-defined, i.e., for all $f \in \mathcal{E}_\tau$ there exists unique $g \in E_\tau = E_{\tau_2}^{E_{\tau_1}}$ making the following diagram commute



Let $f \in \mathcal{E}_\tau$. First notice that, since $\mathfrak{p}_{\tau_1}$ is surjective by hypothesis, for all $e \in E_{\tau_1}$ there exists $d \in \mathcal{E}_{\tau_1}$ such that $\mathfrak{p}_{\tau_1}(d) = e$. Moreover, since $\mathfrak{p}_{\tau_2}$ is well-defined by hypothesis, for all $e \in E_{\tau_1}$ and $d \in \mathfrak{p}_{\tau_1}^{-1}(e)$, we get an element $\mathfrak{p}_{\tau_2}(f(d)) \in E_{\tau_2}$. It remains to show that for all $e \in E_{\tau_1}$, if $d_1, d_2 \in \mathfrak{p}_{\tau_1}^{-1}(e) \subseteq \mathcal{E}_{\tau_1}$, then $\mathfrak{p}_{\tau_2}(f(d_1)) = \mathfrak{p}_{\tau_2}(f(d_2))$. Let $e \in E_{\tau_1}$ and $d_1, d_2 \in \mathfrak{p}_{\tau_1}^{-1}(e)$. By induction hypothesis for Item 4, there exists $d_3 \in \mathfrak{p}_{\tau_1}^{-1}(e)$ such that $d_3 \leq d_1$, $d_3 \leq d_2$. Since $f \in \mathcal{E}_\tau = \mathcal{E}_{\tau_2}^{\mathcal{E}_{\tau_1}}$ is a morphism of cpo's, it is monotonic. Hence, it holds that $f(d_3) \leq f(d_1)$, $f(d_3) \leq f(d_2)$. By induction hypothesis for Item 3, it follows that $\mathfrak{p}_{\tau_2}(f(d_1)) = \mathfrak{p}_{\tau_2}(f(d_2))$. Thus, $\mathfrak{p}_\tau(f) := g$ and $\mathfrak{p}_\tau$ is well-defined.
We now show that $\mathfrak{p}_\tau$ is surjective. Let $g \in E_\tau = E_{\tau_2}^{E_{\tau_1}}$. By the induction hypothesis on τ_2 for Item 4, for each $e \in E_{\tau_1}$, we can define an element $d_e := \mathrm{glb}(\mathfrak{p}_{\tau_2}^{-1}(g(e))) \in \mathfrak{p}_{\tau_2}^{-1}(g(e))$. By Proposition 4.5, for each $a \in App(E_{\tau_1}) \setminus \mathcal{E}_{\tau_1}$ such that there exists (at least one) $b \in \mathcal{E}_{\tau_1}$ with $b \leq a$, we can define

$$c_a := \mathrm{lub}(\{d_{\mathfrak{p}_{\tau_1}(b)} \mid b \in \mathcal{E}_{\tau_1} \text{ such that } b \leq a\}).$$

We define $f : \ \mathrm{App}(E_{\tau_1}) \to \mathrm{App}(E_{\tau_2})$ for all $a \in \mathrm{App}(E_{\tau_1})$ as follows:

$$f(a) := \begin{cases} d_{\mathfrak{p}_{\tau_1}(a)} & \text{if } a \in \mathcal{E}_{\tau_1}, \\ c_a & \text{if } a \notin \mathcal{E}_{\tau_1} \text{ and exists } b \in \mathcal{E}_{\tau_1} \text{ such that } b \leq a, \\ \bot_{\mathrm{App}(E_{\tau_2})} & \text{otherwise.} \end{cases} \tag{4.1}$$

In the following we show that f is monotonic. Let $a_1, a_2 \in \mathrm{App}(E_{\tau_1})$ such that $a_1 \leq a_2$. If $a_1 \notin \mathcal{E}_{\tau_1}$ and for all $b \in \mathcal{E}_{\tau_1}$ is not the case that $b \leq a$, then clearly $f(a_1) = \bot_{\mathrm{App}(E_{\tau_2})} \leq f(a_2)$. If $a_1, a_2 \in \mathcal{E}_{\tau_1}$, then $f(a_1) = f(a_2)$ by the induction hypothesis for Item 3. If a_1 is exact but a_2 is not, then clearly $f(a_1) \leq f(a_2)$. If both $a_1, a_2 \notin \mathcal{E}_{\tau_1}$ and they are greater than some exact $b \in \mathcal{E}_{\tau_1}$, then $\{b \in \mathcal{E}_{\tau_1} \mid b \leq a_1\} \subseteq \{b \in \mathcal{E}_{\tau_1} \mid b \leq a_2\}$.

Hence, $f(a_1) = c_{a_1} \leq c_{a_2} = f(a_2)$, as desired. It follows that $f \in \text{App}(E_\tau)$. Moreover, it is clear that f sends exact elements to exact elements, i.e., $f \in \mathcal{E}_\tau$. For all $e \in E_{\tau_1}$, $\mathfrak{p}_\tau(f)(e) = \mathfrak{p}_{\tau_2}(f(c)) = \mathfrak{p}_{\tau_2}(d_{\mathfrak{p}_{\tau_1}(c)}) = \mathfrak{p}_{\tau_2}(d_e) = g(e)$, where c is some element in the preimage $\mathfrak{p}_{\tau_1}^{-1}(e)$. Thus, $\mathfrak{p}_\tau(f) = g$, as desired.
We proceed to show that Item 3 holds for τ. Let $f_1, f_2 \in \mathcal{E}_\tau$ such that $f_1 \leq f_2$, and let $e \in E_{\tau_1}$. We have already shown in Item 1 for τ that $\mathfrak{p}_\tau$ is well defined. In particular, $\mathfrak{p}_\tau(g)(e) = \mathfrak{p}_{\tau_2}(g(d_1)) = \mathfrak{p}_{\tau_2}(g(d_2))$ for all $g \in \mathcal{E}_\tau$ and $d_1, d_2 \in \mathfrak{p}_{\tau_1}^{-1}(e)$. By the definition of the order on morphisms, $f_1(d) \leq f_2(d)$ for all $d \in \mathcal{E}_{\tau_1}$. By the induction hypothesis for Item 3, it holds that $\mathfrak{p}_\tau(f_1)(e) = \mathfrak{p}_{\tau_2}(f_1(d)) = \mathfrak{p}_{\tau_2}(f_2(d)) = \mathfrak{p}_\tau(f_2)(e)$. Hence, $\mathfrak{p}_\tau(f_1) = \mathfrak{p}_\tau(f_2)$, as desired.
Finally, we show that Item 4 holds for τ. Let $e \in E_\tau$. We can construct a morphism $f \in \mathcal{E}_\tau$ using the same technique as in (4.1). By the proof of Item 2, we already have $\mathfrak{p}_\tau(f) = e$. It remains to show that $f = \text{glb}(\mathfrak{p}_\tau^{-1}(e))$. Let $h \in \mathfrak{p}_\tau^{-1}(e)$. First notice that since $\mathfrak{p}_\tau(h) = \mathfrak{p}_\tau(f) = e$, it holds that for all $e' \in E_{\tau_1}$, $\mathfrak{p}_{\tau_2}(f(l)) = \mathfrak{p}_{\tau_2}(h(l)) = e(e')$ for all $l \in \mathfrak{p}_{\tau_1}^{-1}(e')$. In particular, for all $a \in \mathcal{E}_{\tau_1}$, it holds that $f(a), h(a) \in \mathfrak{p}_{\tau_2}^{-1}(e(\mathfrak{p}_{\tau_1}(a)))$. Hence, for all $a \in \mathcal{E}_{\tau_1}$, we have that $f(a) = d_{\mathfrak{p}_{\tau_1}(a)} = \text{glb}(\mathfrak{p}_{\tau_2}^{-1}(e(\mathfrak{p}_{\tau_1}(a)))) \leq h(a)$. Now let $a \notin \mathcal{E}_{\tau_1}$ such that there exists $b \in \mathcal{E}_{\tau_1}$ such that $b \leq a$. Since we have already shown that $f(c) = d_{\mathfrak{p}_{\tau_1}(c)} \leq h(c)$ for all $c \in \mathcal{E}_{\tau_1}$, it is easy to see that $f(a) = c_a \leq h(a)$. For all the other cases of $a \in \text{App}(E_{\tau_1})$ it is obvious that $f(a) \leq h(a)$. Hence, f is a lower bound of $\mathfrak{p}_\tau^{-1}(e)$. Since $f \in \mathfrak{p}_\tau^{-1}(e)$, we get $f = \text{glb}(\mathfrak{p}_\tau^{-1}(e))$, as desired. □

Recall that, intuitively, the function $\mathfrak{p}_\tau$ sends an exact element of type τ to the element it represents in the semantics of τ; hence, we call $\mathfrak{p}_\tau$ the *representation function* for τ.

In most applications of AFT, for approximation spaces of base types, there exists a unique exact element representing an object of a semantics, and Items 3 and 4 of Proposition 4.6 are trivially verified. However, for higher-order approximation spaces, this is not always the case, as we illustrate in the following example.

Example 4.2

Let o be the Boolean type, with semantics $E_o := \langle \{\mathbf{f}, \mathbf{t}\}, \leq_o \rangle$, where $\leq_o$ is the standard truth order. In standard AFT, we would define the approximation space for E_o to be the complete lattice $\text{App}(E_o) := \langle E_o \times E_o, \leq_p \rangle$, with $\leq_p$ the precision order induced by $\leq_o$. Then, the semantics for $o \to o$ is the poset of functions from E_o to E_o, and the approximation space for it is the exponential, i.e., the set of monotonic functions from $\text{App}(E_o)$ to itself, ordered pointwise. Clearly, we can set the exact elements of $\text{App}(E_o)$ to be $(\mathbf{f}, \mathbf{f})$ and $(\mathbf{t}, \mathbf{t})$, and $\mathfrak{p}_o$ to send them to $\mathbf{f}$ and $\mathbf{t}$, respectively. Now consider the following two functions: $f, g : \text{App}(E_o) \to \text{App}(E_o)$ defined by $f(\mathbf{f}, \mathbf{t}) = (\mathbf{f}, \mathbf{t})$, $g(\mathbf{f}, \mathbf{t}) = f(\mathbf{f}, \mathbf{f}) = g(\mathbf{f}, \mathbf{f}) = f(\mathbf{t}, \mathbf{t}) = g(\mathbf{t}, \mathbf{t}) = (\mathbf{t}, \mathbf{t})$, and $f(\mathbf{t}, \mathbf{f}) = g(\mathbf{t}, \mathbf{f}) = (\mathbf{t}, \mathbf{f})$. Clearly, both f and g send exacts to exacts, thus, they are exact.

Moreover, even though $f \neq g$, it is easy to see that $\mathfrak{p}_{o\to o}(f) = \mathfrak{p}_{o\to o}(g) = h : E_o \to E_o$, where $h(\mathbf{f}) = h(\mathbf{t}) = \mathbf{t}$.

We conclude this section with the definition of *consistent* elements.

Definition 4.15

Let $E_\tau \in S_{\mathbb{K}}$. An element $c \in \mathrm{App}(E_\tau)$ is *consistent* if there exists $e \in \mathcal{E}_\tau$ such that $c \leq e$.

Notice that a function of the family $\{\mathfrak{p}_\tau : \mathcal{E}_\tau \to E_\tau\}_{E_\tau \in S_{\mathbb{K}}}$ not only determines which element of the semantics an exact element represents, but it also helps understanding what a consistent element is approximating: if $c \in \mathrm{App}(E_\tau)$ is consistent and $c \leq e$ for some exact e, then c approximates $\mathfrak{p}_\tau(e)$. Clearly, consistent elements may approximate more than one element of a semantics.

4 Higher-Order Standard AFT

In this section, we show how our new framework extends the standard AFT setting to higher-order definitions.

The main building block of an approximation system is the category **Approx**, containing all the desired approximation spaces. Hence, we start by showing that the approximation spaces used in standard AFT form a Cartesian closed category.

First, recall that in standard AFT the approximation space of a complete lattice $\mathcal{L} = \langle L, \leq \rangle$ is the complete lattice $\mathcal{A}_L^{\mathsf{s}} = \langle L \times L, \leq_p \rangle$, where $\leq_p$ is the *precision order*, i.e., $(x_1, y_1) \leq_p (x_2, y_2)$ iff $x_1 \leq x_2$ and $y_2 \leq y_1$. If we view these objects from a category-theoretic perspective, we can consider $\mathcal{A}_L^{\mathsf{s}}$ as the (categorical) product between $\mathcal{L}$ and $\mathcal{L}^{op}$ in **CLat**. Hence, we can define the category **SqLat** as follows:

$$\begin{aligned} \mathrm{Ob}(\mathbf{SqLat}) &:= \left\{ \mathcal{L} \times \mathcal{L}^{op} \mid \mathcal{L} \in \mathbf{CLat} \right\} \\ \mathrm{Mor}(\mathbf{SqLat}) &:= \{ f : \mathcal{L}_1 \to \mathcal{L}_2 \mid \mathcal{L}_1, \mathcal{L}_2 \in \mathrm{Ob}(\mathbf{SqLat}) \wedge f \text{ monotonic} \} \end{aligned}$$

Note that, given $\mathcal{L} \times \mathcal{L}^{op} \in \mathrm{Ob}(\mathbf{SqLat})$, it suffices to reverse the order on the second component to retrieve the truth order $\leq_{tr}$ on $L \times L$ (see Definition 2.9). We will denote an element $\mathcal{L} \times \mathcal{L}^{op}$ of **SqLat** by $\overline{\mathcal{L}}$.

Lemma 4.1

The category **SqLat** is a full subcategory of **CLat**.

Proof. Clearly, if $\mathcal{L} \in \mathbf{CLat}$, then $\mathcal{L}^{op} \in \mathbf{CLat}$. Since **CLat** is Cartesian closed, for all $\mathcal{L} \in \mathbf{CLat}$, we have that $\mathcal{L} \times \mathcal{L}^{op} \in \mathbf{CLat}$. We conclude by the definition of **SqLat**. □

By Lemma 4.1, proving that **SqLat** is Cartesian closed reduces to showing that the following isomorphisms of complete lattices hold for all $\overline{\mathcal{L}_1}, \overline{\mathcal{L}_2} \in \mathbf{SqLat}$:

1. $\mathcal{T} \cong \overline{\mathcal{T}}$, where $\mathcal{T}$ is the terminal object of **CLat**,
2. $\overline{\mathcal{L}_1} \times \overline{\mathcal{L}_2} \cong \overline{(\mathcal{L}_1 \times \mathcal{L}_2)}$,
3. $\overline{\mathcal{L}_2}^{\overline{\mathcal{L}_1}} \cong \overline{\mathcal{L}_2^{\overline{\mathcal{L}_1}}}$.

While the first two isomorphisms are rather straightforward, the latter deserves some attention. Consider a morphism of square lattices f from $\overline{\mathcal{L}_1}$ to $\overline{\mathcal{L}_2}$. Since $\overline{\mathcal{L}_2} = \mathcal{L}_2 \times \mathcal{L}_2{}^{op}$, we can write f as a pair (f_1, f_2) of morphisms of complete lattices, where $f_1 : \overline{\mathcal{L}_1} \to \mathcal{L}_2$ and $f_2 : \overline{\mathcal{L}_1} \to \mathcal{L}_2{}^{op}$. It follows easily that $\overline{\mathcal{L}_2}^{\overline{\mathcal{L}_1}} \cong \mathcal{L}_2^{\overline{\mathcal{L}_1}} \times (\mathcal{L}_2{}^{op})^{\overline{\mathcal{L}_1}}$. Then, the isomoprhism $\varphi : \overline{\mathcal{L}_2}^{\overline{\mathcal{L}_1}} \to \overline{\mathcal{L}_2^{\overline{\mathcal{L}_1}}}$, is realised by mapping $f \hat{=} (f_1, f_2)$ to a new pair $\varphi(f) := (f_1, f_2') \in \mathcal{L}_2^{\overline{\mathcal{L}_1}} \times (\mathcal{L}_2^{\overline{\mathcal{L}_1}})^{op} = \mathcal{L}_2^{\overline{\mathcal{L}_1}}$, where the second component is defined by $f_2'(x, y) := f_2(y, x)$. Notice that, since f_2 is a monotonic function from $\overline{\mathcal{L}_1}$ to $\mathcal{L}_2{}^{op}$, f_2' is indeed a monotonic function from $\overline{\mathcal{L}_1}$ to $\mathcal{L}_2$. Further details regarding the isomorphisms listed above are contained in the proof of Theorem 4.1.

Theorem 4.1

The category **SqLat** is Cartesian closed.

Proof. Since **CLat** is Cartesian closed, and **SqLat** is a full-subcategory of **CLat** (Lemma 4.1), it is sufficient to show that the terminal object of **CLat** is an object of **SqLat** and that for all $\overline{\mathcal{L}_1}, \overline{\mathcal{L}_2} \in \mathbf{SqLat}$, the product $\overline{\mathcal{L}_1} \times \overline{\mathcal{L}_2}$ and the exponential $\overline{\mathcal{L}_2}^{\overline{\mathcal{L}_1}}$, computed in the category **CLat**, are also objects of **SqLat**.

- *Terminal object.* There is an obvious isomorphism from the terminal object $\mathcal{T}$ of **CLat**, i.e., the lattice with just one element and trivial order, and the object $\mathcal{T} \times \mathcal{T}^{op} \in \mathbf{SqLat}$.
- *Product.* Let $\overline{\mathcal{L}_1}, \overline{\mathcal{L}_2} \in \mathbf{SqLat}$. By Cartesian closedness of **CLat**, $\overline{\mathcal{L}_1} \times \overline{\mathcal{L}_2}$ is an object of **SqLat**. We define a function $\varphi : \overline{\mathcal{L}_1} \times \overline{\mathcal{L}_2} \to \overline{\mathcal{L}_1 \times \mathcal{L}_2}$ by sending an element $((a_1, b_1), (a_2, b_2))$ to $((a_1, a_2), (b_1, b_2))$. Clearly, φ is bijective. Moreover, by the definition of the product order, the following double-implications hold

for all $a_1, b_1, x_1, y_1 \in \mathcal{L}_1$, and for all $b_1, b_2, x_2, y_2 \in \mathcal{L}_2$

$$\begin{aligned}((a_1,b_1),(a_2,b_2)) \leq ((x_1,y_1),(x_2,y_2)) \\ \iff (a_1,b_1) \leq (x_1,y_1) \wedge (a_2,b_2) \leq (x_2,y_2) \\ \iff a_1 \leq x_1 \wedge y_1 \leq b_1 \wedge a_2 \leq x_2 \wedge y_2 \leq b_2 \\ \iff (a_1,a_2) \leq (x_1,x_2) \wedge (y_1,y_2) \leq (b_1,b_2) \\ \iff ((a_1,a_2),(b_1,b_2)) \leq ((x_1,x_2),(y_1,y_2)).\end{aligned}$$

Hence, φ and its inverse are monotonic functions, i.e., morphisms. It follows that $\overline{\mathcal{L}_1} \times \overline{\mathcal{L}_2} \cong \overline{\mathcal{L}_1 \times \mathcal{L}_2} \in \mathbf{SqLat}$, as desired.

- *Exponential.* Let $\overline{\mathcal{L}_1}, \overline{\mathcal{L}_2} \in \mathbf{SqLat}$. By Cartesian closedness of **CLat**, $\overline{\mathcal{L}_2}^{\overline{\mathcal{L}_1}}$ is an object of **SqLat**. Let $\delta : \overline{\mathcal{L}_1} \to \overline{\mathcal{L}_1}$ be the function sending (x, y) to (y, x). We define a function $\psi : \overline{\mathcal{L}_2}^{\overline{\mathcal{L}_1}} \to \overline{\mathcal{L}_1^{\mathcal{L}_2}}$ by sending a morphism $f := (f_1, f_2)$ to $(f_1, f_2 \circ \delta)$, where $f_1 : \overline{\mathcal{L}_1} \to \mathcal{L}_2$ and $f_2 : \overline{\mathcal{L}_1} \to {\mathcal{L}_2}^{op}$ are the components of f. Since f_2 is an antimonotonic function from $\overline{\mathcal{L}_1}$ to $\mathcal{L}_2$, it is easy to check that $f_2 \circ \delta$ is a monotonic function from $\overline{\mathcal{L}_1}$ to $\mathcal{L}_2$, as desired. Clearly, φ is bijective. Moreover, by the definition of the pointwise order, the following double-implications hold for all $f_1, g_1 \in \mathcal{L}_2^{\mathcal{L}_1}$, and for all $f_2, g_2 \in ({\mathcal{L}_2}^{op})^{\mathcal{L}_1}$

$$\begin{aligned}(f_1,f_2) \leq (g_1,g_2) \\ \iff \forall (x,y) \in \overline{\mathcal{L}_1}, (f_1(x,y), f_2(x,y)) \leq (g_1(x,y), g_2(x,y)) \\ \iff \forall (x,y) \in \overline{\mathcal{L}_1}, f_1(x,y) \leq g_1(x,y) \wedge g_2(x,y) \leq f_2(x,y) \\ \iff \forall (x,y) \in \overline{\mathcal{L}_1}, f_1(x,y) \leq g_1(x,y) \wedge g_2(y,x) \leq f_2(y,x) \\ \iff f_1 \leq g_1 \wedge g_2 \circ \delta \leq f_2 \circ \delta \\ \iff (f_1, f_2 \circ \delta) \leq (g_1, g_2 \circ \delta).\end{aligned}$$

Hence, ψ and its inverse are monotonic functions, i.e., morphisms. It follows that $\overline{\mathcal{L}_2}^{\overline{\mathcal{L}_1}} \cong \overline{\mathcal{L}_1^{\mathcal{L}_2}} \in \mathbf{SqLat}$, as desired.

□

It is interesting to observe that the approximators used in standard AFT, i.e., the *symmetric* approximators, when viewed in their square lattice approximator space, correspond to pairs of equal functions, i.e., the classic definition of exact pair (Definition 2.11 or [Denecker et al., 2000]). Similarly, a *gracefully degrading* approximator $A = (A_1, A_2) : \overline{\mathcal{L}} \to \overline{\mathcal{L}}$, i.e., such that $A_1(x, y) \leq A_2(y, x)$ for all $(x, y) \in \overline{\mathcal{L}}$ [Denecker and Vennekens, 2007], when viewed in $\overline{\mathcal{L}^{\overline{\mathcal{L}}}}$ is a pair $\varphi(A) = (A_1, A_2')$ with $A_1 \leq A_2'$, i.e., a consistent pair according to the classic definition of AFT.

Thanks to Theorem 4.1 and **SqLat** $\subseteq$ **CPO**, we can fix **SqLat** as our approximation category. This can be done for *any* application in which we want to use standard AFT techniques. Nevertheless, depending on the application at hand, the approximation system may differ. Let us show how to define an approximation system given a language based on a type hierarchy $\mathbb{H}$. Let $S_{\mathbb{K}}$ be the set of the semantics of types of $\mathbb{H}$ we want to approximate, and assume that such semantics are complete lattices, as is usually the case in logic programming. Then, we can inductively define a mapping App : $S_{\mathbb{K}} \to \mathrm{Ob}(\mathbf{SqLat})$ by setting, for all $\tau \in \mathcal{B}_{\mathbb{K}}$, $\mathrm{App}(E_\tau) := \overline{E_\tau}$, and proceed using the conditions in Definition 4.13. Notice that the base case of the induction is nothing more than what is usually done in standard AFT: from a complete lattice $\langle L, \leq \rangle$ we obtain the square lattice $\langle L^2, \leq_p \rangle$. The remaining steps are naturally provided by following the Cartesian closed structure of **SqLat**.

For each base type $\tau \in \mathcal{B}_{\mathbb{K}}$, the exact elements of $\mathrm{App}(E_\tau)$ are defined as in standard AFT: $(x, y) \in \mathrm{App}(E_\tau)$ is exact if $x = y$, i.e., $\mathcal{E}_\tau = \{(x, x) \mid x \in E_\tau\}$. Notice that, since **SqLat** $\subseteq$ **CLat** $\subseteq$ **CJSLat**, the condition 3b in Defintion 4.13 is satisfied. Finally, for each base type τ, we define $\mathfrak{p}^0_\tau : \mathcal{E}_\tau \to E_\tau$ by sending (x, x) to x. Both conditions 4b and 4c in Definition 4.13 hold since $(\mathfrak{p}^0_\tau)^{-1}(x) = \{(x, x)\}$. Hence, we have obtained an approximation system $(\mathbf{SqLat}, \mathrm{App}, \{\mathcal{E}_\tau\}_{\tau \in \mathcal{B}}, \{\mathfrak{p}^0_\tau\}_{\tau \in \mathcal{B}})$ for $S_{\mathbb{K}}$.

In standard AFT, we are ultimately interested in the approximation space of interpretations. Given a vocabulary V, $S_{\mathbb{K}}$ can be easily chosen to contain the semantics of the types of the symbols in V and the space of interpretations for V, i.e., the complete lattice $\Pi_{s \in V'} E_{\tau(s)}$, where $\tau(s)$ is the type of the symbol s. It follows that the approximation space of interpretations is $\mathrm{App}(\Pi_{s \in V'} E_{\tau(s)}) = \Pi_{s \in V'} \mathrm{App}(E_{\tau(s)}) \in \mathbf{SqLat}$. Clearly, if we restrict to a vocabulary with only symbols of base type, then we retrieve the usual framework of standard AFT.

5 Revised Higher-Order Consistent AFT

Charalambidis et al. [2018] developed an extension of consistent AFT (Section 2.2, Chapter 2) to generalize the well-founded semantics for classical logic programs to one for programs with higher-order predicates. As already pointed out in Section 2.1, this generalization bears some issues.

In this section, we examine in detail the work of Charalambidis et al. [2018] under the lenses of our novel categorical framework. First, in Subsection 5.1, we present their extension of consistent AFT with their version of approximation spaces, and we prove that this new class of mathematical objects forms a Cartesian closed category. Then, in Subsection 5.2, we briefly recall the types and semantics used by Charalambidis et al. [2018], and we define an approximation system for it. Thanks to the inductive nature of

Cartesian closed categories, from the tuple defining the approximation system, we can effortlessly retrieve the entire, complex hierarchy built by Charalambidis et al. [2018]. From the definition of the approximation system, we already obtain a concept of exactness for higher-order objects, which was previously missing in [Charalambidis et al., 2018]. Finally, in Subsection 5.3, we present our solution to the problem encountered in the work of Charalambidis et al. [2018] concerning the semantics of logic programs with existential quantifiers. In particular, we propose a new approximator which provides the expected well-founded semantics. We conclude the subsection with two examples of logic programs in which we need to apply an approximate object on another approximate object.

5.1 The Approximation Category

Recall that, in consistent AFT (Section 2.2, Chapter 2), an approximation space is the cpo formed by the consistent part of a square lattice, i.e., given a square lattice $\langle L \times L, \leq_p \rangle$ of a complete lattice $\langle L, \leq \rangle$, only the subset $\{(x, y) \mid x \leq y\} \subseteq L \times L$ of consistent elements is taken into account. Charalambidis et al. [2018] extended consistent AFT to a new class of approximation spaces: the sets of the form $L \otimes U := \{(x, y) \mid x \in L, y \in U, x \leq y\}$, comprising the consistent elements of the cartesian product between a set L of *lower bounds* and a set U of *upper bounds*, where L may differ from U.

In order for the machinery of consistent AFT to work over these new spaces, Charalambidis et al. [2018] added some conditions to restrain the possible choices for L and U.

Definition 4.16

An *approximation tuple* is a tuple $(L, U \leq)$, where L and U are sets, and $\leq$ is a partial order on $L \cup U$ such that the following conditions hold:

1. $\langle L \cup U, \leq \rangle$ has a top element $\top_{L \cup U}$ and a bottom element $\bot_{L \cup U}$,
2. $\top_{L \cup U}, \bot_{L \cup U} \in L \cap U$,
3. $\langle L, \leq \rangle$ and $\langle U, \leq \rangle$ are complete lattices,
4. *Interlattice Least upper bound Property (ILP)*: for all $b \in U$ and for all $S \subseteq L$ such that for all $x \in S$, $x \leq b$, we have $\mathrm{lub}_L(S) \leq b$,
5. *Interlattice Greatest lower bound Property (IGP)*: for all $a \in L$ and for all $S \subseteq U$ such that for all $x \in S, a \leq x$, we have $a \leq \mathrm{glb}_U(S)$.

Definition 4.17

Let $(L, U, \leq)$ be an approximation tuple. The *approximation space* (associated to $(L, U, \leq)$) is the poset $\langle L \otimes U, \leq_p \rangle$, where $L \otimes U := \{(x, y) \mid x \in L, y \in U, x \leq y\}$, and $\leq_p$ is the binary relation on $L \otimes U$ defined for all $(x_1, y_1), (x_2, y_2) \in L \otimes U$ by: $(x_1, y_1) \leq_p (x_2, y_2)$ iff $x_1 \leq x_2$ and $y_2 \leq y_1$. We call $\leq_p$ the *precision order* on $L \otimes U$.

It is easy to show that the binary relation $\leq_p$ is indeed a partial order on $L \otimes U$. In the remainder of this subsection, we prove that the new class of approximation spaces defined in Definition 4.17 forms a Cartesian closed full subcategory of **CPO** (Theorem 4.2). First, we define a new category **LUcons**, with objects the approximation spaces just introduced, as follows:

$$\begin{aligned} \mathrm{Ob}(\mathbf{LUcons}) &:= \{\langle L \otimes U, \leq_p \rangle \mid (L, U, \leq) \text{ is an approximation tuple}\} \\ \mathrm{Mor}(\mathbf{LUcons}) &:= \{f : A \to B \mid A, B \in \mathrm{Ob}(\mathbf{LUcons}) \wedge f \text{ monotonic}\}. \end{aligned}$$

Theorem 4.2

The category **LUcons** is a Cartesian closed full subcategory of **CPO**.

We split the proof of Theorem 4.2 into smaller results: first we show that **LUcons** is a full subcategory of **CPO**, then we prove it is Cartesian closed.

Proposition 4.7

Let $L \otimes U \in \mathrm{Ob}(\mathbf{LUcons})$. Then $L \otimes U$ is a cpo.

Proof. Let $L \otimes U \in \mathrm{Ob}(\mathbf{LUcons})$, and $S \subseteq L \otimes U$ a chain. We denote by $p_1 : L \otimes U \to L$ the function of sets sending (x, y) to x, and by $p_2 : L \otimes U \to U$ the function sending (x, y) to y. Clearly, $p_1(S)$ and $p_2(S)$ are chains in $\langle L, \leq \rangle$ and $\langle U, \leq \rangle$, respectively. Since $\langle L, \leq \rangle$ and $\langle U, \leq \rangle$ are complete lattices, there exist $\mathrm{lub}_L(p_1(S)) =: x \in L$ and $\mathrm{glb}_U(p_2(S)) =: y \in U$. We now show that $(x, y) \in L \otimes U$, i.e., $x \leq y$. Let $a \in p_1(S) \subseteq L$ and let $a' \in p_2(S)$ such that $(a, a') \in S$. For each $x' \in p_2(S)$, let $x \in p_1(S)$ such that $(x, x') \in S$. Since S is a chain, we either have $(a, a') \leq_p (x, x')$ or $(x, x') \leq_p (a, a')$. In both cases, $a \leq x'$. By IGP for a and $p_2(S)$, we have that $a \leq y$. Since this hold for any $a \in p_1(S) \subseteq L$, we can conclude that $x \leq y$ by ILP for y and $p_1(S)$, as desired. Hence, $(x, y) \in L \otimes U$. Then, it is easy to see that $(x, y) = \mathrm{lub}_{L \otimes U}(S) \in L \otimes U$. □

Corollary 4.1

LUcons is a full subcategory of **CPO**.

Proof. Clear from Proposition 4.7 and the definition of Mor(**LUcons**). □

Proposition 4.8

LUcons is a Cartesian closed category.

Proof. Since **CPO** is Cartesian closed, and **LUcons** is a full-subcategory of **CPO** (Corollary 4.1), it is sufficient to show that the terminal object of **CPO** is an object of **LUcons** and that for all $\mathcal{A}, \mathcal{B} \in$ **LUcons**, the product $\mathcal{A} \times \mathcal{B}$ and the exponential $\mathcal{B}^{\mathcal{A}}$, computed in the category **CPO**, are also objects of **LUcons**.
Let $\mathcal{A}, \mathcal{B} \in$ **LUcons**. Let $(L_{\mathcal{A}}, U_{\mathcal{A}}, \leq_{\mathcal{A}})$ and $(L_{\mathcal{B}}, U_{\mathcal{B}}, \leq_{\mathcal{B}})$ be the approximation tuples of the approximation spaces $\mathcal{A}, \mathcal{B} \in$ **LUcons**. We denote the orders on $\mathcal{A}$ and $\mathcal{B}$ as $\leq_{p,\mathcal{A}}$ and $\leq_{p,\mathcal{B}}$, respectively.

- *Terminal object.* Let $\mathcal{T} = \langle \{*\}, \leq \rangle$ be the cpo with one element, i.e., the terminal object of **CPO**. Clearly, $(\{*\}, \{*\}, \leq)$ is an approximation tuple, and $\mathcal{T} \cong \langle \{*\} \otimes \{*\}, \leq \rangle$ in **CPO**. Hence, $\mathcal{T} \in \mathrm{Ob}($**LUcons**$)$.
- *Product.* This follows from Proposition 4.2 and Corollary 4.1.
- *Exponential.* We have to show that $\mathcal{B}^{\mathcal{A}} \in$ **CPO** is isomorphic (in **CPO**) to some $\mathcal{C} = L_{\mathcal{C}} \otimes U_{\mathcal{C}} \in$ **LUcons** associated to an approximating tuple $(L_{\mathcal{C}}, U_{\mathcal{C}}, \leq_{\mathcal{C}})$. Let $L_{\mathcal{C}} := \hom_{\mathbf{CPO}}(\mathcal{A}, \langle L_{\mathcal{B}}, \leq_{\mathcal{B}} \rangle)$, $U_{\mathcal{C}} := \hom_{\mathbf{CPO}}(\mathcal{A}, \langle U_{\mathcal{B}}, \geq_{\mathcal{B}} \rangle)$, and $\leq_{\mathcal{C}}$ be the restriction onto $L_{\mathcal{C}} \cup U_{\mathcal{C}}$ of the pointwise extension of $\leq_{\mathcal{B}}$, namely for all $f, g \in L_{\mathcal{C}} \cup U_{\mathcal{C}}$,
$$f \leq_{\mathcal{C}} g \iff \forall x \in \mathcal{A}, f(x) \leq_{\mathcal{B}} g(x)$$
We first show that $L_{\mathcal{C}} \otimes U_{\mathcal{C}}$ with the precision order $\leq_{p,\mathcal{C}}$ induced by $\leq_{\mathcal{C}}$ is an object of **LUcons**. In other words, we show that $(L_{\mathcal{C}}, U_{\mathcal{C}}, \leq_{\mathcal{C}})$ is an approximation tuple.
  1. The morphisms $\bot_{\mathcal{C}} : x \mapsto \bot_{\mathcal{B}}$ and $\top_{\mathcal{C}} : x \mapsto \top_{\mathcal{B}}$ are the bottom and top element of $L_{\mathcal{C}} \cup U_{\mathcal{C}}$, respectively.
  2. Clearly, $\bot_{\mathcal{C}}, \top_{\mathcal{C}} \in L_{\mathcal{C}} \cap U_{\mathcal{C}}$.
  3. Since $\langle L_{\mathcal{B}}, \leq_{\mathcal{B}} \rangle$ and $\langle U_{\mathcal{B}}, \leq_{\mathcal{B}} \rangle$ are complete lattices by definition of approximation space, and $\leq_{\mathcal{C}}$ is the pointwise extension of $\leq_{\mathcal{B}}$, it is straightforward to see that $\langle L_{\mathcal{C}}, \leq_{\mathcal{C}} \rangle$ and $\langle U_{\mathcal{C}}, \leq_{\mathcal{C}} \rangle$ are also complete lattices.
  4. Let $g \in U_{\mathcal{C}}$, and let $S \subseteq L_{\mathcal{C}}$ such that for all $f \in S$, $f \leq_{\mathcal{C}} g$, i.e., for all $x \in \mathcal{A}$ we have $f(x) \leq_{\mathcal{B}} g(x)$. We have to show that $\mathrm{lub}_{L_{\mathcal{C}}}(S) \leq_{\mathcal{C}} g$. Since $g \in U_{\mathcal{C}}$ and $f \in L_{\mathcal{C}}$, for all $x \in \mathcal{A}$ we have $g(x) \in U_{\mathcal{B}}$ and $S_x := \{f(x) \mid f \in S\} \subseteq L_{\mathcal{B}}$.

By using the ILP on $\mathcal{B}$, we get that $\text{lub}_{L_\mathcal{B}}(S_x) \leq_\mathcal{B} g(x)$, for all $x \in \mathcal{A}$. It is not difficult to see that $\text{lub}_{L_\mathcal{C}}(S(x)) = \text{lub}_{L_\mathcal{B}}(S_x)$, for all $x \in \mathcal{A}$. Hence, $\text{lub}_{L_\mathcal{C}}(S) \leq_\mathcal{C} g$, as desired.

5. Analogous to Item 4.

Hence, $\mathcal{C} \in \textbf{LUcons}$. It remains to show that $\mathcal{C}$ is isomorphic to $\mathcal{B}^\mathcal{A}$ in **CPO**. Notice that there is an obvious isomorphism of sets

$$\begin{aligned} \mu : \ \text{hom}_{\textbf{CPO}}(\mathcal{A}, L_\mathcal{B} \times U_\mathcal{B}) &\to L_\mathcal{C} \times U_\mathcal{C} \\ f &\mapsto (f_1, f_2), \end{aligned}$$

where f_1, f_2 are the two components of f. By the definitions of the orders (notice the inversion of the order $\leq_\mathcal{B}$ on $U_\mathcal{B}$ in $U_\mathcal{C}$), it is easy to check that μ and μ^{-1} are both well-defined, i.e., they send a monotonic function to a pair of monotonic functions, and a pair of monotonic functions to a monotonic function, respectively. Now, let $f \in \text{hom}_{\textbf{CPO}}(\mathcal{A}, L_\mathcal{B} \times U_\mathcal{B})$. Then

$$\begin{aligned} f \in \text{hom}_{\textbf{CPO}}(\mathcal{A}, \mathcal{B}) &\iff \forall x \in \mathcal{A}, f(x) \in \mathcal{B} \\ &\iff \forall x \in \mathcal{A}, \mu_1(f)(x) = f_1(x) \leq_\mathcal{B} f_2(x) = \mu_2(f)(x) \\ &\iff \mu_1(f) \leq_\mathcal{C} \mu_2(f) \\ &\iff \mu(f) \in L_\mathcal{C} \otimes U_\mathcal{C}. \end{aligned}$$

Analogously, if $(g, h) \in L_\mathcal{C} \otimes U_\mathcal{C}$, then $\mu^{-1}(f, g) \in \text{hom}_{\textbf{CPO}}(\mathcal{A}, \mathcal{B})$. Hence, by restricting domain and codomain of μ, we get another isomorphism of sets $\nu : \ \text{hom}_{\textbf{CPO}}(\mathcal{A}, \mathcal{B}) \to L_\mathcal{C} \otimes U_\mathcal{C}$. It remains to show that ν and ν^{-1} are monotonic. Let $f, g \in \text{hom}_{\textbf{CPO}}(\mathcal{A}, \mathcal{B})$ such that $f \leq_{\mathcal{B}^\mathcal{A}} g$, i.e., for all $x \in \mathcal{A}$, $f(x) \leq_{p,\mathcal{B}} g(x)$. By the defintion of the precision order, this means that $\nu_1(f)(x) \leq_\mathcal{B} \nu_1(g)(x) \leq_\mathcal{B} \nu_2(g)(x) \leq_\mathcal{B} \nu_2(f)(x)$ for all $x \in \mathcal{A}$. Hence, $\nu_1(f) \leq_\mathcal{C} \nu_1(g)$ and $\nu_2(g) \leq_\mathcal{C} \nu_2(f)$. It follows by definition that $\nu(f) \leq_{p,\mathcal{C}} \nu(g)$, as desired. The analogous result holds for ν^{-1} and can be shown similarly. Therefore, the corresponding morphism $\nu' : \ \mathcal{B}^\mathcal{A} \to \mathcal{C}$ between cpo's is an isomorphism. By Corollary 4.1, $\mathcal{B}^\mathcal{A} \in \textbf{LUcons}$, as desired.

□

5.2 An Approximation System

Thanks to Theorem 4.2, we can take **LUcons** as the approximation category for any application in which we wish to apply the version of AFT of Charalambidis et al. [2018]. Depending on the specific language and semantics at hand, demanded by the application, we would define a different approximation system with **LUcons**. In this subsection, we

present the approximation system for the language $\mathcal{HOL}$ and the semantics used in [Charalambidis et al., 2018] to tackle higher-order logic programs.

The language $\mathcal{HOL}$ is based on a type hierarchy $\mathbb{H}$ with base types o, the boolean type, and ι, the type of individuals. The composite types are morphism types obtained from o and ι. In particular, the types are divided into *functional types* $\sigma := \iota \mid \iota \to \sigma$, *predicate types* $\pi := o \mid \rho \to \pi$, and *parameter types* $\rho := \iota \mid \pi$. The semantics of the base types are defined as usual: $E_o := \{\mathbf{t}, \mathbf{f}\}$ with the order $\mathbf{f} \leq_o \mathbf{t}$, and $E_\iota = D$ with the trivial order ($d_1 \leq d_2$ iff $d_1 = d_2$), where D is some fixed domain for individuals. The semantics for composite types are defined following the Cartesian closed structure of **POSet**. For instance, the semantics of type $o \to o$ is simply the poset of functions from E_o to itself, i.e., $E_{o\to o} := (E_o \to E_o)$.

Since the ultimate goal of this application is studying the well-founded semantics of higher order logic programs via AFT, we are interested in the approximation space of Herbrand interpretations. Since Herbrand interpretations fix the value assigned to symbols of functional types, we only need the approximation spaces for the semantics E_π, for all predicate types π. In other terms, we can focus on the smallest subset S of Ob(**POSet**) containing E_π for all π, and closed under generalized product.

Now the definition of a suitable approximation system for S is very straightforward: we just have to define the approximation space $\mathrm{App}(E_o)$, the set of exact elements $\mathcal{E}_o$, and the projection $\mathfrak{p}_o$. All the other elements are defined inductively following the Cartesian closed structure of **LUcons**. We define: $\mathrm{App}(E_o) := E_o \otimes E_o = \langle\{(\mathbf{t},\mathbf{t}),(\mathbf{f},\mathbf{t}),(\mathbf{f},\mathbf{f})\}, \leq_p\rangle$; $\mathcal{E}_o = \{(\mathbf{t},\mathbf{t}),(\mathbf{f},\mathbf{f})\}$; and $\mathfrak{p}_o(\mathbf{t},\mathbf{t}) := \mathbf{t}$ and $\mathfrak{p}_o(\mathbf{f},\mathbf{f}) := \mathbf{f}$. Finally, given a vocabulary V for $\mathcal{HOL}$ containing symbols of predicate type, and a program $\mathfrak{P}$ over V, the approximation space of Herbrand interpretations of $\mathfrak{P}$ is $\mathcal{H}_\mathfrak{P} := \mathrm{App}\left(\Pi_{s\in V} E_{\tau(s)}\right) = \Pi_{s\in V}\,\mathrm{App}(E_{\tau(s)}) \in \mathrm{Ob}(\mathbf{Approx})$, where $\tau(s)$ is the type of the symbol s.

This greatly simplifies the construction of [Charalambidis et al., 2018]. In particular, notice that the pairs of *monotonte-antimonotonic* and *antimonotonic-monotonic* functions they defined are *precisely* the elements of the exponential objects of **LUcons**. Moreover, by changing the base types and their semantics, this approximation system can be readily adapted to suit other applications.

In conclusion, it is important to stress that we now have a clear concept of exactness: for the base type o the exact elements are $\mathcal{E}_o = \{(\mathbf{t},\mathbf{t}),(\mathbf{f},\mathbf{f})\}$, and for higher-order types, we follow Definition 4.14. The work of Charalambidis et al. [2018] lacked a notion of exactness, making it impossible to determine whether a model is actually two-valued; they discussed this question in their future work section. Let us illustrate on their example accompanying the discussion.

Example 4.3

Let $\mathfrak{P}$ be a logic program with the single rule $\mathsf{p}(\mathsf{R}) \leftarrow \mathsf{R}$, where p is a predicate of type $o \to o$ and R is a variable of type o. The space of interpretations for p is simply $\mathrm{App}(E_{o\to o}) = \mathrm{App}(E_o)^{\mathrm{App}(E_o)}$, i.e., all the monotonic functions from $\mathrm{App}(E_o) = \{(\mathbf{f},\mathbf{t}),(\mathbf{f},\mathbf{f}),(\mathbf{t},\mathbf{t})\}$ to itself, as defined above. By the semantics of Charalambidis et al. [2018], the meaning of this program is given by the interpretation (I, J), where $I(\mathsf{p})(\mathbf{t},\mathbf{t}) = J(\mathsf{p})(\mathbf{t},\mathbf{t}) = J(\mathsf{p})(\mathbf{f},\mathbf{t}) = \mathbf{t}$, and $I(\mathsf{p})(\mathbf{f},\mathbf{f}) = J(\mathsf{p})(\mathbf{f},\mathbf{f}) = I(\mathsf{p})(\mathbf{f},\mathbf{t}) = \mathbf{f}$. Since $I \neq J$, (I, J) is not exact according to the classical definition of AFT [Denecker et al., 2000] , even though we would expect to find a 2-valued model, i.e., the one assigning to p the identity function over $\{\mathbf{f},\mathbf{t}\}$. Nevertheless, according to our definition, (I, J) is indeed exact: it sends exacts of E_o to exacts of E_o. Furthermore, by the approximation system we defined in this section, it is easy to see that (I, J) represents $\mathfrak{p}_{o\to o}(I, J) = \mathcal{I} \in E_{o\to o} = (E_o \to E_o)$, where $\mathcal{I}(\mathbf{t}) = \mathbf{t}$ and $\mathcal{I}(\mathbf{f}) = \mathbf{f}$, as desired.

5.3 A New Approximator

As presented at the end of Section 2.1, the approximator of Charalambidis et al. [2018] does not provide the expected well-founded semantics for logic programs when there is an existential quantifier in the body of a rule. In this subsection, we propose a new approximator that solves this issue. We achieve this by restricting the set of elements over which certain variables can range. In particular, variables that are arguments of a predicate being defined, i.e., in the head of a rule, can range over all the elements of the approximation spaces of the corresponding types: we want to define the approximation of a higher order predicate also when the argument is an approximation. On the contrary, variables that appear exclusively in the body of a rule do not need to be approximated. In other words, a variable of type τ that is not argument of any predicate being defined, will be forced to range only over the set $\mathcal{E}_\tau$ of exact elements. For instance, in a rule of the form $\mathsf{p}(\mathsf{R}) \leftarrow \mathsf{q}(\mathsf{R},\mathsf{S})$, where p and q are predicate constants, and R and S are variables of the correct type, we let R vary over any possible approximation of its type, while we contraint S to range only over exact elements of its type.

Before stating the new definition for the approximator, we briefly recall the full syntax of $\mathcal{HOL}$. We slightly modify the one presented in [Charalambidis et al., 2018] to make it less heavy.

The *alphabet* of $\mathcal{HOL}$ consists of the following: predicate variables/constants of every predicate type π; individual variables/constants of type ι; the equality constant $\approx$ of type $\iota \to \iota \to o$ for comparing individuals of type ι; the conjunction constant $\wedge$ of type

$o \to o \to o$; the rule operator constant $\leftarrow$ of type $o \to o \to o$; and the negation constant $\neg$ of type $o \to o$.

Every predicate variable/constant and every individual variable/constant is a *term*; if E_1 is a term of type $\rho \to \pi$ and E_2 a term of type ρ then $(\mathsf{E}_1\ \mathsf{E}_2)$ is a term of type π. Every term is also an *expression*; if E is a term of type o then $(\sim\mathsf{E})$ is an expression of type o; if E_1 and E_2 are terms of type ι, then $(\mathsf{E}_1 \approx \mathsf{E}_2)$ is an expression of type o.

A *rule* of $\mathcal{HOL}$ is a formula $\mathsf{p}\ \mathsf{R}_1 \cdots \mathsf{R}_n \leftarrow \mathsf{E}_1 \wedge \ldots \wedge \mathsf{E}_m$, where p is a predicate constant of type $\rho_1 \to \cdots \to \rho_n \to o$, $\mathsf{R}_1, \ldots, \mathsf{R}_n$ are distinct variables of types $\rho_1, \ldots, \rho_n$ respectively and the E_i are expressions of type o. The term $\mathsf{p}\ \mathsf{R}_1 \cdots \mathsf{R}_n$ is the *head* of the rule and $\mathsf{E}_1 \wedge \ldots \wedge \mathsf{E}_m$ is the *body* of the rule. For the sake of simplicity, we often write $\mathsf{E}_1, \ldots, \mathsf{E}_m$, in place of $\mathsf{E}_1 \wedge \ldots \wedge \mathsf{E}_m$. A *program* $\mathfrak{P}$ of $\mathcal{HOL}$ is a finite set of rules. A *state* $\mathfrak{s}$ of a program $\mathfrak{P}$ is a function that assigns to each variable R of type ρ, an element of $\mathrm{App}(E_\rho)$, if $\rho \neq \iota$, or an element of $E_\iota = D$, if $\rho = \iota$. We denote by $\mathfrak{s}[\mathsf{R}_1/d_1, \ldots, \mathsf{R}_n/d_n]$ the state that assigns d_i to R_i for every $i \in \{1, \ldots, n\}$, and coincides with $\mathfrak{s}$ anywhere else.

Finally, we provide the definition for the three-valued semantics of Charalambidis et al. [2018] adapted to the above, slightly-modified definitions.

Definition 4.18

Let $\mathfrak{P}$ be a program, $\mathcal{I}$ an interpretation of $\mathfrak{P}$, and $\mathfrak{s}$ a state. The *three-valued semantics* of expressions and bodies is defined as follows:

1. $[\![\mathsf{c}]\!]_{\mathfrak{s}}(\mathcal{I}) = \mathsf{c}$, for every individual constant c,
2. $[\![\mathsf{p}]\!]_{\mathfrak{s}}(\mathcal{I}) = \mathcal{I}(\mathsf{p})$, for every predicate constant p,
3. $[\![\mathsf{R}]\!]_{\mathfrak{s}}(\mathcal{I}) = \mathfrak{s}(\mathsf{R})$, for every variable R,
4. $[\![(\mathsf{E}_1\ \mathsf{E}_2)]\!]_{\mathfrak{s}}(\mathcal{I}) = [\![\mathsf{E}_1]\!]_{\mathfrak{s}}(\mathcal{I})\,([\![\mathsf{E}_2]\!]_{\mathfrak{s}}(\mathcal{I}))$,
5. $[\![(\mathsf{E}_1 \bigwedge \mathsf{E}_2)]\!]_{\mathfrak{s}}(\mathcal{I}) = \mathrm{glb}_{\leq_{tr}}(\{[\![\mathsf{E}_1]\!]_{\mathfrak{s}}(\mathcal{I}), [\![\mathsf{E}_2]\!]_{\mathfrak{s}}(\mathcal{I})\})$,
6. $[\![(\neg\mathsf{E})]\!]_{\mathfrak{s}}(\mathcal{I}) = ([\![\mathsf{E}]\!]_{\mathfrak{s}}(\mathcal{I}))^{-1}$,
7. $[\![(\mathsf{E}_1 \approx \mathsf{E}_2)]\!]_{\mathfrak{s}}(\mathcal{I}) = \begin{cases} (\mathbf{t},\mathbf{t}), & \text{if } [\![\mathsf{E}_1]\!]_{\mathfrak{s}}(\mathcal{I}) = [\![\mathsf{E}_2]\!]_{\mathfrak{s}}(\mathcal{I}) \\ (\mathbf{f},\mathbf{f}), & \text{otherwise} \end{cases}$.

where $\leq_{tr}$ is the *truth order* defined by $\mathbf{f} \leq_{tr} \mathbf{u} \leq_{tr} \mathbf{t}$, and we define $(\mathbf{t},\mathbf{t})^{-1} = (\mathbf{f},\mathbf{f})$, $(\mathbf{f},\mathbf{f})^{-1} = (\mathbf{t},\mathbf{t})$ and $(\mathbf{f},\mathbf{t})^{-1} = (\mathbf{f},\mathbf{t})$.

As already explained, we want to restrict a variable appearing only in the body of a rule to range over the elements of $\mathcal{E}_\tau$, with τ being the type of the variable. We call a state $\mathfrak{s}$ *exact* if for all variables R of type $\tau \neq \iota$, it holds that $\mathfrak{s}(\mathsf{R}) \in \mathcal{E}_\tau$. We denote by $\mathcal{E}^{states}$ the set of exact states.

We have now all the elements to introduce the definition of three-valued models and the new approximator, i.e., the *three-valued immediate consequence operator*. For the sake of simplicity, in the following we define $\mathrm{App}(E_\iota) := E_\iota = D$, even though $E_\iota \notin S$ and has no associated approximation space.

Definition 4.19

Let $\mathfrak{P}$ be a program. An interpretation $\mathcal{I} \in \mathcal{H}_\mathfrak{P}$ is a *(three-valued) model* of $\mathfrak{P}$ iff for every rule $(\mathsf{p}\ \mathsf{R}_1 \cdots \mathsf{R}_n \leftarrow \mathsf{E})$ in $\mathfrak{P}$, for all $d_1 \in \mathrm{App}(E_{\rho_1}), \ldots, d_n \in \mathrm{App}(E_{\rho_n})$, and for all $s \in \mathcal{E}^{states}$, it holds that $[\![\mathsf{E}]\!]_{s[\mathsf{R}_1/d_1,\ldots,\mathsf{R}_n/d_n]}(\mathcal{I}) \leq_{tr} [\![\mathsf{p}\ \mathsf{R}_1 \cdots \mathsf{R}_n]\!]_{s[\mathsf{R}_1/d_1,\ldots,\mathsf{R}_n/d_n]}(\mathcal{I})$.

Definition 4.20

Let $\mathfrak{P}$ be a program. The *three-valued immediate consequence operator* $\Psi^\mathfrak{P} : \mathcal{H}_\mathfrak{P} \to \mathcal{H}_\mathfrak{P}$ is defined for every predicate constant $\mathsf{p} : \rho_1 \to \cdots \to \rho_n \to o$ in $\mathfrak{P}$, and for all $d_1 \in \mathrm{App}(E_{\rho_1}), \ldots, d_n \in \mathrm{App}(E_{\rho_n})$, as:

$$\Psi^\mathfrak{P}(\mathcal{I})(\mathsf{p})\, d_1 \cdots d_n = \mathrm{lub}_{\leq_{tr}}(\{[\![\mathsf{E}]\!]_{s[\mathsf{R}_1/d_1,\ldots,\mathsf{R}_n/d_n]}(\mathcal{I}) \mid s \in \mathcal{E}^{states} \text{ and } (\mathsf{p}\ \mathsf{R}_1 \cdots \mathsf{R}_n \leftarrow \mathsf{E}) \text{ in } \mathfrak{P}\}).$$

With Definition 4.20, we solve the issue linked to existential quantifiers. Let us review the example presented in Subsection 2.1. We considered a program $\mathfrak{P}$ with just one rule $\mathsf{p} \leftarrow \mathsf{R} \wedge \neg\mathsf{R}$, where p is a predicate constant of type o, and R is a variable of type o. Observe that in this case the space of Herbrand interpretations is just $\mathcal{H}_\mathfrak{P} := \mathrm{App}(E_o) \in \mathrm{Ob}(\mathbf{Approx})$, as we are only interested in the interpretation of the predicate p. For all interpretations $\mathcal{I} \in \mathcal{H}_P$, we have

$$\begin{aligned}\Psi^\mathfrak{P}(\mathcal{I})(\mathsf{p}) &= \mathrm{lub}_{x\leq_{tr}}(\{[\![\mathsf{E}]\!]_s(\mathcal{I}) \mid s \in \mathcal{E}^{states} \text{ and } (\mathsf{p} \leftarrow \mathsf{E}) \in \mathfrak{P}\}) = \\ &= \mathrm{lub}_{\leq_{tr}}(\{[\![\mathsf{R} \wedge \neg\mathsf{R}]\!]_s(\mathcal{I}) \mid s \in \mathcal{E}^{states}\}) = \mathrm{lub}_{\leq_{tr}}(\{s(\mathsf{R}) \wedge s(\mathsf{R})^{-1} \mid s \in \mathcal{E}^{states}\}) = \\ &= \mathrm{lub}_{\leq_{tr}}(\{(\mathbf{f},\mathbf{f}) \wedge (\mathbf{t},\mathbf{t}), (\mathbf{t},\mathbf{t}) \wedge (\mathbf{f},\mathbf{f})\}) = (\mathbf{f},\mathbf{f}).\end{aligned} \tag{4.2}$$

Notice that for this specific program $\mathfrak{P}$, both the approximator $\Psi^\mathfrak{P}$ and the old version from [Charalambidis et al., 2018] do not depend on the interpretation $\mathcal{I}$, but only on the states. While the approximator of Charalambidis et al. [2018] considered any possible

state, even the one sending R to $(\mathbf{f},\mathbf{t})$, $\Psi^{\mathfrak{P}}$ takes into account only exact states, i.e., R can only be sent to an element of $\mathcal{E}_o = \{(\mathbf{f},\mathbf{f}),(\mathbf{t},\mathbf{t})\}$. This limitation removes the formula $(\mathbf{f},\mathbf{t}) \wedge (\mathbf{f},\mathbf{t}) = (\mathbf{f},\mathbf{t})$ from the least upper bound computation in (4.2), which was the one causing the evaluation of p being $(\mathbf{f},\mathbf{t})$ for the approximator of Charalambidis et al. [2018].

Since the approximator $\Psi^{\mathfrak{P}}$ does not depend on the interpretation, it is immediate to see that the stable revision operator $\mathcal{S}_{\Psi^{\mathfrak{P}}}$ coincides with the approximator $\Psi^{\mathfrak{P}}$ for all $(I,J) \in \mathcal{H}_{\mathfrak{P}}$:

$$\mathcal{S}_{\Psi^{\mathfrak{P}}}(I,J) = \left(\mathrm{lfp}(\Psi^{\mathfrak{P}}(\cdot,J)_1), \mathrm{lfp}(\Psi^{\mathfrak{P}}(I,\cdot)_2)\right) = \left(\Psi^{\mathfrak{P}}(\cdot,J)_1, \Psi^{\mathfrak{P}}(I,\cdot)_2\right) = \Psi^{\mathfrak{P}}(I,J).$$

It follows that $\mathcal{S}_{\Psi^{\mathfrak{P}}}$ does not depend on the interpretation either. Thus, the least fixpoint of $\mathcal{S}_{\Psi^{\mathfrak{P}}}$, which corresponds to the well-founded model of $\mathfrak{P}$, is just the interpretation sending p to $(\mathbf{f},\mathbf{f})$, resulting in a sensible account for the well-founded sementics. Moreover, observe that this interpretation is also exact by our new definition, and it corresponds to the unique exact stable model of the program $\mathfrak{P}$.

In the remainder of this section, we present two examples that highlight the importance of enabling the application of approximate objects to approximate objects.

Example 4.4

Consider an undirected graph given by a predicate node : $\iota \to o$, containing all the nodes of the graph, and a predicate edge : $\iota \to \iota \to o$ defining the edge relation, which we assume to be symmetric. Some nodes of the graph are *marked*. We call a non-empty set S of nodes a *covering* if for every marked node n there exists a node in S with an edge to n. Now, suppose that a Player can modify the set of marked nodes by swapping a marked node with a neighbouring, unmarked node. The goal of the Player is reached when the set of marked nodes is a covering. At that point, the game is over and the Player cannot swap nodes anymore.

The key predicates of our example are contained in Listing 4.1. We have defined them in terms of a time parameter T of type ι, assuming that the Player can only do one swap at a time. In particular, here are the signatures and meanings of the predicates of Listing 4.1:

- swap : $\iota \to \iota \to \iota \to o$ indicates whether at a certain time, two nodes are swapped;
- add : $\iota \to \iota \to o$ (resp. remove : $\iota \to \iota \to o$) tells whether a node is added (resp. removed) at a certain time;

- `marked` : $\iota \to \iota \to o$ tells whether a node at a specific time is marked (Lines 6 to 8). Note that `marked T` : $\iota \to o$ represents the set of all marked nodes at time `T`;
- `succ` : $\iota \to \iota \to o$ tells whether a time point is the successor of another time point (if `succ T' T` holds, then `T` is the successor of `T'`);
- `gameOver` : ι expresses whether the game is over at a certain time (Line 21);
- `nextTo` : $(\iota \to o) \to \iota \to o$ tells whether a node is connected by an edge to some node of a set of nodes;
- `edge` : $\iota \to \iota \to o$ indicates whether two nodes are connected by an edge;
- `subset` : $(\iota \to o) \to (\iota \to o) \to o$ (resp. `nonsubset` : $(\iota \to o) \to (\iota \to o) \to o$) tells whether a set is (resp. is not) a subset of another set;
- `empty` : $(\iota \to o) \to o$ (resp. `nonempty` : $(\iota \to o) \to o$) tells whether a set is (resp. is not) empty;
- `node` : $\iota \to o$ indicates whether an atom is a node;
- `covering` : $\iota \to (\iota \to o) \to o$ (resp. `ncovering` : $\iota \to (\iota \to o) \to o$) tells whether at a certain time a set of nodes is (respis not) a covering (Lines 16, 17, and 18).

Listing 4.1: Graph Game.

```
1  % We define predicates to add/remove nodes to/from marked
       ↪ set.
2  add T X  ← swap T X Y
3  remove T Y ← swap T X Y
4
5  % We define the set of marked nodes based on the marked
       ↪ nodes at the previous time point and the last swap.
6  marked T X ← succ T T', ¬(gameOver T'), marked T' X,
       ↪ ¬(remove T' X).
7  marked T X ← succ T T', ¬(gameOver T'), add T' X.
8  marked T X ← succ T T', gameOver T', marked T' X.
9
10 % We define what a covering of a set of nodes is.
11 nextTo S X ← S Y, edge Y X.
12 nonsubset S Q ← S X, ¬(Q X).
13 subset S Q ←  ¬(nonsubset S Q).
14 nonempty S ← S X.
15 empty S ← nonempty S.
16 ncovering T S ← empty S.
17 ncovering T S ← marked T X, ¬(nextTo S X).
18 covering T S ← subset S node, ¬(ncovering T S).
19
20 % We define when the game ends.
21 gameOver T ← covering T (marked T).
```

In the formalization of Charalambidis et al. [2018], natural numbers are not taken into account. Thus, we regard the time variable T as an individual variable of type ι and we limit our example to only four time points, expressed by the individual constants a, b, c, and d, related by the successor relation succ : $\iota \to \iota \to o$, as expressed in Listing 4.2. In the same Listing, we also instantiate the nodes and edges of the graph that we chose for this example, and the initial set of marked nodes, i.e., `marked a`.

Listing 4.2: Instantiation of time points, graph's nodes and edges, and initial set of marked nodes.

```
% Time points a, b, c, and d: b successor of a, c of b, and
    ↪ d of c.
time a.
time b.
time c.
time d.
succ b a.
succ c b.
succ d c.
% Nodes.
node x.
node y.
node z.
node u.
node v.
% Edges.
edge x y.
edge x z.
edge x u.
edge z v.
edge u v.
% Marked nodes at the start of the game (time point a).
marked a y.
marked a u.
```

We are only missing the swaps the Player makes at each of the three time points. For example, we could have those listed in Listing 4.3.

Listing 4.3: Swaps.

```
% Nodes x, y, z, u, and v, and swaps at the time points a, b
    ↪ , and c.
swap a v u.
swap b x y.
swap c z x.
```

By joining Listings 4.1, 4.2, and 4.3, we obtain a program $\mathfrak{P}$ encoding a specific run of the game.

Using the machinery of AFT, we can easily find the well-founded, the stable, the Kripke-Kleene, and the supported models of $\mathfrak{P}$. Let us see in some more detail how to obtain the Kripke-Kleene model. This can be achieved very easily: we just have to iteratively apply the three-valued immediate consequence operator $\Psi^{\mathfrak{P}}$ of Definition 4.20 starting from the bottom of $\mathcal{H}_{\mathfrak{P}}$, i.e., the interpretation $\mathcal{I}_0$ that sends every atom to $\mathbf{u}$. After the first application of $\Psi^{\mathfrak{P}}$, we obtain another interpretation $\mathcal{I}_1 := \Psi^{\mathfrak{P}}(\mathcal{I}_0)$:

- Since the predicates `time`, `succ`, `node`, `edge`, and `swap` appear only in rules with empty bodies (Listings 4.2 and 4.3), $\mathcal{I}_1$ already provides a two valued interpretation for them. For instance,

$$\mathcal{I}_1(\mathsf{time})\ \mathsf{a} = \mathcal{I}_1(\mathsf{time})\ \mathsf{b} = \mathcal{I}_1(\mathsf{time})\ \mathsf{c} = \mathcal{I}_1(\mathsf{time})\ \mathsf{d} = \mathbf{t},$$
$$\mathcal{I}_1(\mathsf{time})\ \mathsf{i} = \mathbf{f}, \qquad \text{for all } \mathsf{i} \in E_\iota \setminus \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}\}.$$

- Thanks to the rules in Listing 4.2, we have $\mathcal{I}_1(\mathtt{marked})\ \mathtt{a}\ \mathtt{y} = \mathcal{I}_1(\mathtt{marked})\ \mathtt{a}\ \mathtt{u} = \mathbf{t}$. However, since `marked` appears in the head of some rules in Listing 4.1 as well, it holds that $\mathcal{I}_1(\mathtt{marked})\ \mathtt{i}_1\ \mathtt{i}_2 = \mathbf{u}$ for all $(\mathtt{i}_1, \mathtt{i}_2) \in E_\iota \times E_\iota \setminus \{(\mathsf{a}, \mathsf{y}), (\mathsf{a}, \mathsf{u})\}$.
- All the remaining predicates defined in Listing 4.1 are still interpreted as fully undefined by $\mathcal{I}_1$.

Intuitively, $\mathcal{I}_1$ only provides the information regarding a possible initial setup of the game, namely the one determined by the rules in Listing 4.2, and the moves of a possible run, namely those specified by the swaps in Listing 4.3. By the iterative application of $\Psi^{\mathfrak{P}}$, we simulate the full run of the game. Let us see in detail how the predicates `marked` and `gameOver` are evaluated at each application of $\Psi^{\mathfrak{P}}$, until we reach the least fixpoint of $\Psi^{\mathfrak{P}}$, i.e., the Kripke-Kleene model. In the tables below, we will denote by $\mathcal{I}_{n+1}$ the interpretation $\Psi^{\mathfrak{P}}(\mathcal{I}_n)$ for all $n \in \mathbb{N}$. We only specify for which individual constants $\mathtt{i}_1, \mathtt{i}_2$, and $\mathtt{i}$ the atoms $\mathtt{marked}\ \mathtt{i}_1\ \mathtt{i}_2$ and $\mathtt{gameOver}\ \mathtt{i}$ are interpreted as true or undefined via $\mathcal{I}_n$ for all $n \in \mathbb{N}$ (for all the other individual constants such atoms are interpreted as false).

n	$\{(\mathtt{i}_1, \mathtt{i}_2) \in E_\iota^2 \mid \mathcal{I}_n(\mathtt{marked})\mathtt{i}_1\mathtt{i}_2 = \mathbf{t}\}$	$\{(\mathtt{i}_1, \mathtt{i}_2) \in E_\iota^2 \mid \mathcal{I}_n(\mathtt{marked})\mathtt{i}_1\mathtt{i}_2 = \mathbf{u}\}$
0	$\emptyset$	$E_\iota \times E_\iota$
1	$\{(\mathsf{a},\mathsf{y}),(\mathsf{a},\mathsf{u})\}$	$E_\iota \times E_\iota \setminus \{(\mathsf{a},\mathsf{y}),(\mathsf{a},\mathsf{u})\}$
2	$\{(\mathsf{a},\mathsf{y}),(\mathsf{a},\mathsf{u})\}$	$\{\mathsf{b}\} \times E_\iota \cup \{\mathsf{c}\} \times E_\iota \cup \{\mathsf{d}\} \times E_\iota$
3	$\{(\mathsf{a},\mathsf{y}),(\mathsf{a},\mathsf{u})\}$	$\{(\mathsf{b},\mathsf{y}),(\mathsf{b},\mathsf{u}),(\mathsf{b},\mathsf{v})\}$ $\cup \{\mathsf{c}\} \times E_\iota \cup \{\mathsf{d}\} \times E_\iota$
4	$\{(\mathsf{a},\mathsf{y}),(\mathsf{a},\mathsf{u}),(\mathsf{b},\mathsf{y}),(\mathsf{b},\mathsf{v})\}$	$\{(\mathsf{c},\mathsf{y}),(\mathsf{c},\mathsf{u}),(\mathsf{c},\mathsf{v}),(\mathsf{c},\mathsf{x})\} \cup \{\mathsf{d}\} \times E_\iota$
5	$\{(\mathsf{a},\mathsf{y}),(\mathsf{a},\mathsf{u}),(\mathsf{b},\mathsf{y}),(\mathsf{b},\mathsf{v})\}$	$\{(\mathsf{c},\mathsf{y}),(\mathsf{c},\mathsf{v}),(\mathsf{c},\mathsf{x})\}$ $\cup \{(\mathsf{d},\mathsf{y}),(\mathsf{d},\mathsf{u}),(\mathsf{d},\mathsf{v}),(\mathsf{d},\mathsf{x}),(\mathsf{d},\mathsf{z})\}$
6	$\{(\mathsf{a},\mathsf{y}),(\mathsf{a},\mathsf{u}),(\mathsf{b},\mathsf{y}),$ $(\mathsf{b},\mathsf{v}),(\mathsf{c},\mathsf{x}),(\mathsf{c},\mathsf{v})\}$	$\{(\mathsf{d},\mathsf{y}),(\mathsf{d},\mathsf{v}),(\mathsf{d},\mathsf{x}),(\mathsf{d},\mathsf{z})\}$
7	$\{(\mathsf{a},\mathsf{y}),(\mathsf{a},\mathsf{u}),(\mathsf{b},\mathsf{y}),$ $(\mathsf{b},\mathsf{v}),(\mathsf{c},\mathsf{x}),(\mathsf{c},\mathsf{v})\}$	$\{(\mathsf{d},\mathsf{v}),(\mathsf{d},\mathsf{x}),(\mathsf{d},\mathsf{z})\}$
≥ 8	$\{(\mathsf{a},\mathsf{y}),(\mathsf{a},\mathsf{u}),(\mathsf{b},\mathsf{y}),(\mathsf{b},\mathsf{v}),$ $(\mathsf{c},\mathsf{x}),(\mathsf{c},\mathsf{v}),(\mathsf{d},\mathsf{z}),(\mathsf{d},\mathsf{v})\}$	$\emptyset$

n	$\{\mathtt{i} \in E_\iota \mid \mathcal{I}_n(\mathtt{gameOver})\mathtt{i} = \mathbf{t}\}$	$\{\mathtt{i} \in E_\iota \mid \mathcal{I}_n(\mathtt{gameOver})\mathtt{i} = \mathbf{u}\}$
0, 1, 2	$\emptyset$	E_ι
3, 4	$\emptyset$	$\{\mathsf{b},\mathsf{c},\mathsf{d}\}$
5, 6	$\emptyset$	$\{\mathsf{c},\mathsf{d}\}$
7, 8	$\emptyset$	$\{\mathsf{d}\}$
≥ 9	$\{\mathsf{d}\}$	$\emptyset$

The interpretation $\mathcal{I}_9$, obtained by applying nine times the three-valued immediate consequence operator, is the least fixpoint of $\Psi^{\mathfrak{P}}$, thus, $\mathcal{I}_9$ is the Kripke-Kleene fixpoint of $\mathfrak{P}$. It is not difficult to check that $\mathcal{I}_9$ is exact. In particular, $\mathcal{I}_9$ represents a unique two-valued Herbrand interpretation I_9. Since the Kripke-Kleene fixpoint approximates every supported fixpoint, any possible supported fixpoint of $\mathfrak{P}$ is also exact and represents the same two-valued interpretation I_9. The same holds for the well-founded fixpoint and the stable fixpoints, since the well-founded fixpoint is always at least as precise as the Kripke-Kleene fixpoint and it approximates every stable fixpoint.

Alternatively, to obtain the well-founded model, we could compute the least fixpoint of the stable revision operator of the approximator contained in Definition 4.20, i.e., the least fixpoint of $\mathcal{S}_{\Psi^{\mathfrak{P}}} : (x, y) \mapsto (\mathrm{lfp}(\Psi^{\mathfrak{P}}{}_1(\cdot, y)), \mathrm{lfp}(\Psi^{\mathfrak{P}}{}_1(\cdot, x)))$. Since the stable revision operator is monotonic, to find its least fixpoint it is sufficient to repeatedely

apply $\mathcal{S}_{\Psi^{\mathfrak{P}}}$ starting from the bottom element of its domain, i.e., $\mathcal{I}_0$, similarly to what we did to compute the Kripke-Kleene fixpoint above.

Notice from the tables above that during the first iterative applications of either the three-valued immediate consequence operator $\Psi^{\mathfrak{P}}$ or the stable revision operator $\mathcal{S}_{\Psi^{\mathfrak{P}}}$, the predicates `marked`, `covering`, and `gameOver` are being defined only for the first time points, i.e., they are partially defined. In other words, the three-valued interpretations that we obtain from the first computations leading to the fixpoint, send the aforementioned predicates to approximate objects of the respective approximation spaces. Since `marked`, `covering`, and `gameOver` are all defined by mutual induction, we are forced to apply an approximate object on another approximate object. In particular, in Line 21 of Listing 4.1, for the definition of `gameOver`, the predicate `covering` is applied on `marked`. Only when the fixpoint is reached, all the predicates being defined will be exact, i.e., two-valued.

In Listing 4.3, we have provided a specific set of swaps the Player makes. We can obtain a more general setup by using choice rules to define the predicate `swap`, as we do in Listing 4.4.

Listing 4.4: Choice Rules.

```
1   % Choice rules: the user can swap one edge with a neighbour
        ↪ at each time.
2   swap T X Y ← node X, node Y, time T, ¬(nswap T X Y).
3   nswap T X Y ← node X, node Y, time T, ¬(swap T X Y).
4   % First element must be in the marked set, second cannot
5   ← swap T X Y, ¬(marked T X).
6   ← swap T X Y, marked T Y.
7   % At most one swap at a time.
8   ← swap T X Y, swap T X' Y', ¬(X' ≈ X).
9   ← swap T X Y, swap T X' Y', ¬(Y' ≈ Y).
10  % Can only swap neighbors.
11  ← swap T X Y, ¬(edge X Y).
```

By joining Listings 4.1, 4.2, and 4.4, we obtain another program $\mathfrak{P}'$, and we can again compute the models of interest via AFT. In particular, now each stable model corresponds to a possible run of the game with starting set of marked nodes `marked a`. Notice that, because of the choice rules in Listing 4.4, the Kripke-Kleene and the well-founded model of $\mathfrak{P}'$ leave most of the predicates undefined. In fact, if we start from the bottom of $\mathcal{H}'_{\mathfrak{P}}$, which we denote by $\mathcal{J}_0$, we only have to apply to the three-valued immediate consequence operator $\Psi^{\mathfrak{P}'}$ three times to obtain the Kripke-Kleene fixpoint. In the two tables below, we keep track of the evaluations

of the predicates marked and gameOver under the interpretation $\mathcal{J}_n$ for each $n \in \mathbb{N}$, where $\mathcal{J}_{n+1} := \Psi^{\mathfrak{P}'}(\mathcal{J}_n)$.

n	$\{(\mathtt{i}_1, \mathtt{i}_2) \in E_\iota^2 \mid \mathcal{J}_n(\mathtt{marked})\mathtt{i}_1\mathtt{i}_2 = \mathbf{t}\}$	$\{(\mathtt{i}_1, \mathtt{i}_2) \in E_\iota^2 \mid \mathcal{J}_n(\mathtt{marked})\mathtt{i}_1\mathtt{i}_2 = \mathbf{u}\}$
0	$\emptyset$	$E_\iota \times E_\iota$
1	$\{(\mathsf{a}, \mathsf{y}), (\mathsf{a}, \mathsf{u})\}$	$E_\iota \times E_\iota \setminus \{(\mathsf{a}, \mathsf{y}), (\mathsf{a}, \mathsf{u})\}$
≥ 2	$\{(\mathsf{a}, \mathsf{y}), (\mathsf{a}, \mathsf{u})\}$	$\{\mathsf{b}\} \times E_\iota \cup \{\mathsf{c}\} \times E_\iota \cup \{\mathsf{d}\} \times E_\iota$

n	$\{\mathtt{i} \in E_\iota \mid \mathcal{J}_n(\mathtt{gameOver})\mathtt{i} = \mathbf{t}\}$	$\{\mathtt{i} \in E_\iota \mid \mathcal{J}_n(\mathtt{gameOver})\mathtt{i} = \mathbf{u}\}$
$0, 1, 2$	$\emptyset$	E_ι
≥ 3	$\emptyset$	$\{\mathsf{b}, \mathsf{c}, \mathsf{d}\}$

It is not hard to show that $\mathcal{J}_3$ is not only the Kripke-Kleene model of $\mathfrak{P}'$, but also its well-founded model.

Example 4.5

Let us consider a manufacturing company that aims at growing and diversifying its production. We represent raw materials with individual constants of type ι, and finished products with predicate constants of type $\iota \to o$, such that if P is any finished product, and M is any raw material, then P M is true if and only if M is necessary to craft P. We denote by materials : $\iota \to o$ the predicate corresponding to the set of all raw materials, and by products : $(\iota \to o) \to o$ the predicate corresponding to the set of all finished products.

We want to define a predicate production : $\iota \to \iota \to o$ (Lines 26 and 28 in Listing 4.5) that indicates which raw materials the company has to acquire for production at a certain time: production T M is true if and only if the company acquires the material M at time T. As in Example 4.4, we regard the time variable T as an individual variable of type ι and we limit our program in Listing 4.5 to only four time points, related by the successor relation (Lines 2 to 8 in Listing 4.5). We fix the initial set production a : $\iota \to o$ of materials the company starts with. At each time point, the company decides which new materials to acquire: we encode the information about these potential new ingredients with the predicate candidates : $\iota \to (\iota \to o) \to o$ (Lines 20 and 22 in Listing 4.5), which takes as argument a time point, i.e., an individual variable, and a set of materials, i.e., a predicate of type $\iota \to o$. The selection of new materials the company takes into consideration obeys a few criteria:

1. *Maximize profit*: products necessitating more raw materials to be crafted require more expertise and more capital to invest, but they provide more profit. Hence, as time progresses, the company aims at products more and more complex: at time `T`, a set `P` of raw materials is a candidate if it can be covered by sets corresponding to some finished products of complexity `T` (Lines 16, 18, and 20 of Listing 4.5). We assume a constant predicate `complexity` : $(\iota \to o) \to \iota \to o$ indicating the complexity of a product is given (Lines 21 to 25 in Listing 4.7).

2. *Cautiousness*: producing items using only new materials may be risky and time consuming, as the manifacturing team has to acquire novel knowledge, and new suppliers for the raw materials need to be found. Hence, the newly accepted products are required to share at least one raw material with a product in production at the prevous time point (Lines 10, 16, 18, and 20 in Listing 4.5).

3. *Efficient growth*: as time passes and the company produces more complex items, older, simpler products can be put out of production. This is done gradually: if at time `T'` every material in `production T'` is needed to produce some product of complexity `T'`, then the materials in `production T'` are still kept for the production at the successor time point `T`; otherwise, all the materials of `production T'` that are not needed to craft a product of complexity `T` are dropped (Lines 22 in Listing 4.5).

Finally, `production` is just the union of all the candidates (Line 26 in Listing 4.5). If there are no candidates at a certain time point, the production does not vary at the next time point, and the company ends its expansion (Lines 24 and 28 in Listing 4.5).

Note that, in Listing 4.5, the symbols `T`, `M`, and `X`, highlighted in blue, are variables of type ι, whereas the symbols `Q`, `R`, and `S`, highlighted in red, are predicate variables of type $\iota \to o$ that are used in a higher-order style in the rules in Lines 10, 12, 18, and 26 in Listing 4.5. Variables that appear in a rule but not in its head are existentially quantified.

Listing 4.5: The growth of the manifacturing company.

```
1  % a, b, c, and d are time points, b is the successor of a, c
       ↪ of b, and d of c.
2  time a.
3  time b.
4  time c.
5  time d.
6  succ b a.
7  succ c b.
```

```
8         succ d c.
9         % intersect Q R is true if the intersection between Q and R
              ↪ is non-empty.
10        intersects Q R ← Q X, R X.
11        % subset Q R (nonsubset Q R) is true if R is (is not) a
              ↪ subset of Q.
12        nonsubset Q R ← Q X, ¬(R X).
13        subset Q R ←  ¬(nonsubset Q R).
14
15        % There exists a product P made of T raw materials, all
              ↪ belonging to S, some belonging to production T', and
              ↪ including M.
16        existsubprod T S M ← products P, complexity P T, P M,
              ↪ subset P S,  intersects P (production T'), succ T T'.
17        % S cannot be covered by products of size T if there exists
              ↪ a material M in S that is never part of a subproduct
              ↪ of S.
18        notcovered T S ← S M,  ¬(existsubprod T S M), time T.
19        % S is a candidate at time T if it can be covered and it is
              ↪ a set of materials (maximize profit+cautiousness).
20        candidates T S ← time T, subset S materials,
              ↪ ¬notcovered T S.
21        % S is a candidate at time T if it is a product of size T',
              ↪ in production at time T', and if in production at time
              ↪  T' there were only materials needed for products of size
              ↪  T' (efficient growth).
22        candidates T S ← products S, complexity S T',
              ↪ subset S (production T'), succ T T',
              ↪ candidates T' (production T').
23        % There exists a candidate at time T.
24        existcandidate T ← candidates T S.
25        % A material is needed for production at time T if it is in
              ↪ one of the candidate sets at time T.
26        production T M ← candidates T S, S M.
27        % If there are no candidates at time T, then the production
              ↪ remains the same.
28        production T M ← ¬(existcandidate T), production T' M,
              ↪ succ T T'.
```

Let us see how to obtain the fixpoints of interest with an instantiation of the predicates `materials`, `products`, `production a`, and `complexity`. Suppose the company manufactures furniture, and that the materials are wood (`w`), steel (`s`), plastic (`p`), glass (`g`), fabric (`f`), brass (`br`), and cushioning foam (`cf`) (Lines 2 to 8 of Listing 4.6). We assume the company starts with only wood as material (Line 10 of Listing 4.6).

Listing 4.6: The possible materials of the furniture company.

```
% w, s, p, g, f, br, and cf are materials.
materials w.
materials s.
materials p.
materials g.
materials f.
materials br.
materials cf.
% The production at time a only uses w.
production a w.
```

It remains to define the products and their complexity. We consider the following five products (Lines 2 to 6 in Listing 4.7):

- a chair (`chair`) made entirely of wood;
- a nightstand (`nightstand`) made of wood (structure), and brass (hinges and handles);
- a table (`table`) made of steel (structure), and wood (horizontal top);
- a glass cabinet (`gcabinet`) made of wood (structure), brass (hinges and handles), and glass (doors);
- and an office chair (`ochair`) made of steel (structure), plastic (wheels), cushioning foam (cushions), and fabric (cover).

Finally, we define the complexity of a product as the number of different materials needed to craft it. For this purpose, we assume that the four time points `a`, `b`, `c`, and `d` that we consider in this example coincide with 1, 2, 3, and 4, respectively.

Listing 4.7: The possible products and their complexity.

```
%We define the possible products.
products chair.
products nightstand.
products table.
products gcabinet.
products ochair.
%We specify the materials the products are made of.
chair w.
nightstand w.
nightstand br.
table s.
table w.
gcabinet w.
gcabinet br.
gcabinet g.
ochair s.
```

```
17        ochair p.
18        ochair cf.
19        ochair f.
20        %We define the complexity of the products depending on the
              ↪ amount of different materials they need to be crafted.
21        complexity chair a.
22        complexity nightstand b.
23        complexity table b.
24        complexity gcabinet c.
25        complexity ochair d.
```

By joining Listings 4.5, 4.6, and 4.7, we obtain a program $\mathfrak{P}$. The Kripke-Kleene and the well-founded model can be obtained easily by iterative application of the three-valued immediate consequence operator $\Psi^{\mathfrak{P}}$ and its stable revision operator $\mathcal{S}_{\Psi^{\mathfrak{P}}}$, respectively, starting from the three-valued interpretation sending every atom to $\mathbf{u}$. It is not difficult to see that the Kripke-Kleene fixpoint and the well-founded fixpoint are exact and all the fixpoints of interest (Kripke-Kleene, supported, well-founded, and stable) represent the same two valued model I. For the sake of simplicity, we only write below the interpretation of the predicates `production` and `candidates`. For all $\mathsf{t}, \mathsf{m} \in E_\iota$, for all $\mathsf{s} \in E_{\iota\to o}$, we have:

$$
\begin{aligned}
I(\mathsf{production})\ \mathsf{t}\ \mathsf{m} = \mathbf{t} \iff (\mathsf{t},\mathsf{m}) \in\ & \{(\mathsf{a},\mathsf{w})\} \\
& \cup \{\mathsf{b}\} \times \{\mathsf{w},\mathsf{s},\mathsf{br}\} \\
& \cup \{\mathsf{c}\} \times \{\mathsf{w},\mathsf{s},\mathsf{br},\mathsf{g}\} \\
& \cup \{\mathsf{d}\} \times \{\mathsf{s},\mathsf{f},\mathsf{cf},\mathsf{p}\} \\
I(\mathsf{candidates})\ \mathsf{t}\ \mathsf{s} = \mathbf{t} \iff (\mathsf{t},\mathsf{s}) \in\ & \{\mathsf{b}\} \times \{\mathsf{table},\mathsf{nightstand},\{\mathsf{w},\mathsf{s},\mathsf{br}\}\} \\
& \cup \{\mathsf{c}\} \times \{\mathsf{gcabinet},\{\mathsf{w},\mathsf{s},\mathsf{br}\}\} \\
& \cup \{(\mathsf{d},\mathsf{ochair})\}
\end{aligned}
$$

Similarly to what happens for Example 4.4, since the predicates `candidates` and `production` are defined by mutual induction, they will be fully defined only at the end of the least fixpoint construction (for both the Kripke-Kleene and the well-founded fixpoint). However, in order to define `candidates` we need to apply it to `production` (Line 22). In particular, before reaching the Kripke-Kleene or the well-founded fixpoint, we will need to apply an approximate object, namely `candidates`, on another approximate object, i.e., `production`.

6 Different Approaches to Higher-Order Consistent AFT for LP

In this section, we compare two approaches [Charalambidis et al., 2018; Bogaerts et al., 2024] to higher-order consistent AFT for LP and provide formal claims and conjectures about their relation.

In the previous section, we have shown how our novel categorical framework perfects and extends the one proposed by Charalambidis et al. [2018] for higher-order LP. Charalambidis et al. [2018] defined higher-order approximation spaces directly in terms of the set of morphisms from one approximation space to another. For instance, in Example 4.3, we have seen that the approximation space $\mathrm{App}(E_{o\to o})$ for predicates of type $o \to o$ is the set of monotonic functions from the approximation space for objects of type o to itself, i.e., from $\{(\mathbf{f},\mathbf{t}),(\mathbf{f},\mathbf{f}),(\mathbf{t},\mathbf{t})\}$ to itself.

However, this is not the only way to extend consistent AFT so that it can be applied to higher-order LP: Bogaerts et al. [2024] did so by following another route. The main difference lies in the definition of the approximation spaces for what we called *morphism types*, i.e., higher-order types of the form $\tau_1 \to \tau_2$, and the resultant three-valued semantics. In [Bogaerts et al., 2024], the approximation space for a type $\tau_1 \to \tau_2$ is the set of all functions from the semantics of τ_1 to the approximation space for the semantics of τ_2.

Example 4.6

In [Bogaerts et al., 2024], the approximation space for predicates of type $o \to o$ is the set of all functions from $\{\mathbf{f},\mathbf{t}\}$ to $\{(\mathbf{f},\mathbf{t}),(\mathbf{f},\mathbf{f}),(\mathbf{t},\mathbf{t})\}$.

This should come as a surprise for the attentive reader: we have expressed the need to apply approximate objects on approximate objects, but here the domains of the functions comprise only two-valued, non-approximated elements ($\mathbf{f}$ and $\mathbf{t}$ in Example 4.6). It follows that the three-valued semantics of expressions of the form $(\mathsf{E}_1\ \mathsf{E}_2)$ cannot be defined as a simple application like in our framework (see Item 4 of Definition 4.18 in Chapter 4), but it resorts to a clever trick, similar to some sort of ultimate construction [Denecker et al., 2004]: first, one has to apply the approximate interpretation of E_1 to (the projection of) each exact element approximated by the interpretation of E_2, and then compute the greatest lower bound of the values obtained at the previous step.

Example 4.7

(Example 4.6 continued). Let us consider the framework of [Bogaerts et al., 2024], and let E_1 be an expression of type $o \to o$ and E_2 an expression of type o. Assume that,

given a three-valued interpretation $\mathcal{I}$, the semantics of E_2 is the value $(\mathbf{f}, \mathbf{t})$, and the semantics of E_1 is a function $f : \{\mathbf{f}, \mathbf{t}\} \to \{(\mathbf{f}, \mathbf{t}), (\mathbf{f}, \mathbf{f}), (\mathbf{t}, \mathbf{t})\}$ sending $\mathbf{f}$ to $(\mathbf{f}, \mathbf{f})$, and $\mathbf{t}$ to $(\mathbf{f}, \mathbf{t})$. To obtain the semantics of $(\mathsf{E}_1\ \mathsf{E}_2)$ we perform the following steps:

1. Consider the exact elements approximated by the semantics of E_2, i.e., by $(\mathbf{f}, \mathbf{t})$. In this case, we have $(\mathbf{f}, \mathbf{f})$ and $(\mathbf{t}, \mathbf{t})$.
2. Take the image via the projection function: $\mathfrak{p}_o(\mathbf{f}, \mathbf{f}) = \mathbf{f}$ and $\mathfrak{p}_o(\mathbf{t}, \mathbf{t}) = \mathbf{t}$.
3. Apply the semantics of E_1, i.e., f. We obtain $f(\mathbf{f}) = (\mathbf{f}, \mathbf{f})$ and $f(\mathbf{t}) = (\mathbf{f}, \mathbf{t})$.
4. Compute the greatest lower bound: $\mathrm{glb}(\{(\mathbf{f}, \mathbf{f}), (\mathbf{f}, \mathbf{t})\}) = (\mathbf{f}, \mathbf{t})$.

Hence, the semantics of $(\mathsf{E}_1\ \mathsf{E}_2)$ in $\mathcal{I}$ is $(\mathbf{f}, \mathbf{t})$.

Both the approach of [Charalambidis et al., 2018], perfected in Chapter 4, and the one of [Bogaerts et al., 2024] define a semantics for higher-order LP, but how do they relate to each other? The possible directions of investigation are multiple. The approximation spaces we considered in Chapter 4 are larger and more refined than those in [Bogaerts et al., 2024]. Is there a correspondence between certain elements? What do the additional elements contribute to? In which cases are they necessary? Do they provide more precise fixpoints? Do we even get the same fixpoints? If not, in which cases do we find discrepancies?

We have already explored some of these questions, although most of the research in this regard is left for future work. We provide below some preliminary ideas and conjectures.

Preliminary Considerations

We begin by comparing the approximation spaces defined in Section 5 and the ones defined in [Bogaerts et al., 2024]. We denote by $\mathrm{App}_{24}(E_\tau)$ the approximation space for the elements of type τ as characterized in [Bogaerts et al., 2024]. Since $\mathrm{App}_{24}(E_o) = \mathrm{App}(E_o)$, and the elements of type ι are not approximated, we are mainly interested in morphism types. Before diving deeper into this matter, we need to define a novel class of elements of the approximation spaces, which we call *tight* elements.

Definition 4.21

Let $E_\tau \in S_{\mathbb{K}}$. An element $x \in \text{App}(E_\tau)$ is *tight* if one of the following conditions holds:

1. $\tau \in \mathcal{B}_{\mathbb{K}}$ and there exists $e \in E_\tau$ such that $x = \text{glb}((\mathfrak{p}_\tau)^{-1}(e))$,
2. $\tau = \Pi_{i \in I} \tau_i$ and for each $i \in I$, the i-th component $\pi_i(x)$ of x is tight,
3. $\tau = \tau_1 \to \tau_2$ and $E_{\tau_1} \notin S_{\mathbb{K}}$,
4. $\tau = \tau_1 \to \tau_2$, $E_{\tau_1} \in S_{\mathbb{K}}$, and $x(a) = \text{glb}(\{x(b) \mid b \in \text{App}(E_{\tau_1}) \text{ s.t. } a \leq_p b\})$ for all $a \in \text{App}(E_{\tau_2})$ such that a is not tight.

We denote by $\mathcal{T}^\tau$ the set of tight elements of type τ.

Notice that this is a general definition that would work for any approximation system. If we restrict our attention to the setup of Section 5 of Chapter 4, the definition above simplifies: since there is only one base type whose semantics is in $S_{\mathbb{K}}$, namely the boolean type o, and for this type there is one exact representative per element of the semantics, Item 1 in Definition 4.21 becomes just "$\tau = o$ and $x \in \{(\mathbf{t}, \mathbf{t}), (\mathbf{f}, \mathbf{f})\}$".

Now, let $\tau := \tau_1 \to \tau_2$ be a morphism type. We make the following claims:

- If $f, g \in \mathcal{E}_\tau$, and they agree on tight elements, i.e., $f(x) = g(x)$ for all $x \in \mathcal{T}^{\tau_1}$, then f and g represent the same element, i.e., $\mathfrak{p}_\tau(f) = \mathfrak{p}_\tau(g) \in E_\tau$.
- There exists a surjective function $T^\tau : \text{App}(E_\tau) \to \text{App}_{24}(E_\tau)$, which we call *tightification function*.
- For each $f \in \text{App}_{24}(E_\tau)$, the preimage of f via T^τ contains one and only one tight element, and this is the most precise element of the preimage, i.e., $(T^\tau)^{-1}(f) \cap \mathcal{T}^\tau = \{\text{lub}((T^\tau)^{-1}(f))\}$.
- We can define an *inverse tightification* function $\mathbb{L}^\tau : \text{App}_{24}(E_\tau) \to \mathcal{T}^\tau \subseteq \text{App}(E_\tau)$ by sending $f \in \text{App}_{24}(E_\tau)$ to $\mathbb{L}^\tau(f) := \{\text{lub}((T^\tau)^{-1}(f))\}$. This function $\mathbb{L}^\tau$ is a right inverse of T^τ, i.e., $T^\tau \circ \mathbb{L}^\tau(f) = f$ for all $f \in \text{App}_{24}(E_\tau)$. Moreover, $\mathbb{L}^\tau \circ T^\tau(f) = f$ for all $f \in \mathcal{T}^\tau$. In particular, the restriction of T^τ to $\mathcal{T}^\tau$ is a bijection.
- Using T^τ, it is possible to define a surjective function T from the set of three-valued interpretations of our approach to the set of three-valued interpretations used in [Bogaerts et al., 2024]. Similarly, its right inverse $\mathbb{L}$ can be obtained.

Concerning the fixpoints obtained via the approach of [Charalambidis et al., 2018] or the one of [Bogaerts et al., 2024], we conjecture the following:

- There is bijection between the set of stable fixpoints of [Bogaerts et al., 2024] and the set of tight stable fixpoints of [Charalambidis et al., 2018]. Moreover, the exact stable fixpoints obtained through either of the approaches are exactly the same.
- The well-founded fixpoint obtained via the technique of [Charalambidis et al., 2018] is at most as precise as the image of the well-founded fixpoint of [Bogaerts et al., 2024] via $\mathcal{L}$.

It would be interesting to understand whether there exists a class of logic programs for which the well-founded fixpoint is guaranteed to be tight when computed with our framework, and for which programs (if any) the well-founded fixpoint obtained via [Charalambidis et al., 2018] is strictly less precise than the image of the well-founded fixpoint of [Bogaerts et al., 2024] via $\mathcal{L}$.

7 Conclusion

We introduced a novel theoretical framework that provides a mathematical foundation for using the machinery of AFT on higher-order objects. In particular, we defined *approximation categories* and *approximation systems*: they employ the notion of Cartesian closedness to inductively construct a hierarchy of approximation spaces for the semantics of each type of a given (higher-order) language. This approach solves the issue of applying approximate objects onto approximate objects and ensures that the approximation spaces have the same mathematical structure at any order of the hierarchy, enabling the application of the same AFT techniques at all levels. Moreover, we defined exact elements of a higher-order approximation space, together with a projection function. This is a non-trivial definition and it is fundamental to obtain a sensible AFT framework, i.e., a framework in which we can determine when an object, and in particular a model, is two-valued, and retrieve the elements that are being approximated.

Despite seeming complicated at first, the use of CT not only provides a solid, formal mathematical foundation to work with, but also allows to reduce the complexity of proofs. The inductive nature and generality of the definition of an approximation system make it extremely easy to adapt the framework to different languages, types, and semantics, as we only have to modify the base elements of the induction. Such generality enables extending different existing versions of AFT while capturing their common underlying characteristics, as we have shown for standard AFT and the extension of consistent AFT of Charalambidis et al. [2018]. Moreover, concerning the latter version of AFT mentioned, we have resolved its problematic features. In particular, we provided a novel approximator which behaves properly, even on programs with existential quantifiers in the body of rules, and we defined the concept of exactness, previously missing in the work of Charalambidis et al. [2018], allowing to consider exact stable models.

As far as future work and developments are concerned, it is paramount to notice that by systematically extending present (and possibly future) versions of AFT to the higher-order setting, this chapter further enriches the vast body of algebraic results on AFT. In particular, this allows us to make all the already-existing formal results regarding AFT readily available in a higher-order AFT context. This includes, but it is not limited to, stratification results [Vennekens et al., 2006; Bogaerts and Cruz-Filipe, 2021], grounded fixpoints [Bogaerts et al., 2015a], well-founded induction [Denecker and Vennekens, 2007], and strong equivalence [Truszczynski, 2006]. Moreover, in light of this newly established bridge to the higher-order environment, one could explore the possibilities within the application fields where AFT previously succeeded, such as abstract argumentation [Strass, 2013; Bogaerts, 2019], active integrity constraints [Bogaerts and Cruz-Filipe, 2018], stream reasoning [Antic, 2020], integrity constraints for the semantic web [Bogaerts and Jakubowski, 2021], and even DatalogMTL$^\neg$ [Pollaci, 2025], which we presented in Chapter 3.

Lastly, it may be of interest to research how the developed higher-order semantics and language $\mathcal{HOL}$ presented in Section 5 relate to Hilog and Prolog with meta-predicates [Chen et al., 1993]. In particular, $\mathcal{HOL}$ and Prolog show two rather different natures: while Prolog is procedural and intensional, the language $\mathcal{HOL}$ provides a declarative and extensional approach. This is indeed not trivial to obtain for the higher-order setting, as was also pointed out by Rondogiannis and Symeonidou [2018].

5

Only Knowing and Common Knowledge

The content of this chapter is based on the conference paper *Mathematical Foundations for Joining Only Knowing and Common Knowledge* [Cramer et al., 2023a], and its extended version [Cramer et al., 2023b], which includes the proofs of all the claims, as well as the additional intermediate results necessary for such proofs. My contributions lie mainly in the proofs and the final formalization of the concept of *biworld*. The original idea and the first, foundational stages of this project, however, were conceived during Pieter Van Hertum's PhD with his supervisor, Marc Dencker, and Marcos Cramer, prior to my arrival. Thus, although I am not listed as first author in the two aforementioned papers, my co-authors agree that Marcos Cramer and I should be considered joint first authors.

This work has been presented at the 20th International Conference on Principles of Knowledge Representation and Reasoning (KR) in Rhodes, Greece, and published in its proceedings. Section 6 of this Chapter is not part of the aforementioned publications

1 Introduction

When developing intelligent agents, it is important that they can reason not just about the world they are placed in, but also about the knowledge or beliefs of other agents in their environment. Consider for instance a traffic situation where two cars meet at a crossing. When the driver of the first car observes their light is green, the typical action to be taken would be to keep on driving. The reason for this is two-fold: on the one hand, the driver knows the traffic regulations and the fact that the other car should stop. But on the other hand, the driver also knows (or believes) that the other driver also knows the regulations (and will likely respect them). It is this knowledge about the other driver's knowledge that allows the first driver to conclude that it is safe to continue.

The formal study of knowledge and how to reason correctly about it has a long history in knowledge representation, dating back at least to the 1960s [Hintikka, 1962]. This study becomes particularly interesting when, as in our example above, multiple agents are involved. Next to the standard knowledge operator K, we are concerned with two epistemic operators, namely the **common-knowledge** and **only-knowing** ones, and their intricate interplay. We say that a statement φ is *common knowledge* among a group G of agents (and denote this $C_G\varphi$) if each agent in G knows φ, and also knows that every agent in G knows φ, and knows that everyone in G knows that everyone in G knows φ, and so on. This operator is useful, for instance, for reasoning about coordination or agreement between agents. We say that an agent $\mathcal{A}$ *only knows* a statement φ (and denote this $O_\mathcal{A}\varphi$) if the agent knows φ (denoted $K_\mathcal{A}\varphi$) and moreover *everything* they know follows from φ (so whenever $K_\mathcal{A}\psi$ holds, it must be the case that φ entails ψ). This operator is useful for instance when we consider that knowledge-based agents do not know anything except for what follows from their knowledge base and we might want to reason about their knowledge as well.

There have been many papers studying these operators in isolation [Fagin et al., 1995; Meyer and van der Hoek, 1995; Halpern and Lakemeyer, 2001; Waaler and Solhaug, 2005; Belle and Lakemeyer, 2015b], and some authors have even studied the combination of the two [Aucher and Belle, 2015; Belle and Lakemeyer, 2015a; Van Hertum, 2016]. However, we will argue that there are some essential properties of only knowing that none of these approaches captures. We will start with the good news: it is easy to develop a Kripke semantics for a logic with these two operators: given a Kripke structure $\mathcal{K}$ with set of worlds W and an accessibility relations $R_\mathcal{A}$ for every agent $\mathcal{A}$, the semantics for C_G and $O_\mathcal{A}$ is given by

- $\mathcal{K}, w \models C_G\varphi$ if $\mathcal{K}, w' \models \varphi$ for all w' reachable from w with edges in $\bigcup_{\mathcal{A}\in G} R_\mathcal{A}$, and
- $\mathcal{K}, w \models O_\mathcal{A}\varphi$ if for every world $w' \in W$, $\mathcal{K}, w' \models \varphi$ if and only if $(w, w') \in R_\mathcal{A}$.

Intuitively, the if-part in the definition of the semantics of $O_{\mathcal{A}}$ states that $\mathcal{A}$ knows φ (in all worlds $\mathcal{A}$ deems possible, φ holds), and the only-if-part ensures that the agent doesn't know anything else (all worlds in which φ holds are indeed deemed possible by the agent in question).

If this is so easy, then where's the catch, one might wonder. Well, the problem lies in the choice of the set W of worlds. The question we tackle is: can we construct a set of worlds W that is rich enough such that, for instance, $O_{\mathcal{A}}\top$ really means that that $\mathcal{A}$ only knows tautologies in the language (i.e., that agent $\mathcal{A}$ "knows nothing")? In other words, can we construct a canonical Kripke structure for this logic? A naive first attempt at doing so would be as follows.

Definition 5.1: World — incorrect definition

Given a propositional vocabulary Σ, a world w consists of

- a (classical propositional) interpretation w^{obj} over Σ (the *objective interpretation* of w), and
- for each agent $\mathcal{A} \in \boldsymbol{\mathcal{A}}$, a set of worlds $\mathcal{A}^w$.

However, the attentive reader might have noticed that this circular definition breaks the basic rules of set theory (a world is defined to consist of, among others, a set of worlds). We are not the first to observe this issue. The most common approach to alleviate it is to approximate the knowledge of agents up to a certain level and defining k+1-worlds as consisting, among others, of a set $\mathcal{A}^w$ of k-worlds for each agent $\mathcal{A}$. There are two challenges with this approach. The first is that as soon as we add common knowledge to our language, there is a strong need for these worlds to be *infinitely deep*. One way to achieve this is to consider an infinite *precision-increasing* sequence of k-worlds, as is done for instance by Fagin et al. [1991], or by Belle and Lakemeyer [2015a]. However, that in itself does not suffice: we show that to evaluate certain formulas there is a need to have worlds that are even deeper than this. The second is that in such approximations, no matter how deep one goes, there is never enough information to conclude that this is "all we know": it might always be that by making the approximation more precise, more knowledge at some later level comes in.

This brings us to the main contribution of this chapter: a solution to the above problem. First we will define a notion of μ-biworld, where μ can be any countable ordinal (thereby resolving the first challenge), and where the "bi" in biworld stands for the fact that we do not just approximate the set of worlds an agent deems possible, but also the set of worlds an agent deems *impossible*. Intuitively, with each μ+1-biworld w, we will associate a set $\mathcal{A}^w$ of μ-biworlds representing the set of μ-biworlds of which the agent deems some extension possible, and a set $\bar{\mathcal{A}}^w$ of μ-biworlds of which the agent deems

some extension impossible. This immediately allows us to see when all of agent $\mathcal{A}$'s knowledge is captured by such a biworld: this is precisely when $\mathcal{A}^w \cap \bar{\mathcal{A}}^w$ is empty. For limit ordinals, the situation is more complex, and this makes the construction of biworlds highly technical and mathematical. This transfinite construction of our biworlds is given in Section 2; Section 3 then shows several properties of them, essentially showing that they are well-behaved, in a precise sense.

In Section 4 we define our logic of common knowledge and only knowing as a simple multi-agent epistemic logic extended with the operators $O_{\mathcal{A}}$ and C_G as described above. We show that the biworlds possess enough information so that the formulas in our logic can be evaluated in them. More specifically, we define a three-valued model semantics for our logic: given a formula φ and a biworld w of any depth, we define what φ^w is. This can be true, false, or unknown, where the last case represents the fact that the biworld is not sufficiently "deep" to evaluate the formula. Moreover, we show that we do not need arbitrarily deep biworlds: if a biworld has depth at least $\omega^2 + 1$, then every formula will evaluate in it either to true or to false. Inspired by this observation, we are able to define our canonical Kripke structure: the *worlds*, are precisely the ω^2+1-biworlds that are *completed*, which is a technical term to denote the fact that it identifies for every other biworld of any other level whether or not the agent believes it is possible, i.e., that it characterizes complete knowledge. The accessibility relations can directly be obtained from the definition of the worlds. We then proceed to show that the semantics obtained from this canonical Kripke-structure actually coincides with the valuation we started from. Finally, this allows us to prove several desirable properties the resulting logic satisfies (see Theorem 5.5 for an extensive list), including the following two:

- For any φ and ψ, if $\varphi \nvDash \psi$, then $O_{\mathcal{A}}\varphi \vDash \neg K_{\mathcal{A}}\psi$.
- For any φ, $O_{\mathcal{A}}\varphi \nvDash \bot$.

The first property states precisely that whenever $O_{\mathcal{A}}\varphi$ holds, all formulas not entailed by φ are not known (and in fact we also have that all properties entailed by φ are known by $\mathcal{A}$). The second property states that for *any* formula φ, there is a world in which $O_{\mathcal{A}}\varphi$ holds. In fact, our results are even stronger than this: we show that there is a unique state-of-mind of agent $\mathcal{A}$ in which they know precisely φ. These two properties of the $O_{\mathcal{A}}$ operator, while quite simple, are — to the best of our knowledge — not satisfied by any other paper combining common knowledge and only knowing.

While developing our worlds, and our logic, we do not enforce any properties that are often associated to *knowledge*. For instance, our worlds do not guarantee that our agents are *truthful* or *introspective*. The main focus of this chapter is on how to create a semantic structure that allows defining a rich enough set of worlds. However, once this semantic structure is in place, it is possible to use it to define a logic that satisfies such properties as well. To illustrate this, we show in Section 5 how we can hard-code into the logic the fact

that agents are positively introspective, which in fact means that it is common knowledge that all agents satisfy this property. We also discuss the possibility of adding truthfulness and negative introspection, highlighting a problem that arises from combining them with only knowing. We use this problem to uncover a major mistake in previous work on combining only knowing and common knowledge.

In Section Section 6, we present a special form knowledge in which only knowing and common knowledge are tightly entwined. While very challenging to precisely represent, this type of knowledge is often implicitly assumed in several common epistemic puzzles. We believe the framework built in this chapter may be able to capture this form of knowledge of the agents, but this would require further investigation. Finally, in Section 7 we extensively discuss related work and explain why previous approaches could not achieve the two properties set out above. We conclude in Section 8.

2 Construction of Biworlds

In this section, we define the concept of biworlds. Intuitively, a biworld w consists of *(1)* an objective interpretation, and *(2)* for each agent $\mathcal{A}$ two sets of biworlds $\mathcal{A}^w$ and $\bar{\mathcal{A}}^w$, where the former set contains all the biworlds the agent deems possible, and the latter the biworlds the agent deems impossible. As explained before, using this definition brings us into set-theoretic problems. To resolve this issue, we define the notion of μ-biworlds, for all countable ordinals μ. Intuitively, a μ-biworld does not describe the complete epistemic state of the different agents, but only their belief about the world up to a certain depth. For some formulas, having a certain depth will suffice to evaluate the formula. For instance, a 1-biworld will suffice to determine whether or not an agent knows p, but might not suffice to determine whether an agent knows that some other agent knows p. Moreover, we will later show that for our logic, the ordinal $\omega^2 + 1$ suffices, in the sense that every set of formulas is either true or false in every ω^2+1-biworld. While we only need the theory of biworlds up to $\omega^2 + 1$, we will still develop the theory in more generality for all *countable* ordinals. From now on, when we say *ordinal*, we mean *countable ordinal*. We repeat this assumption only in the proofs in which we require it.

If μ is a successor ordinal $\alpha + 1$, then a μ-biworld w associates with each agent two sets of α-biworlds, where $\mathcal{A}^w$ represents the set of α-biworlds that can be extended (in depth; for some sufficiently large ordinal) to a biworld the agent deems possible, and $\bar{\mathcal{A}}^w$ represents the set of α-biworlds that can be extended to a biworld the agent deems impossible. Clearly, a natural condition will then be that the union of these two sets is the set of all α-biworlds.

If there is any biworld in the intersection of $\mathcal{A}^w$ and $\bar{\mathcal{A}}^w$, this biworld must be extendible both to a biworld the agent deems possible, and to a biworld the agent deems impossible. If for all agents $\mathcal{A}$, the intersection of $\mathcal{A}^w$ and $\bar{\mathcal{A}}^w$ is empty, this means that the information

in w fully specifies which biworlds the agents deem possible and which ones impossible. Once this is fully specified, there is only one way to extend w to a biworld of a greater depth. When there are multiple ways in which a μ-biworld w can be extended to a μ+1-biworld, then we call w an *incompleted* biworld.

Since a biworld in the intersection of $\mathcal{A}^w$ and $\bar{\mathcal{A}}^w$ must be extendible (for some sufficiently large depth) both to a biworld the agent deems possible and to a biworld the agent deems impossible, it must be incompleted.

We start with the auxiliary notion of a μ-prebiworld, which approximates the more complex notion of a μ-biworld while leaving out some of the structural conditions that we impose on μ-biworlds. We define μ-prebiworlds, restrictions of prebiworlds and a precision order on the prebiworlds through **simultaneous recursion**, but for readability, we split it into Definitions 5.2, 5.3 and 5.4.

Definition 5.2: μ-Prebiworld

Let μ be an ordinal. We define the set of *μ-prebiworlds* over a propositional vocabulary Σ, and a set of agents $\boldsymbol{\mathcal{A}}$ by transfinite induction:

- A 0-prebiworld w is an interpretation I over Σ. We define $w^{obj} = I$ and $d(w) = 0$.
- A μ+1-prebiworld w is a triple $(I, (\mathcal{A}^w)_{\mathcal{A}\in\boldsymbol{\mathcal{A}}}, (\bar{\mathcal{A}}^w)_{\mathcal{A}\in\boldsymbol{\mathcal{A}}})$, where I is an interpretation over Σ, and for each $\mathcal{A} \in \boldsymbol{\mathcal{A}}$, $\mathcal{A}^w$ and $\bar{\mathcal{A}}^w$ are sets of μ-prebiworlds. We define $d(w) = \mu + 1$ and $w^{obj} = I$.
- A λ-prebiworld w for a limit ordinal λ is a precision-increasing transfinite sequence $(w_\alpha)_{\alpha<\lambda}$ of prebiworlds, i.e. for each $\alpha < \lambda$, w_α is an α-prebiworld and for each $\beta \leq \alpha$, $w_\beta \leq_{\mathcal{W}} w_\alpha$. We define $d(w) = \lambda$ and $w^{obj} = w_0$.

For each prebiworld w, we call w^{obj} the *objective world* of w. For each ordinal μ, we denote by $\mathcal{W}_p^\mu$ the set of μ-prebiworlds, and we call the integer $d(w)$ the *depth of* w. If $w \in \mathcal{W}_p^\lambda$ with λ a limit ordinal, then for each $\alpha < \lambda$, we denote by $(w)_\alpha$ the α-prebiworld which is the element in position α in the transfinite sequence represented by w.

The following definition captures what it means to restrict a prebiworld of depth μ to a prebiworld of smaller depth α.

Definition 5.3: Restriction of a prebiworld

Assume w is a μ-prebiworld and $\alpha \leq \mu$. The *restriction* of w to α is the α-prebiworld $w|_\alpha$ defined as follows:

- If $\alpha = 0$, then $w|_\alpha = w^{obj}$.
- If α is a limit ordinal, then $w|_\alpha = (w|_\beta)_{\beta<\alpha}$
- If $\alpha = \alpha' + 1$, we distinguish two cases:
  - If μ is a limit ordinal, then $w|_\alpha = (w)_\alpha$.
  - If μ is a successor ordinal $\mu = \mu' + 1$, then $w|_\alpha^{obj} = w^{obj}$, $\mathcal{A}^{w|_\alpha} = \{w'|_{\alpha'} \mid w' \in \mathcal{A}^w\}$, and $\bar{\mathcal{A}}^{w|_\alpha} = \{w'|_{\alpha'} \mid w' \in \bar{\mathcal{A}}^w\}$.

The precision order[1] on the prebiworlds is based on the notion of restriction:

Definition 5.4: Precision order

If w is a μ-prebiworld and v an α-prebiworld with $\alpha \leq \mu$, we say that v is less precise than w (and denote this $v \leq_{\mathcal{W}} w$) if $w|_\alpha = v$.

If $w \leq_{\mathcal{W}} v$ for some prebiworlds w and v, we call w a *restriction of* v and v an *extention of* w.

Now we define the set of incompleted prebiworlds and the set of biworlds by simultaneous transfinite recursion. For better readability, we separate this simultaneous definition into Definitions 5.5 and 5.6 and explain afterwards why it is a successful definition.

Definition 5.5: Incompleted prebiworld

A μ-prebiworld w is *incompleted* if there exists two distinct μ+1-biworlds v_1 and v_2 such that $v_1, v_2 \geq_{\mathcal{W}} w$.

A prebiworld is called *completed* if it is not incompleted.

[1] Notice that this precision order is different than the one we define on the approximation spaces in the context of AFT (hence, the different notation).

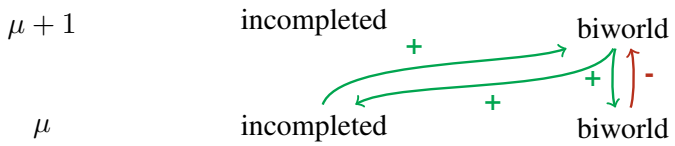


Figure 5.1: Dependencies between the notions of *incompleted prebiworlds* and *biworlds* at levels μ and $\mu + 1$

Definition 5.6: μ-Biworld

A *μ-biworld* is a μ-prebiworld w such that one of the following conditions holds:

(a) $\mu = 0$;

(b) $\mu = \mu' + 1$ is a successor ordinal, and for each agent $\mathcal{A} \in \boldsymbol{\mathcal{A}}$ the following hold:

  (1) the union $\mathcal{A}^w \cup \bar{\mathcal{A}}^w$ is the set of all μ'-biworlds;

  (2) for each $v \in \mathcal{A}^w \cap \bar{\mathcal{A}}^w$, v is incompleted;

(c) μ is a limit ordinal, and $(w)_\alpha$ is an α-biworld for each $\alpha < \mu$.

For each ordinal μ, the set of μ-biworlds is denoted as $\mathcal{W}^\mu$. We often call a 0-biworld an *objective world*.

Since the definition of *incompleted prebiworlds* of depth μ depends on the existence of biworlds of depth $\mu + 1$, an attentive reader may be worried whether the simultaneous definition of *incompleted prebiworlds* and *biworlds* is really a successful definition. We therefore now explain how this definition is to be understood.

For this, Figure 5.1 depicts the dependencies between the notions of *incompleted prebiworlds* and *biworlds* at levels μ and $\mu+1$. Here, a green arrow from one notion to another indicates that the second notion positively depends on the first, i.e., will include more objects when the first notion includes more objects. A red arrow, on the other hand, indicates a negative dependency, so that the second notion will include fewer objects when the first notion includes more objects. For example, the green arrow from *biworld* at level $\mu + 1$ to *incompleted* at level μ indicates that determining additional biworlds at level $\mu + 1$ can lead to determining a prebiworld w at level μ to be incompleted, as one of the newly determined biworlds at level $\mu + 1$ may imply there is more than one way of extending w to a μ+1-biworld. The arrow from *biworld* at level μ to *biworld* at level $\mu + 1$ is red,

because given a μ+1-prebiworld w, determining more biworlds at level μ can show that the union $\mathcal{A}^w \cup \bar{\mathcal{A}}^w$ is not the set of all μ-biworlds, i.e., that w is not a biworld.

Now the simultaneous definition of *incompleted prebiworlds* and *biworlds* is a transfinite recursion over μ, where at each level μ of the induction, the notions of an *incompleted μ-prebiworld* and a μ+1-*biworld* are defined simultaneously by choosing the minimal set of incompleted μ-prebiworlds and μ+1-biworlds that satisfy Definitions 5.5 and 5.6(b). Such a minimal set exists, because the definitions of *incompleted μ-prebiworld* and μ+1-*biworld* only depend positively on each other. Here the negative dependency of μ+1-*biworld* on μ-*biworld* is not a problem, because at this stage in the transfinite recursion over μ, the set of μ-biworlds has already been determined: If μ is a successor ordinal $\mu' + 1$, it has been determined in the previous step μ' of the transfinite recursion. If μ is a limit ordinal, it has been determined by Definition 5.6(c) and the fact that for every $\alpha < \mu$, the set of α-biworlds has already been determined. If $\mu = 0$, the set of μ-biworlds just coincides with the set of μ-prebiworlds by Definition 5.6(a).

Example 5.1

We consider one of the simplest settings, namely we suppose to have just one agent $\mathcal{A}$, and one propositional variable p in the vocabulary Σ. The set $\mathcal{W}^0$ of 0-biworlds coincides with the set of all 0-prebiworlds by Definition 5.6, and it is equal to $\mathcal{W}^0 = \{\{p\}, \{\emptyset\}\}$. The set $\mathcal{W}^1$ already counts considerably more elements, eighteen to be precise: nine with $\{p\}$ as objective world, and other nine with $\{\emptyset\}$. This follows from the limitations imposed by Item b1 in Definition 5.6 of biworlds[a]. For the sake of conciseness, we omit the explicit description of all the 1-biworlds but one:

$$v_1 := (\{p\}, \{\{p\}\}, \{\{p\}, \{\emptyset\}\}).$$

Since $\{p\}$ is the objective world of v_1, p is true in v_1. Moreover, the sets $\mathcal{A}^{v_1}$ and $\bar{\mathcal{A}}^{v_1}$ provide some information on the beliefs of $\mathcal{A}$ in v_1. Recall that $\mathcal{A}^{v_1}$ (resp. $\bar{\mathcal{A}}^{v_1}$) is the set of biworlds that have an extension $\mathcal{A}$ deems possible (resp. impossible). Since $\{p\}$ is the only biworld in $\mathcal{A}^{v_1}$, p is true in any biworld $\mathcal{A}$ deems possible. As we will see later, this means that $\mathcal{A}$ knows p. On the contrary, since $\bar{\mathcal{A}}^{v_1}$ contains both $\{p\}$ and $\{\emptyset\}$, p is true in some of the biworlds $\mathcal{A}$ deems impossible, and false in others.

Starting with the objective world $v_0 := \{p\}$ and continuing with the biworld v_1, we can inductively build an ω-prebiworld $v := (v_\alpha)_{\alpha<\omega}$ as follows

$$v_\alpha := \begin{cases} \{p\} & \text{if } \alpha = 0 \\ (v_0, \{v_{\alpha'}\}, \mathcal{W}^{\alpha'}) & \text{if } \alpha = \alpha' + 1. \end{cases}$$

Since for all successor ordinals $\alpha = \alpha' + 1 < \omega$ we have $\mathcal{A}^{v_\alpha} \cap \bar{\mathcal{A}}^{v_\alpha} = \{v_{\alpha'}\}$, in order to prove that v is an ω-biworld, it suffices to show that for all $\alpha < \omega$, v_α is incompleted. We achieve this with a proof by induction:

- $\alpha = 0$. The 1-prebiworlds

$$u_0^1 := (\{p\}, \{\{p\}\}, \{\{\emptyset\}\})$$
$$u_0^2 := (\{p\}, \{\{\emptyset\}\}, \{\{p\}\})$$

  are clearly distinct 1-biworlds extending v_0. Hence v_0 is incompleted.
- $\alpha = \alpha' + 1$. By induction hypothesis, $v_{\alpha'}$ is incompleted, i.e., it has two distinct α-biworld extensions $u_{\alpha'}^1$ and $u_{\alpha'}^2$. It is easy to see that

$$u_\alpha^1 := (\{p\}, \{u_{\alpha'}^1\}, \mathcal{W}^\alpha \setminus \{u_{\alpha'}^1\})$$
$$u_\alpha^2 := (\{p\}, \{u_{\alpha'}^2\}, \mathcal{W}^\alpha \setminus \{u_{\alpha'}^2\})$$

  are two distinct α+1-biworlds extending v_α. Hence, v_a is incompleted as desired.

We conclude that v is a ω-biworld. Moreover, v is completed. Let us explain why this holds.

Taking into account the second item of Definition 5.3 and the incompletedness of each v_α, one can easily see that v has exactly two ω+1-prebiworlds extending it, namely:

$$u^1 := (\{p\}, \{v\}, \mathcal{W}^\omega \setminus \{v\}), \quad u^2 := (\{p\}, \{v\}, \mathcal{W}^\omega).$$

Since $\mathcal{A}^{u^1} \cup \bar{\mathcal{A}}^{u^1} = \mathcal{W}^\omega$ and $\mathcal{A}^{u^1} \cap \bar{\mathcal{A}}^{u^1} = \emptyset$, u^1 is an ω+1-biworld. However, u^2 is a biworld only if v is incompleted, by Item b of Definition 5.6. Hence, u^2 is a biworld if and only if v is incompleted. Since we construct the concepts of incompleted ω-prebiworld and ω+1-biworld by simultaneous induction by choosing the minimal sets, u^2 is not a biworld and v is not incompleted.

[a] In the single-agent setting, $|\mathcal{W}^1| = n3^n$, where $n = |\mathcal{W}^0|$.

We now dive deeper in certain features of (pre)biworlds with limit ordinal depth. Note that for successor ordinals μ, and for any μ-prebiworld, each agent is equipped with two sets of prebiworlds. For limit ordinals μ, however, this is not the case. The following definition aims at retrieving a similar concept for a prebiworld with limit ordinal depth.

Definition 5.7

Given a limit ordinal λ, a λ-prebiworld w and an agent $\mathcal{A} \in \boldsymbol{\mathcal{A}}$, we define the following sets

$$\mathcal{A}{\uparrow}w := \{v \in \mathcal{W}_p^\lambda \mid \forall \mu < \lambda : (v)_\mu \in \mathcal{A}^{(w)_{\mu+1}}\}$$
$$\bar{\mathcal{A}}{\uparrow}w := \{v \in \mathcal{W}_p^\lambda \mid \forall \mu < \lambda : (v)_\mu \in \bar{\mathcal{A}}^{(w)_{\mu+1}}\}.$$

It is clear that if w in the definition above is a biworld, then $\mathcal{A}{\uparrow}w$ and $\bar{\mathcal{A}}{\uparrow}w$ are sets of biworlds. It is important to notice that the sets defined in Definition 5.7 do not carry the exact same meaning as the successor-ordinal counterpart: if $v \in \mathcal{A}{\uparrow}w$ for some λ-(pre)biworld w, then in w, for each approximation v_α ($\alpha < \lambda$) of v, the agent $\mathcal{A}$ deems some extension of v_α as possible. Analogously, if $v \in \bar{\mathcal{A}}{\uparrow}w$, then in w, for each approximation v_α ($\alpha < \lambda$) of v, the agent $\mathcal{A}$ believes some extension of v_α is impossible.

3 Properties of Biworlds

In this section, we show that the formal definitions stated in Section 2 behave well and that they indeed correspond to the intuitive ideas introduced above. First, we present two propositions regarding the notion of restriction, which show that the induced relation $\leq_{\mathcal{W}}$ is a non-strict partial order. Second, we provide some additional insight into the sets defined in Definition 5.7. Finally, we focus on certain properties concerning biworlds. In particular, we will show the following fundamental facts in Theorem 5.1:

- **Restrictability:** the restriction of a biworld is a biworld.
- **Monotonicity of completedness:** an extension of a completed biworld is a completed biworld.
- **Completability:** all biworlds have a a completed extension.
- **Completedness at successor ordinals:** a μ+1-biworld w is completed if and only if $\mathcal{A}^w \cap \bar{\mathcal{A}}^w = \emptyset$ for all agents $\mathcal{A}$.

Given Definitions 5.3 and 5.5, the first two properties are sensible to have. Completability will be fundamental in Section 4 and it is clearly desirable: a complete biworld characterizes complete knowledge, providing a full description of the epistemic state of the agents. The last property provides a simple characterization of what it means to be completed for biworlds of successor ordinal depth. While the four properties listed above may

seem natural and straightforward, several intermediate results are required to prove them to hold.

As anticipated, we start by showing that Definition 5.4 defines a non-strict partial order on the set of prebiworlds.

Proposition 5.1

Let w be a μ-prebiworld and let $\beta \leq \alpha \leq \mu$. Then $w|_\mu = w$ and $w|_\alpha|_\beta = w|_\beta$.

Proof. We first prove $w|_\mu = w$ by induction on μ.

1. $\mu = 0$. Then $w|_\mu = w^{obj} = w$ by Definitions 5.3 and 5.2.
2. $\mu = \mu' + 1$ is a successor ordinal. By induction hypothesis, we have that $w'|_{\mu'} = w'$ for each μ'-prebiworld w'. By Definition 5.3, we conclude this case.
3. μ is a limit ordinal. Then $w|_\mu = (w|_\alpha)_{\alpha<\mu}$, and we need to show $w|_\alpha = (w)_\alpha$. We proceed by induction on α.
   (a) If $\alpha = 0$ or α is a successor ordinal, then we conclude by Definition 5.3.
   (b) If α is a limit ordinal, then $w|_\alpha = (w|_\beta)_{\beta<\alpha} = ((w)_\beta)_{\beta<\alpha} = (w)_\alpha$, where the equalities hold by Definition 5.3, induction hypothesis on α, and Definition 5.2, respectively.

Now we show that $w|_\alpha|_\beta = w|_\beta$ by induction on μ.

1. $\mu = \alpha$. This is a consequence of the already proven first claim of this proposition.
2. $\mu = \mu' + 1$ is a successor ordinal and $\alpha < \mu$. We proceed by induction on β.
   (a) $\beta = 0$. Then $w|_\beta = w^{obj}$, $w|_\alpha|_\beta = w|_\alpha^{obj}$, and it is easy to see that $w|_\alpha^{obj} = w^{obj}$.
   (b) $\beta = \beta' + 1$ is a successor ordinal. Then $w|_\alpha|_\beta^{obj} = w|_\alpha^{obj} = w^{obj} = w|_\beta^{obj}$. Moreover, if $\alpha = \alpha' + 1$ is a successor ordinal, then for each agent $\mathcal{A} \in \boldsymbol{\mathcal{A}}$

$$\begin{aligned}\mathcal{A}^{w|_\alpha|_\beta} &= \left\{ w'|_{\beta'} \mid w' \in \mathcal{A}^{w|_\alpha} \right\} = \left\{ w'|_{\alpha'}|_{\beta'} \mid w' \in \mathcal{A}^{w} \right\} \\ &= \left\{ w'|_{\beta'} \mid w' \in \mathcal{A}^{w} \right\} = \mathcal{A}^{w|_\beta},\end{aligned}$$

   where the third equality holds by induction hypothesis in μ. Analogously, we have $\bar{\mathcal{A}}^{w|_\alpha|_\beta} = \bar{\mathcal{A}}^{w|_\beta}$. If α is a limit ordinal, then $w|_\alpha|_\beta = \left(w|_\gamma\right)_{\gamma<\alpha}|_\beta = w|_\beta$.
   (c) β is a limit ordinal. Then $w|_\alpha|_\beta = \left(w|_\alpha|_\gamma\right)_{\gamma<\beta} = \left(w|_\gamma\right)_{\gamma<\beta} = w|_\beta$, where the second equality holds by the induction hypothesis on β.

3. μ is a limit ordinal. We can proceed by nested inductions on β and α, as we did above. The proof is exactly the same as for μ being a successor ordinal, except for the case where both β and α are successor ordinals. In this case we have $w|_\alpha|_\beta = (w)_\alpha|_\beta = (w)_\beta = w|_\beta$, where the second equality holds because $(w)_\beta \leq_{\mathcal{W}} (w)_\alpha$.

□

Proposition 5.2

The relation $\leq_{\mathcal{W}}$ is a non-strict partial order (a reflexive, antisymmetric and transitive relation). The induced strict order $<_p$ is a well-founded order.

Before proceeding to the properties of biworlds, we focus on the sets of Definition 5.7. Analogously to what happens in Item b of Definition 5.6 for biworlds of successor ordinal depth, the union of $\mathcal{A}{\uparrow}w$ and $\bar{\mathcal{A}}{\uparrow}w$ is the whole set of biworlds of that depth.

Proposition 5.3

Let λ be a limit ordinal, and w be a λ-biworld. For all $\mathcal{A} \in \boldsymbol{\mathcal{A}}$, we have $\mathcal{A}{\uparrow}w \cup \bar{\mathcal{A}}{\uparrow}w = \mathcal{W}^\lambda$.

Proof. Let $\mathcal{A} \in \boldsymbol{\mathcal{A}}$. Clearly, $\mathcal{A}{\uparrow}w \cup \bar{\mathcal{A}}{\uparrow}w \subseteq \mathcal{W}^\lambda$. To show the other inclusion, let $w' \in \mathcal{W}^\lambda \setminus \bar{\mathcal{A}} \uparrow w$. Then there exists $\mu' < \lambda$ such that $(w')_{\mu'} \notin \bar{\mathcal{A}}^{(w)_{\mu'+1}}$. Since w and w' are biworlds, we must have $(w')_{\mu'} \in \mathcal{A}^{(w)_{\mu'+1}}$. By the precision order of limit biworlds, for all $\mu < \mu'$, we get $(w')_\mu \in \mathcal{A}^{(w)_{\mu+1}}$, and for all $\mu > \mu'$ we get $(w')_\mu \notin \bar{\mathcal{A}}^{(w)_{\mu+1}}$. In particular, this implies $(w')_\mu \in \mathcal{A}^{(w)_{\mu+1}}$ for all $\mu > \mu'$. Hence, $w' \in \mathcal{A}{\uparrow}w$ as desired. □

It is interesting to notice that the second part of Item b of Definition 5.6 does not hold in general for the sets $\mathcal{A}{\uparrow}w$ and $\bar{\mathcal{A}}{\uparrow}w$, i.e., their intersection might contain completed biworlds (see Example 5.2). Nevertheless, this intersection tells us something about the completedness of w, as stated in the following proposition.

Proposition 5.4

Let λ be a limit ordinal and w a λ-biworld. If $\mathcal{A}{\uparrow}w \cap \bar{\mathcal{A}}{\uparrow}w = \emptyset$ for all $\mathcal{A} \in \boldsymbol{\mathcal{A}}$, then w is completed.

Proof. Suppose $w' \geq_{\mathcal{W}} w$ is a $\lambda+1$-biworld. Then $w^{obj} = (w)_0$. Furthermore,for each $\alpha < \lambda$ and for each $\mathcal{A} \in \boldsymbol{\mathcal{A}}$, we have $\mathcal{A}^{w_{\alpha+1}} = \mathcal{A}^{w'|_{\alpha+1}}$ and $\bar{\mathcal{A}}^{w_{\alpha+1}} = \bar{\mathcal{A}}^{w'|_{\alpha+1}}$. Hence, for each $\mathcal{A} \in \boldsymbol{\mathcal{A}}$, $\mathcal{A}^{w'} \subseteq \mathcal{A}{\uparrow}w$ and $\bar{\mathcal{A}}^{w'} \subseteq \bar{\mathcal{A}}{\uparrow}w$. But since $\mathcal{A}{\uparrow}w \cap \bar{\mathcal{A}}{\uparrow}w = \emptyset$ and $\mathcal{A}^{w'} \cup \bar{\mathcal{A}}^{w'} = \mathcal{W}^\lambda$,

the only way this can hold is when $\mathcal{A}^{w'} = \mathcal{A}{\uparrow}w$ and $\bar{\mathcal{A}}^{w'} = \bar{\mathcal{A}}{\uparrow}w$. This fully determines w', so there cannot be two distinct $\lambda+1$-biworlds $w', w'' \geq_{\mathcal{W}} w$. □

Example 5.2

Consider the completed ω-biworld v defined in Example 5.1. It is easy to see that $\mathcal{A}{\uparrow}v = \{v\}$ and $\bar{\mathcal{A}}{\uparrow}v = \mathcal{W}^{\omega}$. Hence, we get $\mathcal{A}{\uparrow}v \cup \bar{\mathcal{A}}{\uparrow}v = \mathcal{W}^{\omega}$ and $\mathcal{A}{\uparrow}v \cap \bar{\mathcal{A}}{\uparrow}v = \{v\}$, which agrees with Proposition 5.3 and shows that the converse of Proposition 5.4 does not hold. Moreover, notice that $\mathcal{A}{\uparrow}v \cap \bar{\mathcal{A}}{\uparrow}v$ contains a completed biworld.

We present one last result regarding the sets defined in Definition 5.7, that turns out to be fundamental to prove the lemmas leading to Theorem 5.1.

Proposition 5.5

Let λ be a limit ordinal, w be a λ-biworld, $\mathcal{A} \in \boldsymbol{\mathcal{A}}$ be an agent, and $\mu < \lambda$ be an ordinal. If $v \in \mathcal{A}^{(w)_{\mu+1}}$, then there exists $v' \in \mathcal{A}{\uparrow}w$ such that $v' \geq_{\mathcal{W}} v$. If $v \in \bar{\mathcal{A}}^{(w)_{\mu+1}}$, then there exists $v' \in \bar{\mathcal{A}}{\uparrow}w$ such that $v' \geq_{\mathcal{W}} v$.

Proof. Recall that we only consider countable ordinals. We prove the part of the statement with $v \in \mathcal{A}^{(w)_{\mu+1}}$, and the second part for $v \in \bar{\mathcal{A}}^{(w)_{\mu+1}}$ is analogous. By the precision order of limit biworlds, for each ordered pair (β, γ) of ordinals with $\beta < \gamma < \lambda$ and for each $u \in \mathcal{A}^{(w)_{\beta+1}}$, there exists $u' \in \mathcal{A}^{(w)_{\gamma+1}}$ such that $u' \geq_{\mathcal{W}} u$. So by the axiom of choice, there exists a family of functions $f^{\gamma}_{\mathcal{A}} : W^{\gamma}_{\mathcal{A}} \to \mathcal{A}^{(w)_{\gamma+1}}$ for $\gamma < \lambda$, where $W^{\gamma}_{\mathcal{A}} := \bigcup_{\beta<\gamma} \mathcal{A}^{(w)_{\beta+1}}$, such that for each $u \in W^{\gamma}_{\mathcal{A}}$, we have $f^{\gamma}_{\mathcal{A}}(u) \geq_{\mathcal{W}} u$. Since λ is a countable limit ordinal, we can choose a strictly increasing sequence $(\rho_n)_{n<\omega}$ of ordinals which has limit λ. W.l.o.g. we assume $\rho_0 > \mu$. Then, we define $\tilde{v} \in \mathcal{W}^{\alpha}$ as follows:

$$\tilde{v}_{\beta} := \begin{cases} v|_{\beta} & \text{if } \beta \leq \mu \\ f^{\rho_0}_{\mathcal{A}}(v)|_{\beta} & \text{if } \mu < \beta \leq \rho_0 \\ f^{\rho_{n+1}}_{\mathcal{A}}(\tilde{v}_{\rho_n})|_{\beta} & \text{if } \rho_n < \beta \leq \rho_{n+1} \end{cases} . \tag{5.1}$$

By the definition of the functions $f^{\rho_i}_{\mathcal{A}}$ and of restriction of biworlds, it is clear that $\tilde{v} \in \mathcal{A}{\uparrow}w$ and $\tilde{v} \geq_{\mathcal{W}} v$. □

The remaining definitions capture the four properties listed at the beginning of this section, but only at limited depth. They are used to inductively prove Lemma 5.7.

The first property is restrictability, i.e., whenever we take the restriction of a biworld, the result is also a biworld. A priori, it is only known that this is a prebiworld.

Definition 5.8

Let μ be an ordinal. We say that *restrictability holds at* μ (R_μ) if for every μ-biworld w, and every $\alpha \leq \mu$, $w|_\alpha$ is again a biworld.

The second property is monotonicity of completedness, which states that any extension of a completed biworld is itself completed. Intuitively, we expect this to hold: if one biworld has complete knowledge, refining it can only result in structures that still have complete knowledge.

Definition 5.9

Let μ be an ordinal. We say that *monotonicity of completedness holds at* μ (MoC_μ) if for each μ-biworld w and each ordinal $\alpha \leq \mu$, whenever $w|_\alpha$ is completed, so is w.

The third useful property is completability. It states that any biworld can always be extended to a completed one. This is in line with or view of incompleted biworlds as approixmations of some completed biworlds.

Definition 5.10

Let μ be an ordinal. We say that *completability holds at* μ (C_μ) if for every $\alpha < \mu$, and every α-biworld w, there exists a completed μ-biworld w' with $w' \geq_w w$.

The fourth and last property is a characterization of completedness for a biworld whose depth is a successor ordinal.

Definition 5.11

Let μ be a successor ordinal. We say that *the completedness condition holds at* μ (CC_μ) if for each μ-biworld w, w is completed if and only if for each agent $\mathcal{A}$, $\mathcal{A}^w \cap \bar{\mathcal{A}}^w = \emptyset$.

Definition 5.12

We say that biworlds are *well-behaved until* μ if restrictability, completability and monotonicity of completedness hold at all ordinals smaller than μ and the completedness condition holds at all successor ordinals smaller than $pl(\mu)$, where $pl(\mu)$ denotes the largest limit ordinal smaller or equal to μ.

In order to prove that every biworld enjoys the four properties listed at the beginning of this section, we need to prove Lemma 5.7 below. The proof of it requires several intermediate technical lemmas, whose interplay is sketched in Figure 5.2.

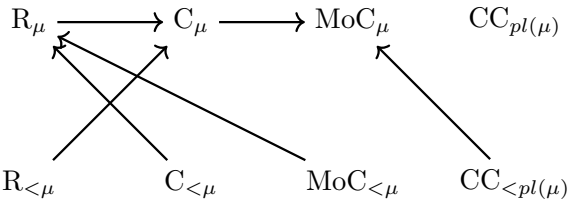


Figure 5.2: The figure depicts the relations expressed in the Lemmas 5.3, 5.5, 5.4, and 5.6. By $R_{<\mu}$, we denote that R_α holds for all $\alpha < \mu$; analogously for $C_{<\mu}$, and $MoC_{<\mu}$. By $CC_{<pl(\mu)}$ we denote that CC_α holds for all successor ordinals $\alpha < pl(\mu)$. A concept is true if the bodies of all the arrows pointing at it are true. For example: if $C_{<\mu}$ and $MoC_{<\mu}$, then R_μ. For the limit ordinal case, we need an additional implication not represented in this figure for the sake of simplicity, namely $C_{\mu+2} \implies CC_{\mu+1}$ (Lemma 5.6).

Lemma 5.1

Let w be a μ+1-biworld. If $\mathcal{A}^w \cap \bar{\mathcal{A}}^w = \emptyset$ for all $\mathcal{A} \in \boldsymbol{\mathcal{A}}$, then w is completed.

Proof. By contradiction, suppose there exist μ+2-biworlds v_1 and v_2, such that $v_1 \neq v_2$ and $v_1, v_2 \geq_{\mathcal{W}} w$. Then, there exists $\mathcal{A} \in \boldsymbol{\mathcal{A}}$ such that either $\mathcal{A}^{v_1} \neq \mathcal{A}^{v_2}$ or $\bar{\mathcal{A}}^{v_1} \neq \bar{\mathcal{A}}^{v_2}$. W.l.o.g., we can assume that $\mathcal{A}^{v_1} \neq \mathcal{A}^{v_2}$ and that there exists $u \in \mathcal{A}^{v_1}$ such that $u \notin \mathcal{A}^{v_2}$. Since v_2 and u are biworlds, u must be in $\bar{\mathcal{A}}^{v_2}$. Then, by definition of restriction, we have that $u|_\mu \in \mathcal{A}^{v_1|_{\mu+1}} \cap \bar{\mathcal{A}}^{v_2|_{\mu+1}} = \mathcal{A}^w \cap \bar{\mathcal{A}}^w = \emptyset$, a contradiction. □

Lemma 5.2

Suppose monotonicity of completedness holds at all ordinals up to and including μ. Suppose $\alpha \leq \mu$ and w is a completed α-biworld. Then there is at most one μ-biworld $w' \geq_{\mathcal{W}} w$. Moreover, if there is a μ-biworld $w' \geq_{\mathcal{W}} w$, w' is completed.

Proof. Fix α. We prove this lemma by induction over μ.

- $\mu = \alpha$. Trivial.
- $\mu = \mu' + 1$ is a successor ordinal greater than α. The induction hypothesis is that there is at most one μ'-biworld $v \geq_{\mathcal{W}} w$, and any such v is completed. Suppose for a contradiction that there are two distinct μ-biworlds $w', w'' \geq_{\mathcal{W}} w$. Then $w'|_{\mu'}$ and $w''|_{\mu'}$ are μ-biworlds, so by the induction hypothesis $w'|_{\mu'} = w''|_{\mu'}$ and $w'|_{\mu'}$ is complete, which contradicts Definition 5.5. So there is at most one μ-biworld $w' \geq_{\mathcal{W}} w$. Now assume that there is some μ-biworld $w' \geq_{\mathcal{W}} w$. Since

w is completed and $w'|_\alpha = w$, the monotonicity of completedness for μ implies that w is completed.

- μ is a limit ordinal. By induction hypothesis, for every $\beta < \mu$ with $\beta \geq \alpha$, there is at most one β-biworld $w_\beta \geq_{\mathcal{W}} w$, and any such w_β is completed. Suppose for a contradiction that there are two distinct μ-biworld s$w', w'' \geq_{\mathcal{W}} w$. Then for some ordinal β with $\alpha < \beta < \mu$, $w''|_\beta \neq w'|_\beta$, contradicting the induction hypothesis (since $w''|_\beta, w'|_\beta \geq_{\mathcal{W}} w$). So there is at most one μ-biworld $\geq_{\mathcal{W}} w$. Assume that $w' \geq_{\mathcal{W}} w$ is such a μ-biworld. Since $w'|_\alpha = w$ and w is completed, the monotonicity of completedness for μ implies that w is completed. □

Lemma 5.3

If monotonicity of completedness and completability hold at all ordinals strictly smaller than μ, then restrictability holds at μ.

Proof. We prove this by induction over μ. The induction hypothesis IH_0 is that restrictability holds at every ordinal $\beta < \mu$.

- $\mu = 0$. Trivial.
- $\mu = \mu' + 1$ is a successor ordinal. Let w be a μ-biworld. We need to show that for every $\alpha \leq \mu$, $w|_\alpha$ is a biworld. We prove this by induction over α. The induction hypothesis IH_1 is that for any $\alpha' < \alpha$, $w|_{\alpha'}$ is a biworld.
  - $\alpha = 0$. $w|_0$ is a biworld by Definition 5.6.
  - $\alpha = \alpha' + 1$ is a successor ordinal. First we show that for each agent $\mathcal{A} \in \boldsymbol{\mathcal{A}}$, $\mathcal{A}^{w|_\alpha} \cap \bar{\mathcal{A}}^{w|_\alpha}$ does not contain completed biworlds. For this, suppose $v \in \mathcal{A}^{w|_\alpha} \cap \bar{\mathcal{A}}^{w|_\alpha}$. By Definition 5.3, there are worlds $w' \in \mathcal{A}^w$ and $w'' \in \bar{\mathcal{A}}^w$ such that $w'|_\alpha = w''|_\alpha = v$. Suppose for a contradiction that v is completed. Since monotonicity of completedness holds for all ordinals $\leq \mu'$, Lemma 5.2 implies that $w' = w''$ and w' is completed. But $w' = w'' \in \mathcal{A}^w \cap \bar{\mathcal{A}}^w = \emptyset$ and w' being completed contradict the fact that w is a biworld.
    What remains to be shown is that $\mathcal{A}^{w|_\alpha} \cup \bar{\mathcal{A}}^{w|_\alpha} = \mathcal{W}^{\alpha'}$.
    $\subseteq$: Assume $v \in \mathcal{A}^{w|_\alpha} \cup \bar{\mathcal{A}}^{w|_\alpha}$. Without loss of generality, assume $v \in \mathcal{A}^{w|_\alpha}$. Since μ is a successor ordinal, $\mathcal{A}^{w|_\alpha} = \{w'|_{\alpha'} \mid w' \in \mathcal{A}^w\}$. So there is some $w' \in \mathcal{A}^w$ such that $v = w'|_{\alpha'}$. By Definition 5.6, the fact that $w' \in \mathcal{A}^w$ implies that w' is a biworld. By induction hypothesis IH_0, restrictability holds at μ'. Since w' is a μ'-biworld and $v = w'|_{\alpha'}$, it follows that v is a biworld, as required.
    $\supseteq$: Assume $v \in \mathcal{W}^{\alpha'}$, i.e., v is an α'-biworld. Since completability holds at μ', there is a μ'-biworld $v' \geq v$. Since w is a μ-biworld, $v' \in \mathcal{A}^w \cup \bar{\mathcal{A}}^w$. Without loss of generality, assume $v' \in \mathcal{A}^w$. Then $v = V'|_\alpha \in \mathcal{A}^{w|_\alpha}$, as required.

  - α is a limit ordinal. By induction hypothesis IH_1, we have that for any $\alpha' < \alpha$, $w|_{\alpha'}$ is a biworld. By Definition 5.3, $w|_\alpha = (w|_{\alpha'})_{\alpha'<\alpha}$. So by Definition 5.6, $w|_\alpha$ is a biworld, as required.
- μ is a limit ordinal. This case directly follows from Definition 5.6. □

Lemma 5.4

If completability holds at μ and the completedness condition holds at all successor ordinals smaller than $pl(\mu)$, then monotonicity of completedness holds at μ.

Proof. Fix α. We need to prove that for each $\mu \geq \alpha$ and each μ-biworld w such that $w|_\alpha$ is completed, w is also completed. We prove this lemma by induction over μ.

- $\mu = \alpha$. Trivial.
- $\mu = \mu' + 1$ is a successor ordinal. Let w be a μ-biworld such that $w|_\alpha$ is completed. By induction hypothesis, $w|_{\mu'}$ is completed. Since completability holds at μ, there is a completed μ-biworld $w' \geq_{\mathcal{W}} w|_{\mu'}$. Since $w, w' \geq_{\mathcal{W}} w|_{\mu'}$ and $w|_{\mu'}$ is completed, $w = w'$. So w is completed, as required.
- μ is a limit ordinal. Let w be a μ-biworld such that $w|_\alpha$ is completed. By induction hypothesis, $w|_{\alpha+1}$ is completed. Since the completedness condition holds at $\alpha+1$, $\mathcal{A}^{w|_{\alpha+1}} \cap \mathcal{A}^{w|_{\alpha+1}} = \emptyset$. Now we can show by a proof by contradiction that $\mathcal{A} \uparrow w \cap \bar{\mathcal{A}} \uparrow w = \emptyset$: If $v \in \mathcal{A} \uparrow w \cap \bar{\mathcal{A}} \uparrow w$, then $v|_\alpha \in \mathcal{A}^{w|_{\alpha+1}} \cap \mathcal{A}^{w|_{\alpha+1}}$, a contradiction. Thus by Proposition 5.4, w is completed, as required.

□

Lemma 5.5

Let μ be an ordinal. If restrictability holds at each $\alpha \leq \mu$, then completability holds at μ.

Proof. First, we show that for any $\alpha < \mu$ and for each α-biworld w there exists a completed α+1-biworld $w' \geq_{\mathcal{W}} w$.
We proceed by induction on α.

1. $\alpha = 0$. We take $w' := \left(w, (\mathcal{W}^0)_{\mathcal{A}\in\boldsymbol{\mathcal{A}}}, (\emptyset)_{\mathcal{A}\in\boldsymbol{\mathcal{A}}}\right)$. Clearly, w' is a 1-biworld extending w. By Lemma 5.1, w' is a completed 1-biworld.

2. $\alpha = \alpha' + 1$ is a successor ordinal. Fix $\mathcal{A} \in \boldsymbol{\mathcal{A}}$. Since w is a biworld, every biworld v in $\mathcal{A}^w \cap \bar{\mathcal{A}}^w$ is incompleted, i.e., there exist distinct α-biworld $v_1, v_2 \geq_{\mathcal{W}} v$. By the axiom of choice, we can define a function $f_{\mathcal{A}} : \mathcal{A}^w \cap \bar{\mathcal{A}}^w \to \mathcal{W}^\alpha$ by sending each v in $\mathcal{A}^w \cap \bar{\mathcal{A}}^w$ to an extension $f_{\mathcal{A}}(v) \in \mathcal{W}^\alpha$. Let $S_{\mathcal{A}} := \{f_{\mathcal{A}}(v) \mid v \in \mathcal{A}^w \cap \bar{\mathcal{A}}^w\}$. Notice that, since biworlds in $\mathcal{A}^w \cap \bar{\mathcal{A}}^w$ are incompleted, each α'-biworld $v \in \mathcal{A}^w \cap \bar{\mathcal{A}}^w$

has an extension in $S_{\mathcal{A}}$ and an extension not in $S_{\mathcal{A}}$. Then we take

$$w' := \Big(w^{obj}, (ext(\mathcal{A}^w) \setminus S_{\mathcal{A}})_{\mathcal{A}\in\boldsymbol{\mathcal{A}}}, ((\mathcal{W}^\alpha \setminus ext(\mathcal{A}^w)) \cup S_{\mathcal{A}})_{\mathcal{A}\in\boldsymbol{\mathcal{A}}}\Big),$$

where $ext(\mathcal{A}^w) := \{u \in \mathcal{W}^\alpha \mid u|_{\alpha'} \in \mathcal{A}^w\}$. It is easy to see that w' is an α+1-biworld, and by Lemma 5.1, w' is completed. It remains to show that $w' \geq_{\mathcal{W}} w$. For each agent $\mathcal{A} \in \boldsymbol{\mathcal{A}}$, we need to prove two equalities:

(a) $\mathcal{A}^{w'|_\alpha} = \mathcal{A}^w$.
Let $v \in \mathcal{A}^{w'|_\alpha}$. Then, by Defintion 5.3, there exists $v' \in \mathcal{A}^{w'} \subseteq ext(\mathcal{A}^w)$ such that $v' \geq_{\mathcal{W}} v$. Hence, $v = v'|_{\alpha'} \in \mathcal{A}^w$, as desired. Finally, for $\mathcal{A}^w \subseteq \mathcal{A}^{w'|_\alpha}$, let $v \in \mathcal{A}^w$. By induction hypothesis, there exists an α-biworld v' such that $v' \geq_{\mathcal{W}} v$. If $v \notin \mathcal{A}^w \cap \bar{\mathcal{A}}^w$, then by uniqueness of restriction $v' \in ext(\mathcal{A}^w) \setminus S_{\mathcal{A}}$. Hence $v = v'|_\alpha \in \mathcal{A}^{w'|_\alpha}$, as desired. If $v \in \mathcal{A}^w \cap \bar{\mathcal{A}}^w$, then there exists $v'' \in ext(\mathcal{A}^w)$, such that $v'' \geq_{\mathcal{W}} v$ and $v'' \neq f_{\mathcal{A}}(v)$. Hence, $v = v''|_{\alpha'} \in \mathcal{A}^{w'|_\alpha}$, as desired.

(b) $\bar{\mathcal{A}}^{w'|_\alpha} = \bar{\mathcal{A}}^w$.
Let $v \in \bar{\mathcal{A}}^{w'|_\alpha}$. Then, by Defintion 5.3, there exists $v' \in \bar{\mathcal{A}}^{w'}$ such that $v' \geq_{\mathcal{W}} v$. If $v' \in S_{\mathcal{A}}$, then $v = v'|_{\alpha'} \in \bar{\mathcal{A}}^w$, as desired. Otherwise, $v' \in \mathcal{W}^\alpha \setminus ext(\mathcal{A}^w)$ and $v = v'|_{\alpha'} \notin \mathcal{A}^w$. Since restrictability holds at level α, $v = v'|_{\alpha'}$ is a biworld. This together with the facts that w is a biworld and that $v \notin \mathcal{A}^w$ implies that $v \in \bar{\mathcal{A}}^w$, as desired. Hence, $\bar{\mathcal{A}}^{w'|_\alpha} \subseteq \bar{\mathcal{A}}^w$. For the other inclusion, let $v \in \bar{\mathcal{A}}^w$. If $v \in \mathcal{A}^w$, then $v = f_{\mathcal{A}}(v)|_{\alpha'} \in \bar{\mathcal{A}}^{w'|_\alpha}$, as desired. If $v \notin \mathcal{A}^w$, then by the induction hypothesis there exists $v' \in \mathcal{W}^\alpha \setminus ext(\mathcal{A}^w)$ such that $v' \geq_{\mathcal{W}} v$. Hence, $v = v'|_{\alpha'} \in \bar{\mathcal{A}}^{w'|_\alpha}$.

3. α is a limit ordinal. Fix $\mathcal{A} \in \boldsymbol{\mathcal{A}}$. By induction hypothesis, for each $v \in \mathcal{A}{\uparrow}w \cap \bar{\mathcal{A}}{\uparrow}w$ and for each $\beta < \alpha$, there exists a completed β+1-biworld $v^*_{\beta+1} \geq_{\mathcal{W}} (v)_\beta$. Using this notation, we define some sets of biworlds to construct the desired completed biworld w':

$$S_0^{\mathcal{A}} := \mathcal{A}{\uparrow}w \setminus \bar{\mathcal{A}}{\uparrow}w$$

$$S_1^{\mathcal{A}} := \left\{ v \in \mathcal{A}{\uparrow}w \cap \bar{\mathcal{A}}{\uparrow}w \;\middle|\; \exists \delta < \alpha \text{ s.t. for every } \beta \text{ s.t. } \delta < \beta < \alpha, v^*_{\beta+1} \in \bar{\mathcal{A}}^{(w)_{\beta+2}} \right\}.$$

Now define $w' := (w^{obj}, (S_0^{\mathcal{A}} \cup S_1^{\mathcal{A}})_{\mathcal{A}\in\boldsymbol{\mathcal{A}}}, (\mathcal{W}^\alpha \setminus (S_0^{\mathcal{A}} \cup S_1^{\mathcal{A}}))_{\mathcal{A}\in\boldsymbol{\mathcal{A}}})$. Clearly w' is a completed $\alpha + 1$-biworld. In order to show that $w \leq_{\mathcal{W}} w'$, one needs to prove that for all $\delta < \alpha$, $(w)_\delta \leq_{\mathcal{W}} w'$. For this it is enough to show that for all successor ordinals $\delta + 1 < \alpha$, $(w)_{\delta+1} \leq_{\mathcal{W}} w'$. In more detail, we need to show that for every $\delta < \alpha$ and every $\mathcal{A} \in \boldsymbol{\mathcal{A}}$, $\mathcal{A}^{(w)_{\delta+1}} = \{(v)_\delta \mid v \in S_0^{\mathcal{A}} \cup S_1^{\mathcal{A}}\}$ and $\bar{\mathcal{A}}^{(w)_{\delta+1}} = \{(v)_\delta \mid v \in \mathcal{W}^\alpha \setminus (S_0^{\mathcal{A}} \cup S_1^{\mathcal{A}})\}$. So fix $\delta < \alpha$ and fix $\mathcal{A} \in \boldsymbol{\mathcal{A}}$. We prove the two required statements one after the other:

(a) $\mathcal{A}^{(w)_{\delta+1}} = \{(v)_\delta \mid v \in S_0^{\mathcal{A}} \cup S_1^{\mathcal{A}}\}$.
If $v \in S_0^{\mathcal{A}} \cup S_1^{\mathcal{A}}$, then $v \in \mathcal{A}{\uparrow}w$, so $(v)_\delta \in \mathcal{A}^{(w)_{\delta+1}}$. Additionally, we need to show that for any $u \in \mathcal{A}^{(w)_{\delta+1}}$, there is an α-biworld $v \geq_{\mathcal{W}} u$ such that $v \in S_0^{\mathcal{A}} \cup S_1^{\mathcal{A}}$. So fix $u \in \mathcal{A}^{(w)_{\delta+1}}$. By Proposition 5.5, there exists $\tilde{u} \in \mathcal{A}{\uparrow}w$ such that $\tilde{u} \geq_{\mathcal{W}} u$. If $\tilde{u} \in S_0^{\mathcal{A}}$ or $\tilde{u} \in S_1^{\mathcal{A}}$, then $\tilde{u}$ is the desired extension of u. If $\tilde{u} \notin S_0^{\mathcal{A}} \cup S_1^{\mathcal{A}}$, then $\tilde{u} \in \mathcal{A}{\uparrow}w \cap \bar{\mathcal{A}}{\uparrow}w$ and for all δ there exists β such that $\delta < \beta < \alpha$ and $\tilde{u}^*_{\beta+1} \notin \bar{\mathcal{A}}^{(w)_{\beta+2}}$. Then we can apply Proposition 5.5 to $\tilde{u}^*_{\beta+1}$ to obtain an α-biworld $\tilde{u}' \geq_{\mathcal{W}} \tilde{u}^*_{\beta+1} \geq_{\mathcal{W}} u$. Since $\tilde{u}^*_{\beta+1} \notin \bar{\mathcal{A}}^{(w)_{\beta+2}}$, $\tilde{u}'$ is in $S_0^{\mathcal{A}}$, as desired.

(b) $\bar{\mathcal{A}}^{(w)_{\delta+1}} = \{(v)_\delta \mid v \in \mathcal{W}^\alpha \setminus (S_0^{\mathcal{A}} \cup S_1^{\mathcal{A}})\}$.
By definition of $S_0^{\mathcal{A}}$, for any $v \in \mathcal{W}^\alpha \setminus (S_0^{\mathcal{A}} \cup S_1^{\mathcal{A}})$, we have $v \in \bar{\mathcal{A}}{\uparrow}w$, i.e., $(v)_\delta \in \bar{\mathcal{A}}^{(w)_{\delta+1}}$. Additionally, we need to show that for any $u \in \bar{\mathcal{A}}^{(w)_{\delta+1}}$, there is an α-biworld $v \geq_{\mathcal{W}} u$ such that $v \in \mathcal{W}^\alpha \setminus (S_0^{\mathcal{A}} \cup S_1^{\mathcal{A}})$, i.e., $v \notin S_0^{\mathcal{A}}$ and $v \notin S_1^{\mathcal{A}}$. So fix $u \in \bar{\mathcal{A}}^{(w)_{\delta+1}}$. We follow a technique similar to the one adopted for case 3a above. By the precision order of limit biworlds, for each $\beta < \gamma < \alpha$ and for each $v \in \bar{\mathcal{A}}^{(w)_{\beta+1}}$ there exists $v' \in \bar{\mathcal{A}}^{(w)_{\gamma+1}}$ such that $v' \geq_{\mathcal{W}} v$. Hence, for each $\eta < \alpha$, we can define a function $f^\eta_{\bar{\mathcal{A}}} : W^\eta_{\bar{\mathcal{A}}} \to \bar{\mathcal{A}}^{(w)_{\eta+1}}$, where $W^\eta_{\bar{\mathcal{A}}} := \bigcup_{\gamma<\eta} \bar{\mathcal{A}}^{(w)_{\gamma+1}}$, such that for each $v \in W^\eta_{\bar{\mathcal{A}}}$, $f^\eta_{\bar{\mathcal{A}}}(v) \in \bar{\mathcal{A}}^{(w)_{\eta+1}}$ and $f^\eta_{\bar{\mathcal{A}}}(v) \geq_{\mathcal{W}} v$. Since α is a countable limit ordinal, we can choose a strictly increasing sequence $(\rho_n)_{n<\omega}$ of ordinals which has limit α. W.l.o.g. we assume $\rho_0 > \delta$. Then, we define $\tilde{u} \in \mathcal{W}^\alpha$ as follows:

$$\tilde{u}_\beta := \begin{cases} u|_\beta & \text{if } \beta \leq \delta \\ f^{\rho_0}_{\bar{\mathcal{A}}}(u)|_\beta & \text{if } \delta < \beta \leq \rho_0 \\ f^{\rho_{n+1}}_{\bar{\mathcal{A}}}(\tilde{u}_{\rho_n})|_\beta & \text{if } \rho_n < \beta \leq \rho_{n+1} \end{cases} . \tag{5.2}$$

Clearly, $\tilde{u} \in \bar{\mathcal{A}}{\uparrow}w$ and $\tilde{u} \geq_{\mathcal{W}} u$. Hence, $\tilde{u} \notin S_0^{\mathcal{A}}$. If $\tilde{u} \notin S_1^{\mathcal{A}}$, then $\tilde{u}$ is the desired extension of u. If $\tilde{u} \in S_1^{\mathcal{A}}$, then there exists $\beta < \alpha$ such that $\tilde{u}^*_{\beta+1} \in \bar{\mathcal{A}}^{(w)_{\beta+2}}$. Notice that, since $\tilde{u}^*_{\beta+1}$ is completed, we have $\tilde{u}^*_{\beta+1} \in \bar{\mathcal{A}}^{(w)_{\beta+2}} \setminus \mathcal{A}^{(w)_{\beta+2}}$. Then we can apply to $\tilde{u}^*_{\beta+1}$ a construction analogous to (5.2), to obtain an α-biworld $\tilde{u}' \geq_{\mathcal{W}} \tilde{u}^*_{\beta+1} \geq_{\mathcal{W}} u$. Clearly, $\tilde{u}'$ is in $\bar{\mathcal{A}}{\uparrow}w \setminus \mathcal{A}{\uparrow}w$, hence not in $S_0^{\mathcal{A}} \cup S_1^{\mathcal{A}}$, as desired.

So far, we have shown that for any $\alpha < \mu$ and for each α-biworld w there exists a completed α+1-biworld $w' \geq_{\mathcal{W}} w$. We want to show that for any $\alpha < \mu$ and any α-biworld w, there exists a completed μ-biworld $w' \geq_{\mathcal{W}} w$. We prove this by induction on μ:

1. $\mu = \mu' + 1$ is a successor ordinal. In this case we include the initial step of the induction, namely $\mu' = \alpha$. Let w be an α-biworld. By induction hypothesis, there

exists a μ'-biworld $w'' \geq_{\mathcal{W}} w$ (for the initial step we have $w'' := w$). By the first part of the proof, there exists a completed μ-biworld $w' \geq_{\mathcal{W}} w'' \geq_{\mathcal{W}} w$.

2. μ is a limit ordinal. Let w be an α-biworld. By induction hypothesis, for each ordered pair of ordinals (γ, δ) such that $\alpha \leq \gamma < \delta < \mu$ there exists a function $f_\gamma^\delta : \mathcal{W}^\gamma \to \mathcal{W}^\delta$ such that, for each $v \in \mathcal{W}^\gamma$, $f_\gamma^\delta(v) \geq_{\mathcal{W}} v$ and $f_\gamma^\delta(v)$ is completed. By the construction of the completed biworld $w' \geq_{\mathcal{W}} w$ in the first part of the proof of this Lemma, not only $f_\gamma^\delta(v)$ is completed, but also $\mathcal{A}^{f_\gamma^\delta(v)} \cap \bar{\mathcal{A}}^{f_\gamma^\delta(v)} = \emptyset$ for any agent $\mathcal{A} \in \boldsymbol{\mathcal{A}}$. Since α is a countable limit ordinal, we can choose a strictly increasing sequence of ordinals $(\rho_n)_{n<\omega}$ which has limit μ. W.l.o.g. we assume $\alpha < \rho_0$. Then, we define $w' \in \mathcal{W}^\mu$ as follows:

$$(w')_\beta := \begin{cases} w|_\beta & \text{if } \beta \leq \alpha \\ f_\alpha^{\rho_0}(w)|_\beta & \text{if } \alpha < \beta \leq \rho_0 \\ f_{\rho_n}^{\rho_{n+1}}((w')_{\rho_n})|_\beta & \text{if } \rho_n < \beta \leq \rho_{n+1} \end{cases}.$$

Clearly, $w' \geq_{\mathcal{W}} w$. Since $\mathcal{A}^{f_\alpha^{\rho_0}(v)} \cap \bar{\mathcal{A}}^{f_\alpha^{\rho_0}(v)} = \emptyset$ for any agent $\mathcal{A} \in \boldsymbol{\mathcal{A}}$, we have that $\mathcal{A}{\uparrow}w' \cap \bar{\mathcal{A}}{\uparrow}w' = \emptyset$ for all $\mathcal{A} \in \boldsymbol{\mathcal{A}}$. By Proposition 5.4, w' is completed, as desired.

□

Lemma 5.6

If completability holds at $\mu + 2$, then the completedness condition holds at $\mu + 1$.

Proof. Let w be a μ+1-biworld. By Lemma 5.1, if for each $\mathcal{A} \in \boldsymbol{\mathcal{A}}$, $\mathcal{A}^w \cap \bar{\mathcal{A}}^w = \emptyset$, then w is completed.
To prove the other direction needed for the completedness condition, assume there exist $\mathcal{A}_* \in \boldsymbol{\mathcal{A}}$ and $v \in \mathcal{A}_*^w \cap \bar{\mathcal{A}}_*^w$. By definition of biworld, v must be incompleted, i.e., there exist $v_1, v_2 \in \mathcal{W}^{\mu+1}$ such that $v_1, v_2 \geq_{\mathcal{W}} v$. We will show that we have two distinct μ+2-biworlds $w_1 \neq w_2$ such that $w_1, w_2 \geq_{\mathcal{W}} w$, i.e., w is not completed. By completability at $\mu + 2$, there exists a completed μ+2-biworld $w_1 \geq_{\mathcal{W}} w$. Note that for $i \in \{1, 2\}$, v_i is in precisely one of the two sets $\mathcal{A}_*^{w_1}$ and $\bar{\mathcal{A}}_*^{w_1}$. Now we define w_2 to be the μ+2-biworld with $w_2^{obj} := w_1^{obj}$, for each $\mathcal{A} \in \boldsymbol{\mathcal{A}} \setminus \{\mathcal{A}_*\}$, $\mathcal{A}^{w_2} := \mathcal{A}^{w_1}$ and $\bar{\mathcal{A}}^{w_2} := \bar{\mathcal{A}}^{w_1}$, and the sets $\mathcal{A}_*^{w_2}$ and $\bar{\mathcal{A}}_*^{w_2}$ defined as follows:

$$\mathcal{A}_*^{w_2} := \begin{cases} \mathcal{A}_*^{w_1} \setminus \{v_2\} & \text{if } v_1, v_2 \in \mathcal{A}_*^{w_1} \\ \mathcal{A}_*^{w_1} \cup \{v_2\} & \text{if } v_1, v_2 \in \bar{\mathcal{A}}_*^{w_1} \\ \mathcal{A}_*^{w_1} \cup \{v_2\} \setminus \{v_1\} & \text{if } v_1 \in \mathcal{A}_*^{w_1}, v_2 \in \bar{\mathcal{A}}_*^{w_1} \\ \mathcal{A}_*^{w_1} \cup \{v_1\} \setminus \{v_2\} & \text{if } v_1 \in \bar{\mathcal{A}}_*^{w_1}, v_2 \in \mathcal{A}_*^{w_1} \end{cases}$$

$$\bar{\mathcal{A}}_*^{w_2} := \begin{cases} \bar{\mathcal{A}}_*^{w_1} \cup \{v_2\} & \text{if } v_1, v_2 \in \mathcal{A}_*^{w_1} \\ \bar{\mathcal{A}}_*^{w_1} \setminus \{v_2\}) & \text{if } v_1, v_2 \in \bar{\mathcal{A}}_*^{w_1} \\ \bar{\mathcal{A}}_*^{w_1} \cup \{v_1\} \setminus \{v_2\} & \text{if } v_1 \in \mathcal{A}_*^{w_1}, v_2 \in \bar{\mathcal{A}}_*^{w_1} \\ \bar{\mathcal{A}}_*^{w_1} \cup \{v_2\} \setminus \{v_1\}) & \text{if } v_1 \in \bar{\mathcal{A}}_*^{w_1}, v_2 \in \mathcal{A}_*^{w_1} \end{cases}$$

It is easy to see that $w_2 \geq_{\mathcal{W}} w$, as v_1 and v_2 are extensions of $v \in \mathcal{A}_*^w \cap \bar{\mathcal{A}}_*^w$. Since $w_1 \neq w_2$, w is incompleted, as required. □

Lemma 5.7

For every ordinal μ, biworlds are well-behaved until μ.

Proof. We prove this by induction over μ:

- $\mu = 0$: Trivial.
- $\mu = \mu' + 1$ is a successor ordinal. The induction hypothesis is that biworlds are well-behaved until μ', i.e., that restrictability, completability and monotonicity of completedness hold at all ordinals smaller than μ' and the completedness condition holds at all successor ordinals smaller than $pl(\mu')$. Now Lemma 5.3 implies that restrictability holds at μ'. This together with Lemma 5.5 implies that completability holds at μ'. It now follows from Lemma 5.4 that monotonicity of completedness holds at μ'. Note that since $pl(\mu) = pl(\mu')$, the induction hypothesis implies that the completedness condition holds at all successor ordinals smaller than $pl(\mu)$. Thus biworlds are well-behaved until μ, as required.
- μ is a limit ordinal. The induction hypothesis is that for every $\alpha < \mu$, biworlds are well-behaved until α, i.e., that restrictability, completability and monotonicity of completedness hold at all ordinals smaller than α and the completedness condition holds at all successor ordinals smaller than $pl(\alpha)$. Since this holds for all $\alpha < \mu$, it directly follows that restrictability, completability and monotonicity of completedness hold at all ordinals smaller than μ. Note that $pl(\mu) = \mu$. So in order to conclude that biworlds are well-behaved until μ, we still need to show that the completedness condition holds at all successor ordinals smaller than μ. Let $\beta + 1 < \mu$. Now the induction hypothesis with $\alpha = \beta + 3$ implies that completability holds at $\beta + 2$. So by Lemma 5.6, the completedness condition holds at $\beta + 1$, as required.

□

Now the following theorem directly follows from Lemma 5.7.

Theorem 5.1

Let $\alpha \leq \mu' \leq \mu < \beta$ be ordinals, let w be a μ-biworld, and w' be a μ'-biworld. The following hold:

1. The restriction $w|_\alpha$ is an α-biworld.
2. If w' is completed and $w \geq_\mathcal{W} w'$, then w is completed.
3. There exists a completed β-biworld v such that $v \geq_\mathcal{W} w$.
4. If μ is a successor ordinal, then w is completed if and only if for each $\mathcal{A} \in \boldsymbol{\mathcal{A}}$, $\mathcal{A}^w \cap \bar{\mathcal{A}}^w = \emptyset$.

Theorem 5.1 directly implies an important corollary:

Corollary 5.1

A μ-biworld w is completed if and only if there exists exactly one μ+1-biworld v such that $v \geq_\mathcal{W} w$. In this case, also v is completed.

Proof. If a μ-biworld w is completed, then it has at most one extension at $\mu + 1$, by definition of completedness. By Item 3 of Theorem 5.1 with $\alpha = \mu + 1$, w has at least one extension, proving one direction. Conversely, if there is exactly one μ+1-biworld v such that $v \geq_\mathcal{W} w$, then w cannot be incompleted, by definition. Notice that v has to be completed by Item 2 of Theorem 5.1. □

4 The Logic $\mathcal{COL}$

In this section we define the syntax and semantics of the logic $\mathcal{COL}$ that allows to speak about knowledge, common knowledge and only knowing. More specifically, we first define a three-valued model semantics, where the value of a $\mathcal{COL}$ formula is either true (**t**), false (**f**), or unknown (**u**). We consider two orders[2] on these truth values: the *precision order* given by $\mathbf{u} \leq_p \mathbf{f}$ and $\mathbf{u} \leq_p \mathbf{t}$, and the *truth order* given by $\mathbf{f} \leq_{tr} \mathbf{u} \leq_{tr} \mathbf{t}$. As usual, we will write v^{-1} for the *inverse* of the truth value v, defined by $\mathbf{f}^{-1} = \mathbf{t}$, $\mathbf{t}^{-1} = \mathbf{f}$, and $\mathbf{u}^{-1} = \mathbf{u}$. We will show that our three-valued semantics is precision-monotonic, in the sense that more precise biworlds give more precise results. Moreover, we will show that in a biworld of depth at least ω^2, every formula evaluates to either true or false. This fact prompts us to define an alternative two-valued semantics. In more detail, since dealing

[2]These coincide with the homonimous orders derived from the order $\leq_o$ used in the context of AFT (Example 2.13 in Chapter 2).

with biworlds with a limit ordinal depth may be complicated and counter-intuitive, we will move our focus to the smallest (for the sake of simplicity) successor ordinal at which biworlds evaluate all formulas (and all sets of formulas) as true or false, namely $\omega^2 + 1$. In addition, we restrict to completed ω^2+1-biworlds: by Item 4 in Theorem 5.1, they have the intuitive property that $\mathcal{A}^w$ and $\bar{\mathcal{A}}^w$ are disjoint, which corresponds to the intuition that these two sets represent the biworlds deemed possible and the biworlds deemed impossible by $\mathcal{A}$. This motivates an alternative semantic characterization of our logic through a canonical Kripke structure consisting of completed ω^2+1-biworlds[3], which we show to coincide with the three-valued semantics on the relevant biworlds. Furthermore, we show that $\mathcal{COL}$ is semantically well-behaved. In Section 7, we will show that no previously proposed semantic approach leads to a logic that is semantically well-behaved in the way specified in this section.

To say that an agent $\mathcal{A}$ only knows φ can be viewed as a conjunction of the statement that $\mathcal{A}$ knows φ, denoted $K_{\mathcal{A}}\varphi$, and the statement that $\mathcal{A}$ knows at most φ, denoted $M_{\mathcal{A}}\varphi$. Given a set G of agents, we write $E_G\varphi$ for the statement that every agent in G knows φ, and we write $C_G\varphi$ for the statement that φ is common knowledge within the set G of agents. The language $\mathcal{COL}$ extends propositional logic with these modal operators as follows:

Definition 5.13

We define the language $\mathcal{COL}$ by structural induction with the standard recursive rules of propositional logic, augmented with:

$$K_{\mathcal{A}}(\psi) \in \mathcal{COL} \text{ if } \psi \in \mathcal{COL} \text{ and } \mathcal{A} \in \boldsymbol{\mathcal{A}}$$
$$M_{\mathcal{A}}(\psi) \in \mathcal{COL} \text{ if } \psi \in \mathcal{COL} \text{ and } \mathcal{A} \in \boldsymbol{\mathcal{A}}$$
$$E_G(\psi) \in \mathcal{COL} \text{ if } \psi \in \mathcal{COL} \text{ and } G \subseteq \boldsymbol{\mathcal{A}}$$
$$C_G(\psi) \in \mathcal{COL} \text{ if } \psi \in \mathcal{COL} \text{ and } G \subseteq \boldsymbol{\mathcal{A}}$$

We use $O_{\mathcal{A}}\varphi$ as syntactic sugar for $K_{\mathcal{A}}\varphi \wedge M_{\mathcal{A}}\varphi$.

In the introduction, we have already discussed what an intuitive Kripke semantics for $O_{\mathcal{A}}\varphi$ would be. Adapting these ideas to the representation of $O_{\mathcal{A}}\varphi$ as $K_{\mathcal{A}}\varphi \wedge M_{\mathcal{A}}\varphi$, we can easily see that the correct way to define the Kripke semantics for $M_{\mathcal{A}}\varphi$ is as follows:

- $\boldsymbol{\mathcal{K}}, w \vDash M_{\mathcal{A}}\varphi$ if for every world $w' \in W$ such that $\boldsymbol{\mathcal{K}}, w' \vDash \varphi$, we have $(w, w') \in R_{\mathcal{A}}$.

[3]For any $\mu \geq \omega^2$, all the results would hold if we considered a canonical Kripke structure consisting of completed μ-biworlds. Taking all completed μ-biworlds for all ordinals μ would also work, but some uniqueness results would be lost, as we would have many worlds representing the same object.

In order to explain why this is a good semantic characterization of "knowing at most φ", we will sketch a proof that the only way in which $M_{\mathcal{A}}\varphi$ and $K_{\mathcal{A}}\psi$ can both be true is when φ entails ψ: Assume $M_{\mathcal{A}}\varphi$ and $K_{\mathcal{A}}\psi$ are true in a world w. We want to show that φ entails ψ, i.e., that ψ is true in every world w' in which φ is true. Assume w' is a world in which φ is true. But the assumption that $M_{\mathcal{A}}\varphi$ is true in w together with the above definition of the Kripke semantics for $M_{\mathcal{A}}\varphi$ implies that $(w, w') \in R_{\mathcal{A}}$. This together with the assumption that $K_{\mathcal{A}}\psi$ is true in w implies that ψ is true in w', as required.

In preparation for the upcoming discussion of a three-valued semantics for $\mathcal{COL}$, note that if we write $(M_{\mathcal{A}}\varphi)^{\mathcal{K},w} = \mathbf{t}$ and $(M_{\mathcal{A}}\varphi)^{\mathcal{K},w} = \mathbf{f}$ for $\mathcal{K}, w \models M_{\mathcal{A}}\varphi$ and $\mathcal{K}, w \not\models M_{\mathcal{A}}\varphi$ respectively, the above characterization of the semantics of $M_{\mathcal{A}}\varphi$ is equivalent to the following:

$$(M_{\mathcal{A}}\varphi)^{\mathcal{K},w} = \mathrm{glb}_{\leq_{tr}}(\{(\varphi^{\mathcal{K},w'})^{-1} \mid w' \notin R^w_{\mathcal{A}}\})$$

Let us now turn to the three-valued valuation of formulas of $\mathcal{COL}$. All parts of this definition are precisely what one would expect when applying a Kleene-style three-valued semantic approach to logics with a common-knowledge operator, taking into account the above rewording of the Kripke semantics of $M_{\mathcal{A}}\varphi$.

Definition 5.14

Given a formula $\varphi \in \mathcal{COL}$ and a μ-biworld w, we define the three-valued valuation function φ^w by induction on μ and the structure of φ:

$$
\begin{aligned}
P^w &= \mathbf{t} \text{ if } P \in w^{obj} \text{ and } P^w = \mathbf{f} \text{ otherwise} \\
(\varphi \wedge \psi)^w &= \mathrm{glb}_{\leq_{tr}}(\{\varphi^w, \psi^w\}) \\
(\neg\varphi)^w &= (\varphi^w)^{-1} \\
(K_{\mathcal{A}}\varphi)^w &= \begin{cases} \mathbf{u} & \text{if } \mu = 0 \\ \mathrm{glb}_{\leq_{tr}}(\{\varphi^{w'} \mid w' \in \mathcal{A}^w\}) & \text{if } \mu = \mu'+1 \\ \mathrm{lub}_{\leq_p}(\{(K_{\mathcal{A}}\varphi)^{(w)_{\mu'}} \mid \mu' < \mu\}) & \text{if } \mu \text{ is a limit ordinal} \end{cases} \\
(M_{\mathcal{A}}\varphi)^w &= \begin{cases} \mathbf{u} & \text{if } \mu = 0 \\ \mathrm{glb}_{\leq_{tr}}(\{(\varphi^{w'})^{-1} \mid w' \in \bar{\mathcal{A}}^w\}) & \text{if } \mu = \mu'+1 \\ \mathrm{lub}_{\leq_p}(\{(M_{\mathcal{A}}\varphi)^{(w)_{\mu'}} \mid \mu' < \mu\}) & \text{if } \mu \text{ is a limit ordinal} \end{cases} \\
(E_G\varphi)^w &= \mathrm{glb}_{\leq_{tr}}(\{(K_{\mathcal{A}}\varphi)^w \mid \mathcal{A} \in G\}) \\
(C_G\varphi)^w &= \mathrm{glb}_{\leq_{tr}}(\{(E_G^k\varphi)^w \mid k \geq 1\})
\end{aligned}
$$

where we define $E_G^k\varphi$ inductively as $E_G^0\varphi = \varphi$, $E_G^{k+1}\varphi = E_G(E_G^k\varphi)$. We say a μ-biworld w *satisfies a formula* φ (notation: $w \models \varphi$) if $\varphi^w = \mathbf{t}$. A μ-biworld w satisfies, or is a model of, a theory if it satisfies all formulas in that theory. We say a μ-biworld w *resolves* a formula φ if $\varphi^w \neq \mathbf{u}$.

The following proposition asserts that the three-valued valuation is monotonic:

Proposition 5.6

For every pair w, w' of biworlds such that $w \leq_{\mathcal{W}} w'$ and every formula $\varphi \in \mathcal{COL}$, we have $\varphi^w \leq_p \varphi^{w'}$.

Proof. We prove this proposition by induction over φ. We distinguish the following cases:

- $\varphi = P$. Since $w'^{obj} = w'|_0 = w|_{d(w)}|_0 = w|_0 = w^{obj}$, we have $\varphi^w = \varphi^{w^{obj}} = \varphi^{w'^{obj}} = \varphi^{w'}$.
- $\varphi = \psi_1 \wedge \psi_2$. Then

$$\varphi^w = \mathrm{glb}_{\leq_{tr}}(\{\psi_1^w, \psi_2^w\}) \leq_p \mathrm{glb}_{\leq_{tr}}(\{\psi_1^{w'}, \psi_2^{w'}\}) = \varphi^{w'},$$

where the central inequality comes from the induction hypothesis and $\mathsf{glb}_{\leq_{tr}}$ being monotonic.

- $\varphi = \neg\psi$. Then $\varphi^w = (\psi^w)^{-1} \leq_p (\psi^{w'})^{-1} = \varphi^{w'}$, where the central inequality comes from the induction hypothesis and $(\cdot)^{-1}$ being monotonic.
- $\varphi = K_{\mathcal{A}}\psi$. We prove this case by an induction on $d(w')$:
  1. $d(w') = d(w)$. Then $w = w'$.
  2. $d(w')$ is a successor ordinal. Here we distinguish three cases:
     (a) $d(w) = 0$. Then $\varphi^w = \mathbf{u} \leq_p \varphi^{w'}$.
     (b) $d(w)$ is a successor ordinal. By the definition of restriction, every biworld in $\mathcal{A}^w$ has an extension in $\mathcal{A}^{w'}$ and the restriction of every biworld in $\mathcal{A}^{w'}$ is in $\mathcal{A}^w$. Hence, by the induction hypothesis on φ, and $\mathsf{glb}_{\leq_{tr}}$ being monotonic, we have $\varphi^w = \mathrm{glb}_{\leq_{tr}}(\{\psi^v \mid v \in \mathcal{A}^w\}) \leq_p \mathrm{glb}_{\leq_{tr}}(\{\psi^v \mid v \in \mathcal{A}^{w'}\}) = \varphi^{w'}$.
     (c) $d(w)$ is a limit ordinal. Then by the limit ordinal case of the definition of $(K_{\mathcal{A}}\varphi)^w$, there is a successor ordinal $\mu' < d(w)$ such that $(K_{\mathcal{A}}\psi)^{w|_{\mu'}} = (K_{\mathcal{A}}\psi)^w$. Then by case 2b, $\varphi^w \leq_p \varphi^{w'}$.
  3. $d(w')$ is a limit ordinal. Then by the limit ordinal case of the definition of $(K_{\mathcal{A}}\varphi)^w$, there is an ordinal $\mu' < d(w')$ such that $(K_{\mathcal{A}}\psi)^{w'|_{\mu'}} = (K_{\mathcal{A}}\psi)^{w'}$. By the induction hypothesis, $\varphi^w \leq_p (K_{\mathcal{A}}\psi)^{w'|_{\mu'}} = \varphi^{w'}$.
- $\varphi = M_{\mathcal{A}}\psi$. The proof is analogous to the one for the case $\varphi = K_{\mathcal{A}}\psi$.
- $\varphi = E_G\psi$. Then

$$\begin{aligned}\varphi^w = (E_G\psi)^w &= \mathrm{glb}_{\leq_{tr}}(\{(K_{\mathcal{A}}\psi)^w \mid \mathcal{A} \in G\}) \\ &\leq_p \mathrm{glb}_{\leq_{tr}}(\{(K_{\mathcal{A}}\psi)^{w'} \mid \mathcal{A} \in G\}) = (E_G\psi)^{w'} = \varphi^{w'},\end{aligned}$$

  where the central inequality comes from the induction hypothesis and $\mathsf{glb}_{\leq_{tr}}$ being mmonotonic.
- $\varphi = C_G\psi$. Then

$$\begin{aligned}\varphi^w = (C_G\psi)^w &= \mathrm{glb}_{\leq_{tr}}(\{(E_G^k\psi)^w \mid k \in \mathbb{N}\}) \\ &\leq_p \mathrm{glb}_{\leq_{tr}}(\{(E_G^k\psi)^{w'} \mid k \in \mathbb{N}\}) = (C_G\psi)^{w'} = \varphi^{w'},\end{aligned}$$

  where the central inequality comes from the induction hypothesis and $\mathsf{glb}_{\leq_{tr}}$ being monotonic.

□

The notion of the *modal depth* of a formula allows us to specify conditions for a formula to be resolved by a biworld:

Definition 5.15: Modal depth

The *modal depth* $MD(\varphi)$ of a formula $\varphi \in \mathcal{COL}$ is defined by inductively as follows:

- $MD(P) = 0$ for every propositional atom P
- $MD(\neg\varphi) = MD(\varphi)$
- $MD(\varphi \wedge \psi) = max(MD(\varphi), MD(\psi))$
- $MD(K_A\varphi) = MD(\varphi) + 1$
- $MD(M_A\varphi) = MD(\varphi) + 1$
- $MD(E_G\varphi) = MD(\varphi) + 1$
- $MD(C_G\varphi) = MD(\varphi) + \omega$, which is the smallest limit ordinal greater than $MD(\varphi)$.

Note that the modal depth of any formula in $\mathcal{COL}$ is less than ω^2. Every formula of a given modal depth is resolved at any biworld of at least this depth:

Theorem 5.2

If w is a μ-biworld and $\varphi \in \mathcal{COL}$ is a formula such that $MD(\varphi) \leq \mu$, then w resolves φ.

Proof. We prove this proposition by a simultaneous induction over μ and the complexity of φ. The induction hypothesis is that if either $\alpha < \mu$ and ψ is any formula or $\mathcal{COL}$ or $\alpha = \mu$ and ψ is a subformula of φ, then any α-biworld resolves ψ. For the purpose of this induction (and some similar induction proofs later), we consider $K_A\psi$ to be a subformula of $E_G\psi$ whenever $\mathcal{A} \in G$, and we consider $E^k_G\psi$ to be a subformula of $C_G\psi$ for any $k \geq 1$. It is enough to show that this induction hypothesis together with the assumptions that w is a μ-biworld and $MD(\varphi) \leq \mu$ implies that w resolves φ. For this, we distinguish the following cases:

- $\varphi = P$. Clearly, $\varphi^w \neq \mathbf{u}$.
- $\varphi = \psi_1 \wedge \psi_2$. By Definition 5.15, we have $MD(\psi_1), MD(\psi_2) \leq \mu$. Hence, by induction hypothesis, $(\psi_1 \wedge \psi_2)^w = \mathrm{glb}_{\leq_{tr}}(\{\psi_1^w, \psi_2^w\}) \neq \mathbf{u}$.
- $\varphi = \neg\psi$. By Definition 5.15, we have $MD(\psi) = MD(\varphi) \leq \mu$. By induction hypothesis, $\psi^w \neq \mathbf{u}$, so $(\neg\psi)^w = (\psi^w)^{-1} \neq \mathbf{u}$.
- $\varphi = K_A\psi$. Then $MD(\varphi) \neq 0$, so $\mu \neq 0$. We distinguish two cases:
  1. μ is a successor ordinal. By the induction hypothesis, any world in $\mathcal{A}^w$ resolves ψ. This together with the successor ordinal case of the definition of $(K_A\varphi)^w$ implies that $(K_A\psi)^w \neq \mathbf{u}$, as required.

2. μ is a limit ordinal. Since $MD(\varphi)$ is a successor ordinal, $MD(\varphi) < \mu$, so by induction hypothesis, $w|_{MD(\varphi)}$ resolves φ. Then by Proposition 5.6, w resolves φ, as required.

- $\varphi = M_{\mathcal{A}}\psi$. The proof is analogous to the one for the case $\varphi = K_{\mathcal{A}}\psi$.
- $\varphi = E_G\psi$. By the induction hypothesis, $(K_{\mathcal{A}}\psi)^w \neq \mathbf{u}$ for any $\mathcal{A} \in G$. Hence, $\varphi^w \neq \mathbf{u}$, as desired.
- $\varphi = C_G\psi$. By Definition 5.15, we have $MD(E^k_G\psi) = MD(\psi) + k \leq MD(\psi) + \omega = MD(\varphi) \leq \mu$. By induction hypothesis, $(E^k_G\psi)^w \neq \mathbf{u}$ for all $k \in \mathbb{N}$. Hence, $\varphi^w \neq \mathbf{u}$, as desired.

□

As explained at the beginning of this section, by Item 4 of Theorem 5.1, a μ+1-biworld w precisely captures the knowledge of every agent $\mathcal{A}$ iff w is completed. Hence, Theorem 5.2 combined with the fact that the modal depth of any formula in $\mathcal{COL}$ is less than ω^2 motivates focusing on completed ω^2+1-biworlds, as $\omega^2 + 1$ is the first successor ordinal greater than the modal depth of all formulas.

Definition 5.16

A *world* is a completed ω^2+1-biworld.

Definition 5.17

The ω^2+1-*completed Kripke structure* $\mathcal{K}^* := (\mathfrak{U}, (R_{\mathcal{A}})_{\mathcal{A}\in\mathcal{A}})$ is the Kripke structure whose underlying world set $\mathfrak{U}$ is the set of all worlds, and whose accessibility relations $R_{\mathcal{A}}$ are given by

$$R_{\mathcal{A}} = \{(w, w') \in \mathfrak{U}^2 \mid w'|_{\omega^2} \in \mathcal{A}^w\}.$$

Instead of $(w, w') \in R_{\mathcal{A}}$, we sometimes write $wR_{\mathcal{A}}w'$.

Interpreting formulas in this canonical Kripke structure $\mathcal{K}^*$ in the standard way (with the above specified semantics for $M_{\mathcal{A}}\varphi$) amounts to a two-valued valuation of $\mathcal{COL}$.

Given a world w, we write $R^w_{\mathcal{A}}$ for the set of all worlds $R_{\mathcal{A}}$-accessible from w.

The following lemma is of central importance, as it establishes a relationship between the two-valued and the three-valued valuation and can be used to prove multiple important theorems.

Lemma 5.8

If $\varphi \in \mathcal{COL}$, w is an $MD(\varphi)$-biworld and $w' \geq_w w$ is a world, then $\varphi^w = \varphi^{\mathcal{K}^*,w'}$.

Proof. By Proposition 5.6, $\varphi^w = \varphi^{w'}$. So all that is left to be shown is that $\varphi^{w'} = \varphi^{\mathcal{K}^*,w'}$. We prove this proposition by induction over the complexity of φ. So suppose as induction hypothesis that the lemma holds for all subformulas ψ of φ. In order to prove that the lemma holds for φ, we distinguish the following cases:

- φ is P. Then $\varphi^{w'} = \varphi^{w'^{obj}} = \varphi^{\mathcal{K}^*,w'}$.
- φ is $\psi_1 \wedge \psi_2$. Then

$$\begin{aligned}\varphi^{w'} &= (\psi_1 \wedge \psi_2)^{w'} = \mathrm{glb}_{\leq_{tr}}(\{\psi_1^{w'}, \psi_2^{w'}\}) \\ &= \mathrm{glb}_{\leq_{tr}}(\{\psi_1^{\mathcal{K}^*,w'}, \psi_2^{\mathcal{K}^*,w'}\}) \qquad \text{by induction hypothesis} \\ &= (\psi_1 \wedge \psi_2)^{\mathcal{K}^*,w'} = \varphi^{\mathcal{K}^*,w'}.\end{aligned}$$

- φ is $\neg\psi$. Then $\varphi^{w'} = (\neg\psi)^{w'} = (\psi^{w'})^{-1} = (\psi^{\mathcal{K}^*,w'})^{-1} = (\neg\psi)^{\mathcal{K}^*,w'} = \varphi^{\mathcal{K}^*,w'}$.
- φ is $K_{\mathcal{A}}\psi$. Then $MD(\varphi)$ is a successor ordinal, so $\varphi^{w'} = (K_{\mathcal{A}}\psi)^{w'} = \mathrm{glb}_{\leq_{tr}}(\{\psi^{w^*} \mid w^* \in \mathcal{A}^{w'}\})$. By Item 3 of Theorem 5.1, every ω^2-biworld w^* can be extended to a completed $\omega^2{+}1$-biworld w^+, and by Proposition 5.6, $\psi^{w^*} = \psi^{w^+}$. Thus $\varphi^{w'} = \mathrm{glb}_{\leq_{tr}}(\{\psi^{w^+} \mid w^+ \in R_{\mathcal{A}}^{w'}\})$, which by the induction hypothesis equals $(K_{\mathcal{A}}\psi)^{\mathcal{K}^*,w'}$, as required.
- $\varphi = M_{\mathcal{A}}\psi$. The proof is analogous to the one for the case $\varphi = K_{\mathcal{A}}\psi$.
- φ is $E_G\psi$. Then

$$\begin{aligned}\varphi^w &= (E_G\psi)^w = \mathrm{glb}_{\leq_{tr}}(\{(K_{\mathcal{A}}\psi)^w \mid \mathcal{A} \in G\}) \\ &= \mathrm{glb}_{\leq_{tr}}(\{(K_{\mathcal{A}}\psi)^{\mathcal{K}^*,w'} \mid \mathcal{A} \in G\}) \qquad \text{by induction hypothesis} \\ &= (E_G\psi)^{\mathcal{K}^*,w'} = \varphi^{\mathcal{K}^*,w'}.\end{aligned}$$

- φ is $C_G\psi$. Then

$$\begin{aligned}\varphi^w &= (C_G\psi)^w = \mathrm{glb}_{\leq_{tr}}(\{(E_G^k\psi)^w \mid k \in \mathbb{N}\}) \\ &= \mathrm{glb}_{\leq_{tr}}(\{(E_G^k\psi)^{\mathcal{K}^*,w'} \mid k \in \mathbb{N}\}) \qquad \text{by induction hypothesis} \\ &= (C_G\psi)^{\mathcal{K}^*,w'} = \varphi^{\mathcal{K}^*,w'}.\end{aligned}$$

□

Combining Lemma 5.8 with Proposition 5.6, we directly get the following theorem, which tells us that the two-valued and three-valued valuations fully coincide on worlds, and thus expresses the relationship between the two-valued and the three-valued valuation in a simpler way than Lemma 5.8 does:

Theorem 5.3

If $\varphi \in \mathcal{COL}$ and $w \in \mathfrak{U}$, then $\varphi^w = \varphi^{\mathcal{K}^*,w}$.

Finally, Lemma 5.8 can also be used to prove the following theorem, which is of central importance to show that our semantics generally avoids a problem that some previous accounts of only knowing and common knowledge had, namely the problem that $O_A \neg C_G p$ is not satisfiable in those accounts, even though it should be (see Section 7 for a discussion of this problem in other accounts). The following theorem shows that no such problems can arise in the ω^2+1-completed Kripke structure $\mathcal{K}^*$:

Theorem 5.4

Let $\mathcal{A}$ be an agent. For every formula $\varphi \in \mathcal{COL}$, there is a world w such that $(O_A \varphi)^w = (O_A \varphi)^{\mathcal{K}^*,w} = \mathbf{t}$. Moreover, if w_1 and w_2 are two such worlds, then $\mathcal{A}^{w_1} = \mathcal{A}^{w_2}$ and $\bar{\mathcal{A}}^{w_1} = \bar{\mathcal{A}}^{w_2}$.

Proof. By Theorem 5.3, $(O_A \varphi)^w = (O_A \varphi)^{\mathcal{K}^*,w}$ for any world w. Hence it is enough to show that there is a world w such that $(O_A \varphi)^{\mathcal{K}^*,w} = \mathbf{t}$.
Let v be an arbitrary 0-biworld. Define

$$\begin{aligned} W &:= \{u|_{\omega^2} \mid u \in \mathfrak{U} \text{ and } \varphi^{\mathcal{K}^*,u} = \mathbf{t}\}, \\ \overline{W} &:= \mathcal{W}^{\omega^2} \setminus W, \\ v' &:= (v, (W)_{\mathcal{A} \in \mathcal{A}}, (\overline{W})_{\mathcal{A} \in \mathcal{A}}). \end{aligned}$$

Clearly v' is a world. It follows directly from the definition of v' that $(K_A \varphi)^{\mathcal{K}^*,v'} = \mathbf{t}$ and that $(M_A \varphi)^{\mathcal{K}^*,v'} = \mathbf{t}$, i.e., that $(O_A \varphi)^{\mathcal{K}^*,v'} = \mathbf{t}$.
For the second statement, let w_1, w_2 be worlds such that $(O_A \varphi)^{w_1} = (O_A \varphi)^{w2} = \mathbf{t}$. By the definition of the only-knowing operator, we have $(K_A \varphi)^{w_i} = (M_A \varphi)^{w_i} = \mathbf{t}$ for $i \in \{1, 2\}$. This means that $\varphi^v = \mathbf{t}$ for all $v \in \mathcal{A}^{w_1} \cup \mathcal{A}^{w_2}$, and $\varphi^v = \mathbf{f}$ for all $v \in \bar{\mathcal{A}}^{w_1} \cup \bar{\mathcal{A}}^{w_2}$. Since $\mathcal{A}^u \cup \bar{\mathcal{A}}^u = \mathcal{W}^{\omega^2}$ for any world u, we must have

$$\begin{aligned} \mathcal{A}^{w_1} = \mathcal{A}^{w_2} &= \{v \in \mathcal{W}^{\omega^2} \mid \varphi^v = \mathbf{t}\} \\ \bar{\mathcal{A}}^{w_1} = \bar{\mathcal{A}}^{w_2} &= \{v \in \mathcal{W}^{\omega^2} \mid \varphi^v = \mathbf{f}\}. \end{aligned}$$

□

We will now show that the two-valued valuation of $\mathcal{COL}$ (and thus by Theorem 5.3 also the three-valued valuation, when restricted to suitable biworlds) gives rise to a sensible entailment relation between formulas of $\mathcal{COL}$. We define this relation as follows:

Definition 5.18

Let $\varphi \in \mathcal{COL}$ be a formula, and $\Gamma \subseteq \mathcal{COL}$ be a set of formulas. We write $\Gamma \models \varphi$ if $\varphi^{\mathcal{K}^*,w} = \mathbf{t}$ for every world w such that $\psi^{\mathcal{K}^*,w} = \mathbf{t}$ for all $\psi \in \Gamma$.

Note that this definition of the entailment relation does not coincide with the standard way of defining the entailment with respect to a Kripke semantics, as in our case we fix a canonical Kripke structure rather than quantifying over all Kripke structures. The fact that this entailment relation behaves in a sensible way is captured by the properties listed in the following theorem:

Theorem 5.5

Let $\varphi, \psi \in \mathcal{COL}$ two formulas, $\Gamma, \Gamma' \subset \mathcal{COL}$ two sets of formulas, $\mathcal{A} \in \mathcal{A}$ an agent, and $G \subseteq \mathcal{A}$ a non-empty set of agents. Then, the following properties hold:

1. (Prop) For each propositional tautology φ, we have $\models \varphi$.
2. (MP) $\varphi, \varphi \Rightarrow \psi \models \psi$.
3. (Mono) If $\Gamma \models \varphi$, then $\Gamma, \psi \models \varphi$.
4. (Cut) If $\Gamma \models \varphi$ and $\Gamma', \varphi \models \psi$, then $\Gamma, \Gamma' \models \psi$.
5. (K) $\models (K_{\mathcal{A}}(\varphi \Rightarrow \psi) \wedge K_{\mathcal{A}}\varphi) \Rightarrow K_{\mathcal{A}}\psi$.
6. (Nec) If $\models \varphi$, then $\models K_{\mathcal{A}}\varphi$.
7. (M) If $\varphi \not\models \psi$, then $M_{\mathcal{A}}\varphi \models \neg K_{\mathcal{A}}\psi$.
8. (O) $O_{\mathcal{A}}\varphi \not\models \bot$.
9. (Fixed point axiom) $\models C_G\varphi \Leftrightarrow E_G(\varphi \wedge C_G\varphi)$.
10. (Induction rule) If $\varphi \models E_G(\varphi \wedge \psi)$, then $\varphi \models C_G\psi$.

Proof. 1. Let φ be any propositional tautology with atoms $P_1, \ldots, P_k$, and $\xi_1, \ldots, \xi_k$ be formulas in $\mathcal{COL}$. For every $i \in [1,k]$, we can substitute P_i with ξ_i in φ to obtain φ_ξ. Let w be a world. Since for every formula $\psi \in \mathcal{COL}$ the evaluation $\psi^{\mathcal{K}^*,w}$ cannot be undefined, for every $i \in [1,k]$, we can assign to P_i the truth value given by $\xi_i^{\mathcal{K}^*,w}$. Since φ is a propositional tautology, it must be true with the above truth assignments. Hence, $\varphi_\xi^{\mathcal{K}^*,w} = \mathbf{t}$ by the definition of the evaluation in the Kripke structure.

2. Let $w \in \mathfrak{U}$ such that $\varphi^{\mathcal{K}^*,w} = \mathbf{t}$ and $(\varphi \Rightarrow \psi)^{\mathcal{K}^*,w} = \mathbf{t}$. Notice that $(\varphi \Rightarrow \psi)^{\mathcal{K}^*,w} = \left((\varphi \wedge \neg\psi)^{\mathcal{K}^*,w}\right)^{-1}$. Hence, $\mathrm{glb}_{\leq_{tr}}(\{\varphi^{\mathcal{K}^*,w}, (\neg\psi)^{\mathcal{K}^*,w}\}) = \mathbf{f}$. Since $\varphi^{\mathcal{K}^*,w} = \mathbf{t}$, we have $(\psi^{\mathcal{K}^*,w})^{-1} = (\neg\psi)^{\mathcal{K}^*,w} = \mathbf{f}$, i.e., $\psi^{\mathcal{K}^*,w} = \mathbf{t}$, as desired.

3. Clear by the definition of logical consequence relation.

4. Let $w \in \mathfrak{U}$ such that $\xi^{\mathcal{K}^*,w} = \mathbf{t}$ for all $\xi \in \Gamma \cup \Gamma'$. Since $\Gamma \models \varphi$, we have $\varphi^{\mathcal{K}^*,w} = \mathbf{t}$. Then, $\psi^{\mathcal{K}^*,w} = \mathbf{t}$ because $\Gamma', \varphi \models \psi$.

5. Let $w \in \mathfrak{U}$. We want to show that $((K_{\mathcal{A}}(\varphi \Rightarrow \psi) \wedge K_{\mathcal{A}}\varphi) \Rightarrow K_{\mathcal{A}}\psi)^{\mathcal{K}^*,w} = \mathbf{t}$. Notice that

$$\begin{aligned} V \ :&= ((K_{\mathcal{A}}(\varphi \Rightarrow \psi) \wedge K_{\mathcal{A}}\varphi) \Rightarrow K_{\mathcal{A}}\psi)^{\mathcal{K}^*,w} \\ &= (\neg((K_{\mathcal{A}}(\neg(\varphi \wedge \neg\psi)) \wedge K_{\mathcal{A}}\varphi) \wedge \neg K_{\mathcal{A}}\psi))^{\mathcal{K}^*,w} \\ &= \left(\mathrm{glb}_{\leq_{tr}}(\{S, ((K_{\mathcal{A}}\psi)^{\mathcal{K}^*,w})^{-1}\})\right)^{-1}, \end{aligned}$$

where $S := \mathrm{glb}_{\leq_{tr}}(\{(K_{\mathcal{A}}(\neg(\varphi \wedge \neg\psi)))^{\mathcal{K}^*,w}, (K_{\mathcal{A}}\varphi)^{\mathcal{K}^*,w}\})$. If $(K_{\mathcal{A}}\psi)^{\mathcal{K}^*,w} = \mathbf{t}$, then $V = \mathbf{t}$, as desired. Now, we assume $(K_{\mathcal{A}}\psi)^{\mathcal{K}^*,w} = \mathbf{f}$. Observe that V reduces to just $V = \left(\mathrm{glb}_{\leq_{tr}}(\{(K_{\mathcal{A}}(\neg(\varphi \wedge \neg\psi)))^{\mathcal{K}^*,w}, (K_{\mathcal{A}}\varphi)^{\mathcal{K}^*,w}\})\right)^{-1}$. If $(K_{\mathcal{A}}\varphi)^{\mathcal{K}^*,w} = \mathbf{f}$, then again $V = \mathbf{t}$, as desired. So, we suppose $(K_{\mathcal{A}}\varphi)^{\mathcal{K}^*,w} = \mathbf{t}$, i.e., $\varphi^{\mathcal{K}^*,v} = \mathbf{t}$ for all $v \in R_{\mathcal{A}}^w$. Hence, we have that

$$\begin{aligned} (K_{\mathcal{A}}(\neg(\varphi \wedge \neg\psi)))^{\mathcal{K}^*,w} &= \{(\neg(\varphi \wedge \neg\psi))^{\mathcal{K}^*,v} \mid v \in R_{\mathcal{A}}^w\} \\ &= \{\mathrm{glb}_{\leq_{tr}}(\{\varphi^{\mathcal{K}^*,v}, (\psi^{\mathcal{K}^*,v})^{-1}\})^{-1} \mid v \in R_{\mathcal{A}}^w\} \\ &= \{((\psi^{\mathcal{K}^*,v})^{-1})^{-1} \mid v \in R_{\mathcal{A}}^w\} = \{\psi^{\mathcal{K}^*,v} \mid v \in R_{\mathcal{A}}^w\} \\ &= (K_{\mathcal{A}}\psi)^{\mathcal{K}^*,w} = \mathbf{f}, \end{aligned}$$

which implies $V = \mathbf{t}$, as desired.

6. Let $w \in \mathfrak{U}$. By definition, $(K_{\mathcal{A}}\varphi)^{\mathcal{K}^*,w} = \mathrm{glb}_{\leq_{tr}}(\{\varphi^{\mathcal{K}^*,w'} | w' \in R_{\mathcal{A}}^w\})$. Since $\varphi^{\mathcal{K}^*,w'} = \mathbf{t}$ for any $w' \in \mathfrak{U}$ by hypothesis, we have $(K_{\mathcal{A}}\varphi)^{\mathcal{K}^*,w} = \mathbf{t}$, as desired.

7. Since $\varphi \nvdash \psi$, there exists $w^* \in \mathfrak{U}$ such that $\varphi^{\mathcal{K}^*,w^*} = \mathbf{t}$ and $\psi^{\mathcal{K}^*,w^*} = \mathbf{f}$. Let $w \in \mathfrak{U}$ be such that $(M_{\mathcal{A}}\varphi)^{\mathcal{K}^*,w} = \mathbf{t}$. Then, $R_{\mathcal{A}}^w \supseteq \{w \in \mathfrak{U} \mid \varphi^{\mathcal{K}^*,w} = \mathbf{t}\}$, and in particular $w^* \in R_{\mathcal{A}}^w$. Hence, $(K_{\mathcal{A}}\psi)^{\mathcal{K}^*,w} = \mathrm{glb}_{\leq_{tr}}(\{\psi^{\mathcal{K}^*,w'} \mid w' \in R_{\mathcal{A}}^w\}) = \mathbf{f}$.

8. By Theorem 5.4, there exists a world w such that $(O_{\mathcal{A}}\varphi)^{\mathcal{K}^*,w} = \mathbf{t}$. Since $\bot^{\mathcal{K}^*,w} = \mathbf{f}$, we have $O_{\mathcal{A}}\varphi \nvdash \bot$.

9. Let $w \in \mathfrak{U}$. By Definition 5.14, we have the following double implications:

$$
\begin{aligned}
&(E_G(\varphi \wedge C_G\varphi))^{\mathcal{K}^*,w} = \mathbf{t} \\
&\quad \Leftrightarrow \forall \mathcal{A} \in G,\ (K_{\mathcal{A}}(\varphi \wedge C_G\varphi))^{\mathcal{K}^*,w} = \mathbf{t} \\
&\quad \Leftrightarrow \forall \mathcal{A} \in G,\ \forall w' \in \mathcal{A}{\uparrow}w,\ (\varphi \wedge C_G\varphi)^{\mathcal{K}^*,w'} = \mathbf{t} \\
&\quad \Leftrightarrow \forall \mathcal{A} \in G,\ \forall w' \in \mathcal{A}{\uparrow}w,\ \varphi^{\mathcal{K}^*,w'} = \mathbf{t} \text{ and } (C_G\varphi)^{\mathcal{K}^*,w'} = \mathbf{t} \\
&\quad \Leftrightarrow \forall \mathcal{A} \in G,\ \forall w' \in \mathcal{A}{\uparrow}w,\ \forall k \geq 1,\ (K_{\mathcal{A}}\varphi)^{\mathcal{K}^*,w'} = \mathbf{t} \text{ and } (E_G^k\varphi)^{\mathcal{K}^*,w'} = \mathbf{t} \\
&\quad \Leftrightarrow \forall \mathcal{A} \in G,\ \forall k \geq 2,\ (E_G\varphi)^{\mathcal{K}^*,w} = \mathbf{t} \text{ and } (K_{\mathcal{A}}(E_G^{k-1}\varphi))^{\mathcal{K}^*,w} = \mathbf{t} \\
&\quad \Leftrightarrow \forall k \geq 2,\ (E_G\varphi)^{\mathcal{K}^*,w} = \mathbf{t},\ \text{and } (E_G^{k-1}\varphi)^{\mathcal{K}^*,w} = \mathbf{t} \\
&\quad \Leftrightarrow (C_G\varphi)^{\mathcal{K}^*,w} = \mathbf{t}.
\end{aligned}
$$

10. Let $w \in \mathfrak{U}$ such that $\varphi^{\mathcal{K}^*,w} = \mathbf{t}$. We say that a world $v \in \mathfrak{U}$ is 1-*reachable from* w if $v \in R^w_{\mathcal{A}}$ for some $\mathcal{A} \in G$. For $k > 1$, we say that a world $v \in \mathfrak{U}$ is k+1-*reachable from* w if v is 1-reachable from a world v' that is k-reachable from w.

    We show by induction on k that for all $k \in \mathbb{N}_{\geq 1}$, for all k-reachable worlds v from w, $(\psi \wedge \varphi)^{\mathcal{K}^*,v} = \mathbf{t}$. In particular, this proves that $E_G^k(\psi)^{\mathcal{K}^*,w} = \mathbf{t}$ for all $k > 0$, i.e., $(C_G\psi)^{\mathcal{K}^*,w} = \mathbf{t}$.

    - $k = 1$. Let v be a world 1-reachable from w. By hypothesis, $\varphi^{\mathcal{K}^*,w} = \mathbf{t}$ and $\varphi \vDash E_G(\varphi \wedge \psi)$, thus $(E_G(\varphi \wedge \psi))^{\mathcal{K}^*,w} = \mathbf{t}$. By definition of E_G and of 1-reachability, we must have $(\varphi \wedge \psi)^{\mathcal{K}^*,v} = \mathbf{t}$, as desired.
    - $k = k' + 1$. Let v be a world k-reachable from w. By definition of k'-reachability, there exists a world v' k'-reachable from w such that v is 1-reachable from v'. By induction hypothesis, $(\varphi \wedge \psi)^{\mathcal{K}^*,v'} = \mathbf{t}$, and $\varphi^{\mathcal{K}^*,v'} = \mathbf{t}$ in particular. Thus, by hypothesis, we get $(E_G(\varphi \wedge \psi))^{\mathcal{K}^*,v'} = \mathbf{t}$. By the definition of the valuation of the operator E_G, it follows that $(\varphi \wedge \psi)^{\mathcal{K}^*,v} = \mathbf{t}$, as desired.

□

Properties 1, 5, 6, 7, and 8 of Theorem 5.5 ensure that the semantics properly captures the intended meaning of the only-knowing operator. In more detail, properties 1, 5, 6, and 7 imply that for every φ and ψ, either $O_{\mathcal{A}}\varphi \Rightarrow K_{\mathcal{A}}\psi$ or $O_{\mathcal{A}}\varphi \Rightarrow \neg K_{\mathcal{A}}\psi$. More specifically, the former holds when ψ is entailed by φ, the latter otherwise. This means that $O_{\mathcal{A}}\varphi$ completely determines the agent's knowledge. Property 8 states that for any formula $\varphi \in \mathcal{COL}$, there is a world in which $O_{\mathcal{A}}\varphi$ holds, i.e. it is possible that an agent knows φ and knows nothing beyond φ.

The following example discusses the construction of a world w that shows that $O_{\mathcal{A}}\neg C_G p$ is satisfiable (in line with item 8 of Theorem 5.5), something that previous attempts

at combining only knowing and common knowledge failed at, even though it should intuitively be the case.

Example 5.3

Consider the setting of Example 5.1. We want to construct a world w that satisfies $O_{\mathcal{A}}\neg C_G p$, where $G = \{\mathcal{A}\}$. By the definition of the only-knowing operator, we must have

$$\mathcal{A}^w = \{w' \in \mathcal{W}^{\omega^2} \mid (C_G p)^{w'} = \mathbf{f}\}$$
$$\bar{\mathcal{A}}^w = \{w' \in \mathcal{W}^{\omega^2} \mid (C_G p)^{w'} = \mathbf{t}\}.$$

We first have to find the above sets.

Notice that, by Theorem 5.2, we have $\mathcal{A}^w \cup \bar{\mathcal{A}}^w = \mathcal{W}^{\omega^2}$, as supposed. By Proposition 5.6, Theorem 5.2, and the fact that any ω^2-biworld is the extension of some ω-biworld, we can reduce to finding the set of ω-biworlds satisfying $C_G p$. It is not hard to see that v from Example 5.1 satisfies $C_G p$. Moreover, since v is completed, by Corollary 5.1, v has a unique extension v' to depth ω^2. In an analogous fashion, we can build another ω^2-biworld u' satisfying $C_G p$ by considering the unique extension of a completed ω-biworld $u := (u_\alpha)_{\alpha<\omega}$ defined as

$$u_\alpha := \begin{cases} \{\emptyset\} & \text{if } \alpha = 0 \\ (u_0, \{v_{\alpha'}\}, \mathcal{W}^{\alpha'}) & \text{if } \alpha = \alpha' + 1 \end{cases}$$

Intuitively, both v' and u' are worlds in which p is common knowledge (for the only agent $\mathcal{A}$), but p is true in the objective world of v' and false in the one of u'. In particular, in u' the agent $\mathcal{A}$ is not truthful.

We claim[a] v and u are the only ω-biworlds satisfying $C_G p$. Hence, $\mathcal{A}^w$ must be $\mathcal{W}^{\omega^2} \setminus \{v', u'\}$ and $\bar{\mathcal{A}}^w$ must be $\{v', u'\}$, and we define $w := (\{p\}, \mathcal{W}^{\omega^2} \setminus \{v', u'\}, \{v', u'\})$.

Notice that by Theorem 5.4, w is unique up to a change of the objective world. In other words, the world defined as w but with $\{\emptyset\}$ as objective world w_0 satisfies $O_{\mathcal{A}}\neg C_G p$ too.

[a]The claim can be proven by induction, and it can already be seen to hold true by writing down all the eighteen 1-biworlds, and reasoning on the conditions that make $C_G p$ satisfied. We omit this reflection for the sake of conciseness.

5 Truthfulness and Introspection

In the previous three sections we have defined and described the construction of a structure of worlds that is rich enough to allow to formally define the semantics of only knowing and common knowledge in a way that matches basic intuitions about these logical modalities in a precisely specified way. A major challenge in designing this construction was to ensure that we have enough worlds to describe all logically possible epistemic states. For this reason, we decided to keep the construction as general as possible, i.e., not to unnecessarily limit the set of worlds.

However, there are certain properties of the knowledge modality that are often taken for granted in epistemic logic and that require limiting the set of worlds. In particular, the following properties are often assumed to hold:

- **Truthfulness:** $K_{\mathcal{A}}\varphi \Rightarrow \varphi$ is satisfied in every world for every agent $\mathcal{A}$.
- **Positive introspection:** $K_{\mathcal{A}}\varphi \Rightarrow K_{\mathcal{A}}K_{\mathcal{A}}\varphi$ is satisfied in every world for every agent $\mathcal{A}$.
- **Negative introspection:** $\neg K_{\mathcal{A}}\varphi \Rightarrow K_{\mathcal{A}}\neg K_{\mathcal{A}}\varphi$ is satisfied in every world for every agent $\mathcal{A}$.

While all three of these properties are commonly assumed in epistemic logic, there are specific issues about ensuring the first or the third in a logic with a modality $O_{\mathcal{A}}\varphi$ for only knowing or a modality $M_{\mathcal{A}}\varphi$ for knowing at most.

In the case of truthfulness, there is an issue concerning the formula $O_{\mathcal{A}}(K_{\mathcal{A}}p \vee K_{\mathcal{A}}q)$. Given the definition of the modality $O_{\mathcal{A}}$, this entails $K_{\mathcal{A}}(K_{\mathcal{A}}p \vee K_{\mathcal{A}}q)$, which by truthfulness entails $K_{\mathcal{A}}p \vee K_{\mathcal{A}}q$, so either $K_{\mathcal{A}}p$ or $K_{\mathcal{A}}q$ has to be true. But since $\mathcal{A}$ only knows $K_{\mathcal{A}}p \vee K_{\mathcal{A}}q$ and since $K_{\mathcal{A}}p \vee K_{\mathcal{A}}q$ entails neither p nor q, neither p nor q can be known, a contradiction. Thus $O_{\mathcal{A}}(K_{\mathcal{A}}p \vee K_{\mathcal{A}}q)$ cannot be satisfiable in a logic with truthfulness, which means that principle (O) from Theorem 5.5 cannot hold in such a logic (assuming principles (Prop), (Cut) and (M) do hold).

In the case of negative introspection, a more severe problem arises. Suppose $M_{\mathcal{A}}q$ is true in some world w. Since q does not entail p, by (M) this should entail that $\neg K_{\mathcal{A}}p$ is true in w, so by negative introspection, $K_{\mathcal{A}}\neg K_{\mathcal{A}}p$ is true in w. But since q does not entail $\neg K_{\mathcal{A}}p$, principle (M) also allows us to conclude that $\neg K_{\mathcal{A}}\neg K_{\mathcal{A}}p$ is true in w, a contradiction. Thus $M_{\mathcal{A}}q$ is not satisfiable, and similarly no formula of the form $M_{\mathcal{A}}\varphi$ or $O_{\mathcal{A}}\varphi$ is satisfiable for any satisfiable φ. It should be stressed that this problem is not caused by our semantic approach to only knowing, but is a direct consequence of basic properties that only knowing has been assumed to satisfy also in other papers.

Remark 5.1

A reader familiar with the literature on only knowing, of which we give an overview in Section 7, may wonder why this problem was not identified in previous papers, e.g. [Belle and Lakemeyer, 2015b], who define a semantics for only knowing in a logic with negative introspection. What the authors did not realize is that by enforcing negative introspection in their logic, they actually made all statements of the form $O_A\varphi$ unsatisfiable (for any satisfiable formula φ). They wrongly claim on page 5 that $O_i(p \wedge Cp)$ is satisfiable, but the alleged proof is wrong. If one defines $V^1 = \{w \mid p, q \in w, w \in \mathcal{W}\}$, $V^{k+1} = \{(w, V^k, \ldots, V^k) \mid w \in V^1\}$, $f'(i,k) = V^k$ and $w' = \{p\}$, then one can easily see that $f' \notin f_i^{w'}$ and $f', w' \models p \wedge Cp$, contradicting their claim that $f, w \models O_i(p \wedge Cp)$.

In order to avoid this problem, one would need to make use of autoepistemic logic [Moore, 1985], or some multi-agent version thereof [Lakemeyer, 1993; Permpoontanalarp and Jiang, 1995; Vlaeminck et al., 2012; Van Hertum et al., 2016] in the definition of the semantics of $M_A\varphi$: intuitively, $M_A\varphi$ should be true in a world in which all worlds that satisfy all formulas entailed by the autoepistemic theory $\{\varphi\}$ are accessible. This would give rise to a logic in which a variant of principle (M) with a negated autoepistemic entailment in the place of the negated entailment is satisfied. We leave it to future work to develop the details of such a theory and investigate whether it behaves as intended.

Given that it is somewhat problematic to incorporate truthfulness and negative introspection in a logic with only knowing, whereas no similar problems arise for positive introspection, we describe how the semantic framework from the previous sections can be used to define a logic of only knowing and common knowledge in which positive introspection is ensured.

Definition 5.19

A world w is called *positively introspective (PI)* if for each $\mathcal{A} \in \boldsymbol{\mathcal{A}}$ and for any worlds w', w'', $w'' R_\mathcal{A} w' R_\mathcal{A} w$, implies $w'' R_\mathcal{A} w$. A world w is called *recursively PI* if w is positively introspective and all the worlds that are reachable from w through the union $\bigcup_{\mathcal{A}\in\boldsymbol{\mathcal{A}}} R_\mathcal{A}$ of all accessibility relations are positively introspective.

We now define $\boldsymbol{\mathcal{K}}^*_{PI} := (\mathfrak{U}^{PI}, (R^{PI}_\mathcal{A})_{\mathcal{A}\in\boldsymbol{\mathcal{A}}})$ to be the Kripke substructure of $\boldsymbol{\mathcal{K}}^*$, where the underlying world set $\mathfrak{U}^{PI}$ is the set of all recursively PI worlds, and the accessibility relations are just the ones coming from $\boldsymbol{\mathcal{K}}^*_{PI}$ being a substructure, i.e., for all $\mathcal{A} \in \boldsymbol{\mathcal{A}}$, $R^{PI}_\mathcal{A} := R_\mathcal{A} \cap (\mathfrak{U}^{PI} \times \mathfrak{U}^{PI})$.

Now, we can define a modified entailment relation that takes into account positive introspection:

Definition 5.20

For $\varphi \in \mathcal{COL}$ and $\Gamma \subseteq \mathcal{COL}$, we write $\Gamma \models_{PI} \varphi$ if $\varphi^{\mathcal{K}^*_{PI},w} = \mathbf{t}$ for every world w such that $\psi^{\mathcal{K}^*_{PI},w} = \mathbf{t}$ for all $\psi \in \Gamma$.

Note that incorporating positive introspection into the logic in this way modifies the meaning of the modalities $M_\mathcal{A}$ and $O_\mathcal{A}$, because knowing at most φ means that any ψ that one knows must be entailed by φ and the entailment relation is different now that truthfulness and introspection are hard-coded into the logic: ψ may be entailed by φ together with truthfulness and introspection even though it was not entailed by φ itself in the original semantics of $\mathcal{COL}$ without truthfulness and introspection. In other words, when $\varphi \models_{PI} \psi$ but $\varphi \not\models \psi$, then $M_\mathcal{A}\varphi \not\models_{PI} \neg K_\mathcal{A}\varphi$ even though $M_\mathcal{A}\varphi \models \neg K_\mathcal{A}\varphi$.

The following theorem establishes that positive introspection does indeed hold in this logic and that the properties that we established for $\models$ in Theorem 5.5 also hold for $\models_{PI}$.

Theorem 5.6

Let $\varphi, \psi \in \mathcal{COL}$ be two formulas, $\Gamma, \Gamma' \subset \mathcal{COL}$ two sets of formulas, $\mathcal{A} \in \boldsymbol{\mathcal{A}}$ an agent, and $G \subseteq \boldsymbol{\mathcal{A}}$ a non-empty set of agents. Then, the following properties hold:

1-10. All properties mentioned in Theorem 5.5 with $\models$ replaced by $\models_{PI}$.

11. (PI) $\models_{PI} K_\mathcal{A}\varphi \Rightarrow K_\mathcal{A}K_\mathcal{A}\varphi$

Proof. All properties apart from (PI) and (O) can be established in a way analogous to the proof of Theorem 5.5.

- (PI) Suppose w is positively introspective and suppose $(K_\mathcal{A}\varphi)^{\mathcal{K}^*_{PI},w} = \mathbf{t}$. Suppose $w''R_\mathcal{A}w'R_\mathcal{A}w$. Since w is positively introspective, it follows that $w''R_\mathcal{A}w$, i.e., that $\varphi^{\mathcal{K}^*_{PI},w''} = \mathbf{t}$. Since w'' was arbitrary, we can conclude that $(K_\mathcal{A}\varphi)^{\mathcal{K}^*_{PI},w'} = \mathbf{t}$. Since w' was arbitrary, this in turn implies that $(K_\mathcal{A}K_\mathcal{A}\varphi)^{\mathcal{K}^*_{PI},w} = \mathbf{t}$, as required.
- (O') For establishing this result, we need a result similar to the first statement of Theorem 5.4: For every agent $\mathcal{A}$ and every formula $\varphi \in \mathcal{COL}$, there is a world $w \in \mathfrak{U}^{PI}$ such that $(O_\mathcal{A}\varphi)^{\mathcal{K}^*_{PI},w} = \mathbf{t}$.
  Let v be an arbitrary 0-biworld.

$$
\begin{aligned}
W &:= \{u|_{\omega^2} \mid u \in \mathfrak{U}^{PI} \text{ and } (\varphi \wedge K_\mathcal{A}\varphi)^{\mathcal{K}^*,u} = \mathbf{t}\},\\
\overline{W} &:= \mathcal{W}^{\omega^2} \setminus W,\\
w &:= (v, (W)_{\mathcal{A}\in\boldsymbol{\mathcal{A}}}, (\overline{W})_{\mathcal{A}\in\boldsymbol{\mathcal{A}}}).
\end{aligned}
$$

> Clearly w is a world. In order to show that $w \in \mathfrak{U}^{PI}$, we need to show that w is recursively PI. From the definition of w it is easy to see that for this, it is enough to show that w is positively introspective.
> So suppose $\mathcal{A} \in \boldsymbol{\mathcal{A}}$ and $w'' R_{\mathcal{A}} w' R_{\mathcal{A}} w$. We need to show that $w'' R_{\mathcal{A}} w$. The fact that $w' R_{\mathcal{A}} w$ means that $w'|_{\omega^2} \in W$, i.e., $(K_{\mathcal{A}}\varphi)^{\mathcal{K}^*_{PI},w'} = \mathbf{t}$. Since $w'' R_{\mathcal{A}} w'$, it follows that $\varphi^{\mathcal{K}^*_{PI},w''} = \mathbf{t}$. Since w' is positively introspective, the fact that $(K_{\mathcal{A}}\varphi)^{\mathcal{K}^*_{PI},w'} = \mathbf{t}$ implies that $(K_{\mathcal{A}}K_{\mathcal{A}}\varphi)^{\mathcal{K}^*_{PI},w'} = \mathbf{t}$, i.e., that $(K_{\mathcal{A}}\varphi)^{\mathcal{K}^*_{PI},w''} = \mathbf{t}$. So $(\varphi \wedge K_{\mathcal{A}}\varphi)^{\mathcal{K}^*_{PI},w''} = \mathbf{t}$, i.e., $w''|_{\omega^2} \in W = \mathcal{A}^w$, i.e., $w'' R_{\mathcal{A}} w$, as required.
> It follows directly from the definition of w that $(K_{\mathcal{A}}\varphi)^{\mathcal{K}^*_{PI},v'} = \mathbf{t}$ and that $(M_{\mathcal{A}}\varphi)^{\mathcal{K}^*_{PI},v'} = \mathbf{t}$, i.e., that $(O_{\mathcal{A}}\varphi)^{\mathcal{K}^*_{PI},v'} = \mathbf{t}$.
>
> □

Given that we could develop a theory with positive introspection, one may wonder what happens if one tries to similarly add negative introspection and/or truthfulness. Due to the problems with negative introspection described above, naively adding negative introspection in this way will yield a logic in which $M_{\mathcal{A}}\varphi$ is not satisfiable for any φ. Adding truthfulness, on the other hand, does not cause such problems. Indeed, truthfulness and positive introspection can meaningfully be added together in a way similar to how positive introspection was added in the above definitions, yielding an entailment relation $\models_{TPI}$. Apart from (O), the properties of Theorem 5.6 will still hold. We conjecture that the following weakening of (O') holds in this context:

(O') If φ is a formula not involving a modality with subscript $\mathcal{A}$, then $O_{\mathcal{A}}\varphi \not\models_{TPI} \bot$.

The proof of this conjecture is left to future work.

6 An Open Challenge: Common Only Knowledge

In this section, we focus our attention on a type of knowledge that we named *common only knowledge* and provide some insight about its nature and why it appears so challenging to precisely express it.

In certain situations, we may want to represent this peculiar type of knowledge, in which common knowledge and only knowing are tightly entwined. This is often the case for epistemic puzzles in which all the agents have common knowledge of a set of rules and know *nothing more* than that. The phrasing "nothing more" in this situation is not precise, as it does not exactly correspond to the only-knowing operator that we have defined, but rather to a more subtle and complicated form of knowledge. Let us better explain this with a concrete example: Cheryl's birthday puzzle.

Example 5.4: Cheryl's Birthday Puzzle

Albert and Bernard just became friends with Cheryl, and they want to know when her birthday is. Cheryl gives them a list of 10 possible dates:

- May 15, May 16, May 19,
- June 17, June 18,
- July 14, July 16,
- August 14, August 15, August 17.

Then, Cheryl tells Albert the month of her birthday (without Bernard listening) and tells Bernard the day (without Albert listening). The following dialogue between Albert and Bernard unfolds:

1. Albert: "*I don't know when Cheryl's birthday is, but I know that Bernard doesn't know either.*"
2. Bernard: "*At first I didn't know when Cheryl's birthday is, but I know now.*"
3. Albert: "*Then I also know when Cheryl's birthday is.*"

When is Cheryl's birthday?

The puzzle above is not hard to solve: the fact that certain possible dates share the same month or day makes it possible to remove a few candidates at each statement made by Albert or Bernard, until only one date remains. We invite the reader to give it a go for a fun break. A detailed solution is provided at the end of this section.

So what is the issue in Cheryl's birthday puzzle, one may wonder. Even though the solution can be found, exactly expressing the knowledge of Albert (A) and Bernard (B) at each step of the puzzle is extremely complicated, even in natural language. For instance, let us consider the initial situation, just before A's first statement[4]. A first attempt at expressing the knowledge of A and B may be: A and B only know that it is common knowledge between the two of them that: a) A knows the month of Cheryl's birthday, b) B knows the day, and c) A and B know the ten possible dates. However, this is not precise, as both A and B know more than just the fact that a), b), and c) are common knowledge. For instance, A knows that B does not know more than the fact that a), b), and c) are common knowledge. This is again not correct, as we could construct a similar

[4]We make some basic assumptions on the agents A and B. For example, we assume that A and B are perfect reasoners, do not lie, and have perfect memory, and that this is common knowledge between them. Moreover, we assume it is common knowledge between A and B how a calendar works. For the sake of simplicity, we omit these assumptions when expressing the knowledge of A and B.

statement for B by swapping A and B, showing that indeed B knows more than the fact that a), b), and c) are common knowledge (and A knows that!). What we are trying to express here is a sort of *common only knowledge* that ties together the concepts of only knowing and common knowledge: the setup of the puzzle is common knowledge between A and B, and the fact that they do not have any additional information is also common knowledge. Notice that this cannot be readily encoded using the only-knowing operator: if we substituted the sentence "they do not have any additional information" with a conjunction of only-knowing operators, we would obtain a formula of the form $C_{\{A,B\}}\left(O_A C_{\{A,B\}}(a \wedge b \wedge c) \wedge O_B C_{\{A,B\}}(a \wedge b \wedge c)\right)$, which is unsatisfiable if we assume both A and B are truthful. In other words, we want to express the fact that the knowledge of A and B is *closed*, in the sense that they only know what can be derived from a certain set of facts, and they are aware of such closure. However, the fact that they know their knowledge is closed plays a role in the closure itself.

Despite its recursive nature, which makes its formal encoding and deep understanding rather complicated, this type of information about the agents' knowledge is something completely natural that we implicitly assume in many epistemic puzzles, like the Cheryl's birthday one. We believe that the framework we built with the notion of biworlds is sufficiently refined to capture *common only knowledge*. Nevertheless, finding a theory precisely expressing common only knowledge appears to be very demanding. Hence, we envision two possible future directions, equally challenging and stimulating:

- Defining a new modal operator for common only knowledge to further enrich $\mathcal{COL}$;
- Building a new framework that allows for recursive theories and in which each theory is paired with a *generator*, providing a recipe to construct the possible worlds and the accessibility relations.

The Solution to Cheryl's Birthday Puzzle

In order to solve Cheryl's birthday puzzle, one has to reason about the set

$$\mathsf{Dates} := \{(15,5),(16,5),(19,5),(17,6),(17,6),(14,7),(16,7),(14,8),(15,8),(17,8)\}$$

of possible dates, and exclude the ones which are no longer possible after every statement made by either Albert (A) or Bernard (B). Notice that, at each stage of the puzzle, if we fix a date, i.e., an objective world, the knowledge of A and B is fully determined. Hence, at the initial step of the puzzle there are only ten possible worlds, one for each possible date. In the following, for each $(d,m) \in \mathsf{Dates}$, we will denote by $w^{d/m}$ the world with objective world the one where only the date (d,m) is true. Drawing the graphs of the Kripke structure at each step of the puzzle is rather simple. Before the first statement of A, if we fix a date $(d,m) \in \mathsf{Dates}$, A considers possible all the dates from Dates with month m, and B all those with day d (Figure 5.3).

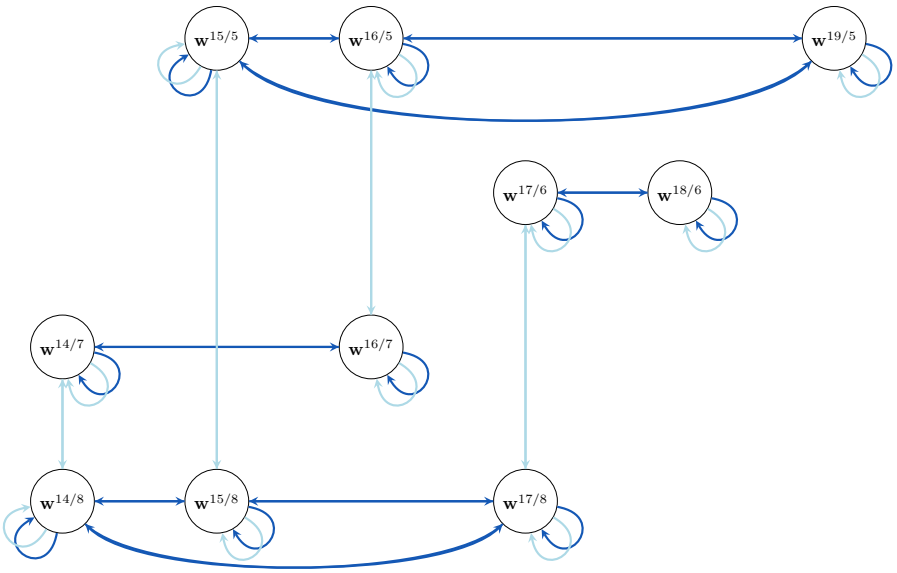


Figure 5.3: The Kripke structure befor A's first statement. The accessibility relation for A is represented by the blue arrows, and for B by the light blue arrows.

Then, A makes his first statement. The first half of it ("I don't know when Cheryl's birthday is") does not add any information: this is the case for any of the possible dates, as clearly represented in Figure 5.3. The second part of A statement ("I know that Bernard doesn't know either") is far more interesting. Since there is only one month with day 18 and one with day 19 in Dates, namely June and May, respectively, if Cheryl's birthday fell on either May 19 or June 18, B would know the full date. It follows that A knows that Cheryl's birthday does not fall on June 18 nor on May 19. Since Cheryl tells A only the month of her birthday, we can derive that the month is neither May nor June. Notice that all the information that we used to make the reasoning above is common knowledge between A and B; thus, the same derivations can be made by both A and B. Hence, the set of possible dates is now restricted to only five elements (Figure 5.4).

B proceeds to make his statement, saying that he now knows the exact date of Cheryl's birthday. From Figure 5.4, it is easy to see that if the day of Cheryl's birthday was the 14, then B would still consider both July 14 and August 14 as possible. Hence, we can eliminate both dates. The three dates remaining (July 17, August 15, and August 17) cannot be excluded yet: if one of these was Cheryl's birthday, B would know at this point.

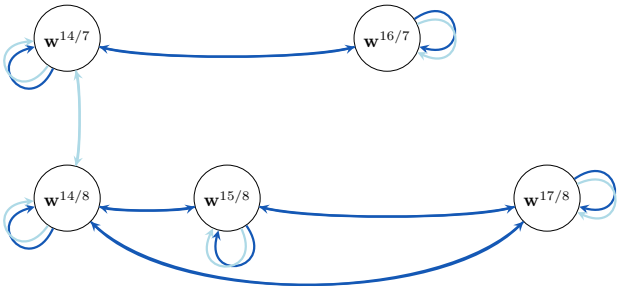


Figure 5.4: The Kripke structure after A's first statement. The accessibility relation for A is represented by the blue arrows, and for B by the light blue arrows.

This is again clear from the Kripke structure in Figure 5.4. It follows that we are left with only three options, as represented in Figure 5.5.

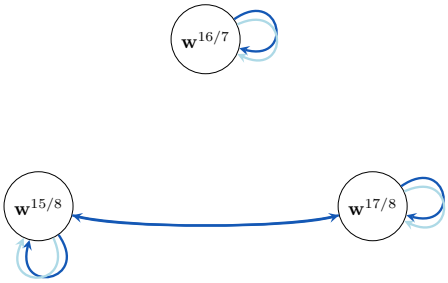


Figure 5.5: The Kripke structure right after B's statement. The accessibility relation for A is represented by the blue arrows, and for B by the light blue arrows.

Finally, A states that he knows the date of the birthday. From Figure 5.5, it is now obvious that Cheryl's birthday falls on July 16, which answers the question formulated in the puzzle.

7 Related Work

In this chapter, we have studied the interplay of common knowledge and only knowing. The former concept is quite well-known and has been extensively studied [Fagin et al., 1995; Meyer and van der Hoek, 1995] since its first mentions in the philosophical [Lewis, 1969], and the mathematical literature [Aumann, 1976]. The latter concept is younger and has been studied intensively more recently.

Levesque [1990] was among the first to introduce the notion of only knowing[5] by presenting a single-agent logic of belief extended with a novel operator O expressing that the agent's beliefs are exactly the ones implied by the knowledge base and nothing more. He intended his logic of only knowing to capture certain types of non-monotonic reasoning patterns, like autoepistemic logic (AEL) [Moore, 1985]. In the 1990s and early 2000s, single-agent only knowing was successfully studied and implemented [Halpern and Lakemeyer, 1995; Rosati, 2000; Levesque and Lakemeyer, 2000], and Lakemeyer and Levesque [2005] further revealed its potential by showing that it is also possible to capture DL [Reiter, 1980] and a variant of AEL proposed by Konolige [1988].

Halpern [1993] and Lakemeyer [1993] were among the first to extend Levesque's only knowing to a multi-agent setting. In a joint publication, they ([Halpern and Lakemeyer, 2001]) improved upon their independent works with an axiom system satisfying all the desired properties for only knowing. Nevertheless, the proposed axiomatization forces to include directly in the language a validity operator and the resulting semantics is not "as natural as we might like", according to the authors. In the early 2000s, first Waaler [2004] alone and then together with Solhaug [Waaler and Solhaug, 2005] tried another route to generalize Levesque's axioms, without encoding the notion of validity into the language itself. Yet, once again these models and the logic itself feel complex and unnatural.

An in-depth analysis of the issues of all the previously mentioned works on multi-agent only knowing is provided by Belle and Lakemeyer (2010; 2015b), together with a natural way to avoid such problems. Belle and Lakemeyer proposed to use models limited to a finite depth k, i.e., models at which beliefs can be nested at most k times, by introducing the concept of k-structure to represent an agent's epistemic state. In this way, they successfully extend Levesque's logic, by keeping the original idea of Levesque's worlds and generalizing its features at the same time. However, they do not take into account common knowledge.

Aucher and Belle [2015] proposed a novel formulation comprising both common knowledge and only knowing at level k. To achieve this result, they make use of the

[5] There are several closely-related notions, like ignorance [Halpern and Moses, 1984; Konolige, 1982], minimal knowledge [van der Hoek and Thijsse, 2002], and total knowledge [Pratt-Hartmann, 2000].

concept of so-called k+1-canonical formulas [Moss, 2007], written as conjunctions of a 0-canonical formula and k-canonical formulas nested in knowledge and common-knowledge operators. Even though the proposed pointed epistemic models can be fully characterized up to modal depth k by a k-canonical formula, such representation at depth k might feel slightly unnatural, as it characterizes the knowledge of an agent only up to level k, but it also determines the common knowledge of a group of agents, which is infinitary in nature. When disregarding the conjunction of the common-knowledge operators in such formulas, one can see a correspondence between our k-biworlds and the proposed k-canonical formulas. However, there is an important limitation to this work. Namely, while our $O_{\mathcal{A}}$ operator fully captures the knowledge and ignorance of an agent $\mathcal{A}$ at any depth, the $O^n_{\mathcal{A}}$ operator proposed by Aucher and Belle [2015] expresses only knowing for an agent $\mathcal{A}$ just up to level n in the sense that an expression of the form $O^1_{\mathcal{A}}\varphi$ should be read as "If I disregard all my knowledge deeper than level one, I only know φ". Formally, as Aucher and Belle pointed out, for any two modal depths $n > m$, if agent $\mathcal{A}$ only knows φ at depth n, then $\mathcal{A}$ only knows φ at depth m, i.e., $O^n_{\mathcal{A}} \rightarrow O^m_{\mathcal{A}}$, but the reverse implication might not hold.

The limitation of the only-knowing operator to a finite depth is overcome in the same year by Belle and Lakemeyer [2015a]. They devised an alternative approach to bring common knowledge and only knowing together, the semantic structures of which are close to our ω-biworlds (except that there is no representation of the set $\bar{\mathcal{A}}{\uparrow}\omega$). However, as explained in Remark 5.1, due to hard-coding negative introspection in their logic, all non-trivial formulas of the form $O_{\mathcal{A}}\varphi$ are unsatisifiable, which goes against what we expressed as Property 8 in Theorem 5.5. Hence, such a semantics does not properly capture the intended meaning of the only-knowing operator.

Van Hertum [2016] observed that in [Belle and Lakemeyer, 2015a], the formula $O_{\mathcal{A}}\neg C p$ cannot be satisfied, and this holds even if negative introspection is dropped (to avoid the problem mentioned in Remark 5.1). Van Hertum [2016] attempted to overcome this problem by proposing four different semantics, but he does not entirely fulfill his purpose. In more detail, two of the proposed semantics (Section 6.3 of [Van Hertum, 2016]) do not allow for arbitrary nesting of the only-knowing operator and thus they do not cover the whole $\mathcal{COL}$ language. A third semantics (Section 6.2.2 of [Van Hertum, 2016]) seems to solve the problem regarding the satisfiability of formulas like $O_{\mathcal{A}}\neg C p$, but it is not precision-monotonic. This is rather problematic: in a given μ-world in such semantics, $O_{\mathcal{A}}\varphi$ might be true, but if we give more precise information (extending it to depth $\mu + 1$), the formula might become false. Hence, one could only define what an agent only knows "up to a certain depth", as in [Aucher and Belle, 2015]. Finally, the semantics presented in Section 6.2.1 makes use of λ-canonical Kripke structures, for any fixed limit ordinal λ, to define a two-valued valuation for $\mathcal{COL}$. In particular, if $\lambda = \omega$, then the proposed semantics corresponds to the semantics of Belle and Lakemeyer [2015a] with some minor

adjustments. The hope of the author was that by choosing a large enough limit ordinal λ, the satisfiability issue would be solved. Unfortunately, there exist formulas that are not satisfiable in any λ-canonical Kripke structure, for example $O_A \neg K_A \bot$, which formalizes the statement "all $\mathcal{A}$ knows is that their knowledge is consistent". Even though this formula may seem like a corner case, all such corner cases need to be avoided in order to have a well-behaved entailment relation that satisfies the desirable properties listed in Theorem 5.5. The unsatisfiability of $O_A \neg K_A \bot$ implies that property (O) is not satisfied, which means that the semantics does not properly capture the intended meaning of the only-knowing operator.

8 Conclusion

We defined a multi-agent epistemic logic with common-knowledge and only-knowing operators, which successfully encodes these notions under the same framework.

First, we introduced the novel concept of μ-biworld for countable ordinals μ, which approximates not only the worlds that an agent deems possible, but also those deemed impossible. This duality proved to be fundamental to successfully deal with the only-knowing operator in a multi-agent setting. Moreover, we have shown that the proposed new definitions are indeed sensible, as they satisfy the properties one would expect (Theorem 5.1).

Second, we defined the language $\mathcal{COL}$, extending propositional logic with the modal operators K_A, M_A, E_G, and C_G, and a three-valued model semantics for it. Furthermore, we defined a canonical Kripke structure over completed ω^2+1-biworlds, and the two-valued semantics obtained from it is shown to coincide with the model semantics. This allowed us to prove several desirable properties (Theorem 5.5) the resulting logic satisfies. In particular, we showed that for any formula φ, there is a unique state-of-mind of a given agent in which they only know φ.

Then, we have considered how our framework can be extended to satisfy properties like truthfulness, positive introspection and negative introspection. For positive introspection we have shown some positive results, whereas for truthfulness and negative introspection, we have identified certain problems that arise when combining them with only knowing. We have motivated the need for further research related to these problems. Finally, in Section 6, we presented the open problem consituted by the encoding of a novel concept of knowledge in which common knowledge and only knowing are tightly entwined.

Another line of future work that we envisage is to generalize the construction of biworlds so that it becomes applicable to other areas of research. More concretely, this amounts to the construction of a set-theoretic universe in which there exists a universal set, similarly

as in topological set theory. If one takes this alternative set theory as one's metatheory, the incorrect definition from the introduction of this chapter could very easily be turned into a correct definition.

6 Conclusion

1 Contributions

This thesis advanced the use of approximation methods to define semantics across diverse fields, demonstrating their versatility and broad applicability. We achieved this through two main devices: approximators, within the framework of AFT, and biworlds, applied to multi-agent AEL. In both contexts, we utilized pairs of objects, like upper and lower bounds, to define approximate elements that encode uncertainty, a range of possibilities, or incongruency. Then, different types of constructions enabled the definition of the desired semantics.

In Chapter 3, we eased into the topic of AFT by presenting a novel member of the growing ranks of AFT applications, namely, DatalogMTL$^\neg$, which enriches Datalog with operators from metric temporal logic and (potentially unstratifiable) negation. Using the solid framework of AFT, we successfully defined four semantics for DatalogMTL$^\neg$: the Kripke-Kleene, well-founded, stable, and supported model semantics. While the stable model semantics had already been defined for DatalogMTL$^\neg$ [Walega et al., 2024] using equilibrium models from the HT logic, AFT retrieves the same stable models with a cleaner and (almost) effortless approach: the only action required is to define the correct *consistent approximator*, i.e., an operator on the space of partial, three-valued interpretations. Like for many other applications of AFT, this does not involve any far-fetched choice, as the

desired operator is the natural adaptation of Fitting's three-valued immediate consequence operator to the setting of DatalogMTL$^\neg$. Once this approximator is defined, we gain access to all the general theoretical results stemming from AFT. Moreover, the well-founded semantics obtained via AFT provides a novel perspective: DatalogMTL$^\neg$ can be viewed as a formal modelling of a particular class of inductive definitions with temporal conditions.

In Chapter 4, we ventured deeper into the realm of AFT by tackling a more fundamental question: how can we extend AFT to make it applicable to higher-order definitions? This fits into a broader effort towards the generalization of AFT [Denecker et al., 2004; Heyninck et al., 2024; Charalambidis et al., 2018; Charalambidis and Rondogiannis, 2023; Vanbesien, 2025]. We employed the concept of Cartesian closedness, drawn from the field of Category Theory (CT), to define the novel notions of *approximation category* and *approximation system*, which allowed us to inductively construct a hierarchy of approximation spaces for the semantics of each type of a given (higher-order) language. This resulted in a general, constructive, and systematic way to extend present (and possibly future) versions of AFT to the higher-order setting. The inductive nature and generality of the definition of approximation system makes it easily adaptable to different languages, types, and semantics, allowing for a wide range of applications. We have shown that, using this new framework, it is possible to generalise both standard AFT [Denecker et al., 2000], and the extension of consistent AFT proposed by Charalambidis et al. [2018]. For the latter, we have also defined a new approximator which resolves an issue concerning programs with existential quantifiers in the body of rules. Moreover, we have provided a notion of exactness for higher-order objects, together with a projection function, linking an exact element in the approximation space to the "real", non-approximated element it represents. These concepts were previously missing for higher-order objects, and they are indeed fundamental when dealing with approximations: they allow us to determine when an object, and in particular a model, is two-valued, and to retrieve the elements that are being approximated.

In Chapter 5, we changed scenario by moving to the field of multi-agent AEL. We focused on two key concepts concerning the knowledge of agents: only knowing, and common knowledge. While these two notions have been extensively studied separately, a logic joining them was still missing. Our main research question regarded the construction of a canonical Kripke structure for this logic. In particular, we were concerned with the definition of the correct set of possible worlds to successfully encode both common knowledge and only knowing under the same framework. We started by defining the novel concept of μ-biworld for countable ordinals μ, which approximates not only the worlds that an agent deems possible, but also those deemed impossible. Then, we proceeded to define the language $\mathcal{COL}$, which extends propositional logic with the modal operators $K_\mathcal{A}$, $M_\mathcal{A}$, E_G, and C_G. Using the biworlds, we were able to define a three-valued model semantics and a two-valued Kripke semantics, which coincide over completed ω^2+1-biworlds. All

the newly proposed definitions, which are highly non-trivial, proved to be sensible, as they satisfy the properties one would expect (Theorem 5.1 and Theorem 5.5). Finally, we investigated whether truthfulness and introspection can be considered in our framework. We have shown that both truthfulness and negative introspection clash with the concept of only knowing, while extending our logic of common knowledge and only knowing with positive introspection turned out to be possible.

2 Future Directions

We have already presented some ideas for future developments in the last section of each of the three main chapters of this thesis. Here, we further expand on a few possible research directions, concercerning both theory and applications.

A Gateway to Higher-Order Approximation Fixpoint Theory In Chapter 4, we have defined a general procedure to extend current (and possibly future) versions of AFT to higher-order definitions. In particular, in Sections 4 and 5 of Chapter 4, we have shown that both standard AFT [Denecker et al., 2000] and AFT from [Charalambidis et al., 2018], which extends consistent AFT [Denecker et al., 2004], fit into our novel framework. It would now be interesting to research which other AFT accounts can be extended to a higher-order setting in the same way. For instance, we could investigate *non-deterministic* AFT [Heyninck et al., 2024], which permits the handling of indefinite information, with applications in disjunctive logic programs [Minker and Seipel, 2002], allowing disjunction in the head of rules, disjunctive default logic [Gelfond et al., 1991], set-based abstract argumentation [Nielsen and Parsons, 2006], and conditional abstract dialectical frameworks [Heyninck et al., 2022]; the version of AFT defined by Liu and You [2022], which extends standard AFT by allowing for non-symmetric operators which may send exact pairs to inconsistent elements to express inconsistencies, for instance in the case of hybrid Minimal Knowledge and Negation as Failure (MKNF) knowledge bases [Motik and Rosati, 2010]; or *generalized* AFT and *flower* AFT [Vanbesien, 2025], which generalize the class of approximation spaces taken into consideration, allowing for a more versatile account of AFT, with applications in LP and AEL. Showing that these versions of AFT can be extended to higher-order definitions using our method amounts to defining a proper approximation system, with the main challenge being to prove that the approximation spaces considered form a Cartesian closed category. Once we have achieved this, all the already-existing formal results regarding AFT become readily available for their higher-order counterparts. This includes, but is not limited to, stratification results [Vennekens et al., 2006; Bogaerts and Cruz-Filipe, 2021], grounded

fixpoints [Bogaerts et al., 2015a,b], well-founded inductions [Denecker and Vennekens, 2007], safe inductions [Bogaerts et al., 2018], and strong equivalence [Truszczynski, 2006].

Finally, another theoretical line of research is the comparison between the different approaches adopted to obtain a higher-order semantics using AFT in the literature. In particular, in Section 6 of Chapter 4, we have already presented the differences between our technique and the one utilized by Charalambidis et al. [2018] when defining the fixpoint semantics for higher-order LP. The interesting part is how to deal with expressions of the form $Q(p)$. In our framework, the domain of approximate higher-order objects (like the three-valued interpretation for Q) includes approximated objects, allowing, in this way, to directly apply an approximate interpretation for Q to an approximate interpretation for p. On the other hand, Bogaerts et al. [2024] use a technique that resembles some sort of *ultimate construction* [Denecker et al., 2004]: the interpretation for $Q(p)$ is obtained by first applying the approximate interpretation for Q to *all* exact interpretations for p that are still possible, and then returning the least precise approximation of all the results. The difference in semantical treatment is reflected in the choice of the approximation spaces, with ours containing more elements and being more refined than those of Bogaerts et al. [2024] for the same type. The arising questions are multiple: is there a correspondence between the fixpoints of the two approaches? Do more refined approximation spaces provide more precise fixpoints? Is there a class of programs for which the fixpoints differ? For more details and a few claims and conjectures in this regard, we refer back to Section 6 of Chapter 4.

Developments in Applications of Approximation Fixpoint Theory Since AFT and HT logic both employ pairs of interpretations, albeit through different techniques, it is often the case that when HT logic is used to define a stable model semantics, AFT can constitute an easier route to the same result, while also giving access to other types of models for free. We have provided a clear example of this by defining the fixpoint semantics for DatalogMTL$^\neg$ in Chapter 3 using AFT, and comparing the resulting stable models with the ones obtained through the logic of HT in the literature (Section 5 of Chapter 3). Inspired by this link between HT logic and AFT, one could look into other applications of HT. For instance, we believe it would be possible to utilize the machinery of AFT to retrieve the *metric stable models* defined by Becker et al. [2024], used to obtain the non-monotonic MEL starting from the Metric Logic of HT, a temporal extension of the logic of HT. If this is indeed the case, AFT could provide a solid and reliable ground for a comparison between MEL developed by Becker et al. [2024] and the extension of Datalog MTL with negation-as-failure of Walega et al. [2024].

Another line of future work concerning Datalog may stem from the combination of the results from Chapters 3 and 4. In particular, the framework of higher-order AFT may allow for the definition of a semantics of a higher-order extension of DatalogMTL$^\neg$. It

is paramount to note that the generality of the categorical setting utilized in Chapter 4 enormously boosts the versatility of our framework, allowing for several applications. In fact, the higher-order accounts of standard AFT and consistent AFT obtained in Sections 4 and 5 of Chapter 4, respectively, may provide higher-order extensions in the application fields where AFT previously succeeded, such as abstract argumentation [Strass, 2013; Bogaerts, 2019], active integrity constraints [Bogaerts and Cruz-Filipe, 2018], stream reasoning [Antic, 2020], and integrity constraints [Bogaerts and Jakubowski, 2021], among others.

Further Advancements in Multi-Agent Autoepistemic Logic There are three properties that are often taken for granted in epistemic logic: *truthfulness*, i.e., what an agent knows is also true, *positive* and *negative introspection*, i.e., each agent knows what they do or do not know, respectively. In Section 5 of Chapter 5, we have shown how to enforce the fact that agents have positive introspection in our framework by restricting the set of possible worlds considered. Adding truthfulness seems to be feasible as well, using an analogous technique. In this case, however, principle (O) from Theorem 5.5 would not hold anymore (assuming principles (Prop), (Cut), and (M) do hold). Nevertheless, we propose a weakened version of such a property, which we believe holds:

(O') If φ is a formula not involving a modality with subscript $\mathcal{A}$, then $O_{\mathcal{A}}\varphi \nvDash_{TPI} \bot$,

where $\vDash_{TPI}$ is the entailment relation for the Kripke structure in which all agents are truthful and positively introspective. For negative introspection, naively adding negative introspection causes a more severe problem to arise: $M_{\mathcal{A}}\varphi$ is not satisfiable for any φ. This is not due to our semantic approach to only knowing, but is a direct consequence of basic properties that only knowing has been assumed to satisfy in other papers. To solve this issue, we propose a variant of principle (M) from Theorem 5.5 and a modified semantics for the knowing-at-most operator. Intuitively, $M_{\mathcal{A}}\varphi$ should be true in a world in which all worlds that satisfy all formulas entailed by the autoepistemic theory $\{\varphi\}$ are accessible. Principle (M) from Theorem 5.5 could then be rephrased with a negated autoepistemic entailment in place of the negated entailment:

(M') If $K_{\mathcal{A}}\varphi \nvdash K_{\mathcal{A}}\psi$, then $M_{\mathcal{A}}\varphi \vDash \neg K_{\mathcal{A}}\psi$.

All these modifications, regarding both truthfulness and negative introspection, are still to be precisely stated and their efficacy to be proven.

Another line of future work concerns a quite natural, yet complicated, type of knowledge resulting from the combination of common knowledge and only knowing. This is very often present in epistemic puzzles and in any situation in which a group of agents has

common knowledge of 1) a set of facts (for instance, the rules of a game), and 2) the fact that each agent "does not know more than" 1). In other words, we could state that the knowledge of each agent is *closed*, so to speak, and that *this* is common knowledge. The difficulty here arises from the fact that the awareness of the closure of the knowledge, given by the common knowledge part, influences the closure itself. We believe that the biworlds provide a set of possible worlds which is fine enough to capture this type of knowledge. However, precisely expressing such an epistemic state with the operators of $\mathcal{COL}$ appears to be rather complicated. We have expanded on this matter in Section 6 of Chapter 5.

Bibliography

Abiteboul, S., R. Hull, and V. Vianu
1995. *Foundations of Databases*. Addison-Wesley. Cited on pages 6 and 33.

Aczel, P.
1977. An introduction to inductive definitions. In *Handbook of Mathematical Logic*, J. Barwise, ed., volume 90 of *Studies in Logic and the Foundations of Mathematics*, Pp. 739–782. Elsevier. Cited on page 3.

Afanasiev, L., T. Grust, M. Marx, J. Rittinger, and J. Teubner
2008. An inflationary fixed point operator in XQuery. In *Proceedings of the 24th International Conference on Data Engineering, ICDE 2008, April 7-12, 2008, Cancún, Mexico*, G. Alonso, J. A. Blakeley, and A. L. P. Chen, eds., Pp. 1504–1506. IEEE Computer Society. Cited on page 3.

Antic, C.
2020. Fixed point semantics for stream reasoning. *Artif. Intell.*, 288:103370. Cited on pages 6, 104, and 157.

Aref, M., B. ten Cate, T. J. Green, B. Kimelfeld, D. Olteanu, E. Pasalic, T. L. Veldhuizen, and G. Washburn
2015. Design and implementation of the LogicBlox system. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015*, T. K. Sellis, S. B. Davidson, and Z. G. Ives, eds., Pp. 1371–1382. ACM. Cited on page 6.

Aucher, G. and V. Belle
2015. Multi-agent only knowing on planet Kripke. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, Q. Yang and M. J. Wooldridge, eds., Pp. 2713–2719. AAAI Press. Cited on pages 11, 106, 148, and 149.

Aumann, R. J.
1976. Agreeing to disagree. *The Annals of Statistics*, 4(6):1236–1239. Cited on pages 10 and 148.

Baral, C.
2003. *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge University Press. Cited on page 2.

Becker, A., P. Cabalar, M. Diéguez, T. Schaub, and A. Schuhmann
2024. Metric temporal equilibrium logic over timed traces. *Theory Pract. Log. Program.*, 24(3):425–452. Cited on pages 55 and 156.

Belle, V. and G. Lakemeyer
2010. Multi-agent only-knowing revisited. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010*, F. Lin, U. Sattler, and M. Truszczynski, eds. AAAI Press. Cited on pages 11 and 148.

Belle, V. and G. Lakemeyer
2015a. Only knowing meets common knowledge. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, Q. Yang and M. J. Wooldridge, eds., Pp. 2755–2761. AAAI Press. Cited on pages 11, 12, 106, 107, and 149.

Belle, V. and G. Lakemeyer
2015b. Semantical considerations on multiagent only knowing. *Artif. Intell.*, 223:1–26. Cited on pages 11, 106, 141, and 148.

Belnap, N. D.
1977. *A Useful Four-Valued Logic*, Pp. 5–37. Dordrecht: Springer Netherlands. Cited on pages 2 and 29.

Bogaerts, B.
2019. Weighted abstract dialectical frameworks through the lens of approximation fixpoint theory. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, Pp. 2686–2693. AAAI Press. Cited on pages 6, 35, 104, and 157.

Bogaerts, B., A. Charalambidis, G. Chatziagapis, B. Kostopoulos, S. Pollaci, and P. Rondogiannis
2024. The stable model semantics for higher-order logic programming. *Theory Pract. Log. Program.*, 24(4):737–754. Cited on pages xv, 59, 60, 61, 100, 101, 102, 103, and 156.

Bogaerts, B. and L. Cruz-Filipe
2018. Fixpoint semantics for active integrity constraints. *Artif. Intell.*, 255:43–70. Cited on pages 6, 104, and 157.

Bogaerts, B. and L. Cruz-Filipe
2021. Stratification in approximation fixpoint theory and its application to active integrity constraints. *ACM Trans. Comput. Log.*, 22(1):6:1–6:19. Cited on pages 55, 104, and 155.

Bogaerts, B. and M. Jakubowski
2021. Fixpoint semantics for recursive SHACL. In *Proceedings 37th International Conference on Logic Programming (Technical Communications), ICLP Technical Communications 2021, Porto (virtual event), 20-27th September 2021*, A. Formisano, Y. A. Liu, B. Bogaerts, A. Brik, V. Dahl, C. Dodaro, P. Fodor, G. L. Pozzato, J. Vennekens, and N. Zhou, eds., volume 345 of *EPTCS*, Pp. 41–47. Cited on pages 6, 104, and 157.

Bogaerts, B., J. Vennekens, and M. Denecker
2015a. Grounded fixpoints and their applications in knowledge representation. *Artif. Intell.*, 224:51–71. Cited on pages 104 and 156.

Bogaerts, B., J. Vennekens, and M. Denecker
2015b. Partial grounded fixpoints. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, Q. Yang and M. J. Wooldridge, eds., Pp. 2784–2790. AAAI Press. Cited on page 156.

Bogaerts, B., J. Vennekens, and M. Denecker
2018. Safe inductions and their applications in knowledge representation. *Artif. Intell.*, 259:167–185. Cited on page 156.

Bonzio, S., J. Gil-Férez, F. Paoli, and L. Peruzzi
2017. On paraconsistent weak Kleene logic: Axiomatisation and algebraic analysis. *Studia Logica: An International Journal for Symbolic Logic*, 105(2):253–297. Cited on page 2.

Boole, G.
2009. *The Mathematical Analysis of Logic: Being an Essay Towards a Calculus of Deductive Reasoning*, Cambridge Library Collection - Mathematics. Cambridge University Press. Cited on page 2.

Brandt, S., E. G. Kalayci, V. Ryzhikov, G. Xiao, and M. Zakharyaschev
2018. Querying log data with metric temporal logic. *J. Artif. Intell. Res.*, 62:829–877. Cited on pages 6, 7, and 33.

Calvanese, D., G. D. Giacomo, and M. Lenzerini
1999. Reasoning in expressive description logics with fixpoints based on automata on infinite trees. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI 99, Stockholm, Sweden, July 31 - August 6, 1999. 2 Volumes, 1450 pages*, T. Dean, ed., Pp. 84–89. Morgan Kaufmann. Cited on page 3.

Charalambidis, A. and P. Rondogiannis
2023. Categorical approximation fixpoint theory. In *Logics in Artificial Intelligence - 18th European Conference, JELIA 2023, Dresden, Germany, September 20-22, 2023, Proceedings*, S. A. Gaggl, M. V. Martinez, and M. Ortiz, eds., volume 14281 of *Lecture Notes in Computer Science*, Pp. 515–530. Springer. Cited on pages 6, 59, and 154.

Charalambidis, A., P. Rondogiannis, and I. Symeonidou
2018. Approximation fixpoint theory and the well-founded semantics of higher-order logic programs. *Theory Pract. Log. Program.*, 18(3-4):421–437. Cited on pages xiv, 8, 9, 26, 57, 58, 59, 60, 61, 62, 80, 81, 84, 85, 86, 87, 88, 89, 91, 100, 101, 102, 103, 154, 155, and 156.

Chen, W., M. Kifer, and D. S. Warren
1993. HILOG: A foundation for higher-order logic programming. *J. Log. Program.*, 15(3):187–230. Cited on pages 8 and 104.

Cintula, P. and P. Hájek
2006. On theories and models in fuzzy predicate logics. *J. Symb. Log.*, 71(3):863–880. Cited on page 2.

Clark, K. L.
1977. Negation as failure. In *Logic and Data Bases, Symposium on Logic and Data Bases, Centre d'études et de recherches de Toulouse, France, 1977*, H. Gallaire and J. Minker, eds., Advances in Data Base Theory, Pp. 293–322, New York. Plemum Press. Cited on page 29.

Colombo, A., L. Bellomarini, S. Ceri, and E. Laurenza
2023. Smart derivative contracts in DatalogMTL. In *Proceedings 26th International Conference on Extending Database Technology, EDBT 2023, Ioannina, Greece, March 28-31, 2023*, J. Stoyanovich, J. Teubner, N. Mamoulis, E. Pitoura, J. Mühlig, K. Hose, S. S. Bhowmick, and M. Lissandrini, eds., Pp. 773–781. OpenProceedings.org. Cited on pages 7 and 33.

Cramer, M., S. Pollaci, and B. Bogaerts
2023a. Mathematical foundations for joining only knowing and common knowledge. In *Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning, KR 2023, Rhodes, Greece, September 2-8, 2023*, P. Marquis, T. C. Son, and G. Kern-Isberner, eds., Pp. 167–177. Cited on pages 5, 13, and 105.

Cramer, M., S. Pollaci, and B. Bogaerts
2023b. Mathematical foundations for joining only knowing and common knowledge (extended version). *CoRR*, abs/2306.03267. Cited on pages 5, 13, and 105.

Cucala, D. J. T., P. A. Walega, B. C. Grau, and E. V. Kostylev
2021. Stratified negation in Datalog with metric temporal operators. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, Pp. 6488–6495. AAAI Press. Cited on pages 34 and 55.

Dasseville, I., M. van der Hallen, B. Bogaerts, G. Janssens, and M. Denecker
2016. A compositional typed higher-order logic with definitions. In *Technical*

Communications of the 32nd International Conference on Logic Programming, ICLP 2016 TCs, New York City, USA, October 16-21, 2016, M. Carro, A. King, N. Saeedloei, and M. D. Vos, eds., volume 52 of *OASIcs*, Pp. 14:1–14:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik. Cited on pages 8, 57, and 59.

Dasseville, I., M. van der Hallen, G. Janssens, and M. Denecker
2015. Semantics of templates in a compositional framework for building logics. *Theory Pract. Log. Program.*, 15(4-5):681–695. Cited on pages 7 and 57.

Davey, B. A. and H. A. Priestley
2002. *Introduction to Lattices and Order, Second Edition*. Cambridge University Press. Cited on page 16.

de Finetti, B.
1995. The logic of probability. *Philosophical Studies*, 77(1):181–190. Cited on page 2.

Denecker, M., M. Bruynooghe, and J. Vennekens
2012. Approximation fixpoint theory and the semantics of logic and answers set programs. In *Correct Reasoning - Essays on Logic-Based AI in Honour of Vladimir Lifschitz*, E. Erdem, J. Lee, Y. Lierler, and D. Pearce, eds., volume 7265 of *Lecture Notes in Computer Science*, Pp. 178–194. Springer. Cited on page 29.

Denecker, M., V. Marek, and M. Truszczyński
2000. Approximations, stable operators, well-founded fixpoints and applications in nonmonotonic reasoning. In *Logic-Based Artificial Intelligence*, volume 597, Pp. 127–144. Springer US. Cited on pages 4, 5, 26, 28, 29, 58, 59, 60, 79, 86, 154, and 155.

Denecker, M., V. W. Marek, and M. Truszczynski
2004. Ultimate approximation and its application in nonmonotonic knowledge representation systems. *Inf. Comput.*, 192(1):84–121. Cited on pages 6, 26, 30, 31, 34, 52, 58, 59, 60, 70, 100, 154, 155, and 156.

Denecker, M., V. W. Marek, and M. Truszczynski
2011. Reiter's default logic is a logic of autoepistemic reasoning and a good one, too. *CoRR*, abs/1108.3278. Cited on page 5.

Denecker, M. and E. Ternovska
2008. A logic of nonmonotone inductive definitions. *ACM Trans. Comput. Log.*, 9(2):14:1–14:52. Cited on page 3.

Denecker, M. and J. Vennekens
2007. Well-founded semantics and the algebraic theory of non-monotone inductive definitions. In *Logic Programming and Nonmonotonic Reasoning, 9th International Conference, LPNMR 2007, Tempe, AZ, USA, May 15-17, 2007, Proceedings*, C. Baral, G. Brewka, and J. S. Schlipf, eds., volume 4483 of *Lecture Notes in Computer Science*, Pp. 84–96. Springer. Cited on pages 3, 4, 22, 29, 30, 79, 104, and 156.

Dunn, J. M.
1976. Intuitive semantics for first-degree entailments and 'coupled trees'. *Philosophical Studies: An International Journal for Philosophy in the Analytic Tradition*, 29(3):149–168. Cited on page 2.

Eilenberg, S. and S. MacLane
1945. General theory of natural equivalences. *Transactions of the American Mathematical Society*, 58:231–294. Cited on page 8.

Fagin, R., J. Y. Halpern, Y. Moses, and M. Y. Vardi
1995. *Reasoning About Knowledge*. MIT Press. Cited on pages 10, 106, and 148.

Fagin, R., J. Y. Halpern, and M. Y. Vardi
1991. A model-theoretic analysis of knowledge. *J. ACM*, 38(2):382–428. Cited on pages 12 and 107.

Feferman, S.
1970. Formal theories for transfinite iterations of generalized inductive definitions and some subsystems of analysis. In *Intuitionism and Proof Theory: Proceedings of the Summer Conference at Buffalo N.Y. 1968*, A. Kino, J. Myhill, and R. Vesley, eds., volume 60 of *Studies in Logic and the Foundations of Mathematics*, Pp. 303–326. Elsevier. Cited on page 3.

Fitting, M.
1985. A Kripke-Kleene semantics for logic programs. *J. Log. Program.*, 2(4):295–312. Cited on page 29.

Fitting, M.
2002. Fixpoint semantics for logic programming a survey. *Theor. Comput. Sci.*, 278(1-2):25–51. Cited on pages 29, 31, 35, 39, and 42.

Gelfond, M. and V. Lifschitz
1988. The stable model semantics for logic programming. In *Logic Programming, Proceedings of the Fifth International Conference and Symposium, Seattle, Washington, USA, August 15-19, 1988 (2 Volumes)*, R. A. Kowalski and K. A. Bowen, eds., Pp. 1070–1080. MIT Press. Cited on page 29.

Gelfond, M., H. Przymusinska, V. Lifschitz, and M. Truszczynski
1991. Disjective defaults. In *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91). Cambridge, MA, USA, April 22-25, 1991*, J. F. Allen, R. Fikes, and E. Sandewall, eds., Pp. 230–237. Morgan Kaufmann. Cited on page 155.

Hájek, P.
1998. *Metamathematics of Fuzzy Logic*, volume 4 of *Trends in Logic*. Kluwer. Cited on page 2.

Hallnäs, L.
1991. Partial inductive definitions. *Theoretical Computer Science*, 87(1):115–142. Cited on page 3.

Halpern, J. Y.
1993. Reasoning about only knowing with many agents. In *Proceedings of the 11th National Conference on Artificial Intelligence. Washington, DC, USA, July 11-15, 1993*, R. Fikes and W. G. Lehnert, eds., Pp. 655–661. AAAI Press / The MIT Press. Cited on pages 11 and 148.

Halpern, J. Y. and G. Lakemeyer
1995. Levesque's axiomatization of only knowing is incomplete. *Artif. Intell.*, 74(2):381–387. Cited on pages 11 and 148.

Halpern, J. Y. and G. Lakemeyer
2001. Multi-agent only knowing. *J. Log. Comput.*, 11(1):41–70. Cited on pages 11, 106, and 148.

Halpern, J. Y. and Y. Moses
1984. Towards a theory of knowledge and ignorance: Preliminary report. In *Logics and Models of Concurrent Systems - Conference proceedings, Colle-sur-Loup (near Nice), France, 8-19 October 1984*, K. R. Apt, ed., NATO ASI Series, Pp. 459–476. Springer. Cited on pages 11 and 148.

Heyninck, J., O. Arieli, and B. Bogaerts
2024. Non-deterministic approximation fixpoint theory and its application in disjunctive logic programming. *Artif. Intell.*, 331:104110. Cited on pages 6, 26, 154, and 155.

Heyninck, J., M. Thimm, G. Kern-Isberner, T. Rienstra, and K. Skiba
2022. Conditional abstract dialectical frameworks. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, Pp. 5692–5699. AAAI Press. Cited on page 155.

Heyting, A.
1930. Die formalen Regeln der intuitionistischen Logik. In *Sitzungsberichte der Preussischen Akademie der Wissenschaften, physikalisch-mathematische klass.* Cited on page 34.

Hintikka, K. J. J.
1962. *Knowledge and Belief: An Introduction to the Logic of the Two Notions.* Ithaca, NY, USA: Cornell University Press. Cited on pages 9 and 106.

Huang, S. S., T. J. Green, and B. T. Loo
2011. Datalog and emerging applications: an interactive tutorial. In *Proceedings of the*

ACM SIGMOD International Conference on Management of Data, SIGMOD 2011, Athens, Greece, June 12-16, 2011, T. K. Sellis, R. J. Miller, A. Kementsietsidis, and Y. Velegrakis, eds., Pp. 1213–1216. ACM. Cited on page 33.

Kalayci, E. G., G. Xiao, V. Ryzhikov, T. E. Kalayci, and D. Calvanese
2018. Ontop-temporal: A tool for ontology-based query answering over temporal data. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, A. Cuzzocrea, J. Allan, N. W. Paton, D. Srivastava, R. Agrawal, A. Z. Broder, M. J. Zaki, K. S. Candan, A. Labrinidis, A. Schuster, and H. Wang, eds., Pp. 1927–1930. ACM. Cited on pages 6 and 33.

Ketsman, B. and P. Koutris
2022. Modern Datalog engines. *Found. Trends Databases*, 12(1):1–68. Cited on page 33.

Kikot, S., V. Ryzhikov, P. A. Walega, and M. Zakharyaschev
2018. On the data complexity of ontology-mediated queries with MTL operators over timed words. In *Proceedings of the 31st International Workshop on Description Logics co-located with 16th International Conference on Principles of Knowledge Representation and Reasoning (KR 2018), Tempe, Arizona, US, October 27th - to - 29th, 2018*, M. Ortiz and T. Schneider, eds., volume 2211 of *CEUR Workshop Proceedings*. CEUR-WS.org. Cited on pages 6 and 33.

Kleene, S. C.
1952. *Introduction to Metamathematics*. Groningen: P. Noordhoff N.V. Cited on pages 2 and 31.

Konolige, K.
1982. Circumscriptive ignorance. In *Proceedings of the National Conference on Artificial Intelligence, Pittsburgh, PA, USA, August 18-20, 1982*, D. L. Waltz, ed., Pp. 202–204. AAAI Press. Cited on pages 11 and 148.

Konolige, K.
1988. On the relation between default and autoepistemic logic. *Artif. Intell.*, 35(3):343–382. Cited on pages 11 and 148.

Koopmann, P.
2019. Ontology-based query answering for probabilistic temporal data. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, Pp. 2903–2910. AAAI Press. Cited on pages 6 and 33.

Koymans, R.
1990. Specifying real-time properties with metric temporal logic. *Real Time Syst.*, 2(4):255–299. Cited on pages 7 and 33.

Kozen, D.
1983. Results on the propositional mu-calculus. *Theor. Comput. Sci.*, 27:333–354. Cited on page 3.

Kreisel, G.
1963. Generalized inductive definitions. *Tech. rep., Section III in the Stanford University report on the Foundations of Analysis.* Cited on page 3.

Lakemeyer, G.
1993. All they know: A study in multi-agent autoepistemic reasoning. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence. Chambéry, France, August 28 - September 3, 1993*, R. Bajcsy, ed., Pp. 376–381. Morgan Kaufmann. Cited on pages 11, 141, and 148.

Lakemeyer, G. and H. J. Levesque
2005. Only-knowing: Taking it beyond autoepistemic reasoning. In *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*, M. M. Veloso and S. Kambhampati, eds., Pp. 633–638. AAAI Press / The MIT Press. Cited on pages 11 and 148.

Levesque, H. J.
1990. All I know: A study in autoepistemic logic. *Artif. Intell.*, 42(2-3):263–309. Cited on pages 11 and 148.

Levesque, H. J. and G. Lakemeyer
2000. *The logic of knowledge bases.* MIT Press. Cited on pages 11 and 148.

Lewis, D.
1969. *Convention: A Philosophical Study.* Wiley-Blackwell. Cited on pages 10 and 148.

Liu, F. and J. You
2022. Alternating fixpoint operator for hybrid MKNF knowledge bases as an approximator of AFT. *Theory Pract. Log. Program.*, 22(2):305–334. Cited on page 155.

Loo, B. T., T. Condie, M. N. Garofalakis, D. E. Gay, J. M. Hellerstein, P. Maniatis, R. Ramakrishnan, T. Roscoe, and I. Stoica
2006. Declarative networking: language, execution and optimization. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Chicago, Illinois, USA, June 27-29, 2006*, S. Chaudhuri, V. Hristidis, and N. Polyzotis, eds., Pp. 97–108. ACM. Cited on page 6.

Marcos, J., A. Přenosil, and P. Egré
2026. Many-valued logic. In *The Stanford Encyclopedia of Philosophy*, E. N. Zalta and U. Nodelman, eds. Metaphysics Research Lab, Stanford University. Cited on page 2.

Marczak, W. R., S. S. Huang, M. Bravenboer, M. Sherr, B. T. Loo, and M. Aref
2010. SecureBlox: customizable secure distributed data processing. In *Proceedings*

of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2010, Indianapolis, Indiana, USA, June 6-10, 2010, A. K. Elmagarmid and D. Agrawal, eds., Pp. 723–734. ACM. Cited on page 6.

Markowsky, G.
1976. Chain-complete posets and directed sets with applications. *Algebra Universalis*, 6:53–68. Cited on page 21.

Marquis, J.
2026. Category Theory. In *The Stanford Encyclopedia of Philosophy*, E. N. Zalta and U. Nodelman, eds. Metaphysics Research Lab, Stanford University. Cited on page 8.

Martin-Löf, P.
1971. Hauptsatz for the intuitionistic theory of iterated inductive definitions. In *Proceedings of the Second Scandinavian Logic Symposium*, J. Fenstad, ed., volume 63 of *Studies in Logic and the Foundations of Mathematics*, Pp. 179–216. Elsevier. Cited on page 3.

Meyer, J. C. and W. van der Hoek
1995. *Epistemic Logic for AI and Computer Science*, volume 41 of *Cambridge tracts in theoretical computer science*. Cambridge University Press. Cited on pages 10, 106, and 148.

Minker, J. and D. Seipel
2002. Disjunctive logic programming: A survey and assessment. In *Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert A. Kowalski, Part I*, A. C. Kakas and F. Sadri, eds., volume 2407 of *Lecture Notes in Computer Science*, Pp. 472–511. Springer. Cited on page 155.

Moore, R. C.
1985. Semantical considerations on nonmonotonic logic. *Artif. Intell.*, 25(1):75–94. Cited on pages 4, 5, 9, 10, 141, and 148.

Morgenstern, L.
1990. A formal theory of multiple agent nonmonotonic reasoning. In *Proceedings of the 8th National Conference on Artificial Intelligence. Boston, Massachusetts, USA, July 29 - August 3, 1990, 2 Volumes*, H. E. Shrobe, T. G. Dietterich, and W. R. Swartout, eds., Pp. 538–544. AAAI Press / The MIT Press. Cited on page 10.

Morgenstern, L. and R. Guerreiro
1996. Epistemic logics for multiple agent nonmonotonic reasoning I. Cited on page 10.

Mori, M., P. Papotti, L. Bellomarini, and O. Giudice
2022. Neural machine translation for fact-checking temporal claims. In *Proceedings of the Fifth Fact Extraction and VERification Workshop (FEVER)*, R. Aly, C. Christodoulopoulos, O. Cocarascu, Z. Guo, A. Mittal, M. Schlichtkrull, J. Thorne, and A. Vlachos, eds.,

Pp. 78–82, Dublin, Ireland. Association for Computational Linguistics. Cited on pages 7 and 33.

Moschovakis, Y. N.
1974. *Elementary Induction on Abstract Structures*. Mineola, N.Y.: Dover Publications. Cited on page 3.

Moss, L. S.
2007. Finite models constructed from canonical formulas. *Journal of Philosophical Logic*, 36(6):605–640. Cited on page 149.

Motik, B. and R. Rosati
2010. Reconciling description logics and rules. *J. ACM*, 57(5):30:1–30:62. Cited on page 155.

Nadathur, G. and D. Miller
1988. An overview of Lambda-PROLOG. In *Logic Programming, Proceedings of the Fifth International Conference and Symposium, Seattle, Washington, USA, August 15-19, 1988 (2 Volumes)*, R. A. Kowalski and K. A. Bowen, eds., Pp. 810–827. MIT Press. Cited on page 8.

Nielsen, S. H. and S. Parsons
2006. A generalization of Dung's abstract framework for argumentation: Arguing with sets of attacking arguments. In *Argumentation in Multi-Agent Systems, Third International Workshop, ArgMAS 2006, Hakodate, Japan, May 8, 2006, Revised Selected and Invited Papers*, N. Maudet, S. Parsons, and I. Rahwan, eds., volume 4766 of *Lecture Notes in Computer Science*, Pp. 54–73. Springer. Cited on page 155.

Nissl, M. and E. Sallinger
2022. Modelling smart contracts with DatalogMTL. In *Proceedings of the Workshops of the EDBT/ICDT 2022 Joint Conference, Edinburgh, UK, March 29, 2022*, M. Ramanath and T. Palpanas, eds., volume 3135 of *CEUR Workshop Proceedings*. CEUR-WS.org. Cited on pages 7 and 33.

Pearce, D.
1996. A new logical characterisation of stable models and answer sets. In *Non-Monotonic Extensions of Logic Programming, NMELP '96, Bad Honnef, Germany, September 5-6, 1996, Selected Papers*, J. Dix, L. M. Pereira, and T. C. Przymusinski, eds., volume 1216 of *Lecture Notes in Computer Science*, Pp. 57–70. Springer. Cited on page 34.

Pelov, N., M. Denecker, and M. Bruynooghe
2007. Well-founded and stable semantics of logic programs with aggregates. *Theory Pract. Log. Program.*, 7(3):301–353. Cited on page 59.

Permpoontanalarp, Y. and J. Y. Jiang
1995. On multi-agent autoepistemic reasoning. In *Proceedings of the Second World*

Conference on the Fundamentals of Artificial Intelligence, WOCFAI 1995, 3-7 July 1995, Paris, France, M. D. Glas and Z. Pawlak, eds., Pp. 307–318. Angkor, 6, rue Maître-Albert, 75005 Paris, France. Cited on page 141.

Pollaci, S.
2025. Fixpoint semantics for DatalogMTL with negation. In *Proceedings 41st International Conference on Logic Programming, ICLP 2025, Rende, Italy, 12-19th September 2025*, M. Gebser, D. Inclezan, F. Ricca, M. Carro, and M. Truszczynski, eds., volume 439 of *EPTCS*, Pp. 263–277. Cited on pages 5, 6, 7, 33, and 104.

Pollaci, S., B. Kostopoulos, M. Denecker, and B. Bogaerts
2024. A category-theoretic perspective on higher-order approximation fixpoint theory. In *Logic Programming and Nonmonotonic Reasoning - 17th International Conference, LPNMR 2024, Dallas, TX, USA, October 11-14, 2024, Proceedings*, C. Dodaro, G. Gupta, and M. V. Martinez, eds., volume 15245 of *Lecture Notes in Computer Science*, Pp. 317–330. Springer. Cited on pages 5, 9, and 57.

Pollaci, S., B. Kostopoulos, M. Denecker, and B. Bogaerts
2025. A category-theoretic perspective on higher-order approximation fixpoint theory. *Theory and Practice of Logic Programming*, P. 1–34. Cited on pages 5, 9, and 57.

Post, E. L.
1943. Formal reductions of the general combinatorial decision problem. *American Journal of Mathematics*, 65:197. Cited on page 3.

Pratt-Hartmann, I.
2000. Total knowledge. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, P. 423–428. AAAI Press. Cited on pages 11 and 148.

Priest, G.
1979. The logic of paradox. *Journal of Philosophical Logic*, 8(1):219–241. Cited on page 2.

Prior, A. N.
1953. Three-valued logic and future contingents. *The Philosophical Quarterly (1950-)*, 3(13):317–326. Cited on page 2.

Putnam, H.
1957. Three-valued logic. *Philosophical Studies: An International Journal for Philosophy in the Analytic Tradition*, 8(5):73–80. Cited on page 2.

Reiter, R.
1980. A logic for default reasoning. *Artif. Intell.*, 13(1-2):81–132. Cited on pages 5, 11, and 148.

Riehl, E.
2017. *Category Theory in Context*, Aurora: Dover modern math originals. Dover Publications. Cited on pages 8, 61, and 68.

Roberts, E. S.
1986. *Thinking Recursively.* Wiley. Cited on page 3.

Rondogiannis, P. and I. Symeonidou
2018. Extensional semantics for higher-order logic programs with negation. *Log. Methods Comput. Sci.*, 14(2). Cited on page 104.

Rosati, R.
2000. On the decidability and complexity of reasoning about only knowing. *Artif. Intell.*, 116(1-2):193–215. Cited on pages 11 and 148.

Rubio-Sánchez, M.
2018. *Introduction to Recursive Programming*, first edition. Taylor & Francis. Cited on page 3.

Seo, J., S. Guo, and M. S. Lam
2015. Socialite: An efficient graph query language based on Datalog. *IEEE Trans. Knowl. Data Eng.*, 27(7):1824–1837. Cited on page 6.

Shkapsky, A., M. Yang, M. Interlandi, H. Chiu, T. Condie, and C. Zaniolo
2016. Big data analytics with Datalog queries on Spark. In *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, F. Özcan, G. Koutrika, and S. Madden, eds., Pp. 1135–1149. ACM. Cited on page 6.

Smaragdakis, Y. and G. Balatsouras
2015. Pointer analysis. *Found. Trends Program. Lang.*, 2(1):1–69. Cited on page 6.

Soare, R. I.
1987. *Recursively Enumerable Sets and Degrees - a study of computable functions and computability generated sets*, Perspectives in mathematical logic. Springer. Cited on page 3.

Spector, C.
1961. Inductively defined sets of natural numbers. *Infinitistic methods (Proceedings of the Symposium on Foundations ofMathematics, Warsaw, 1959)*, Pp. 97 – 102. Cited on page 3.

Strass, H.
2013. Approximating operators and semantics for abstract dialectical frameworks. *Artif. Intell.*, 205:39–70. Cited on pages 6, 104, and 157.

Tarski, A.
1955. A lattice-theoretical fixpoint theorem and its applications. *Pacific journal of mathematics 5*, Pp. 285–309. Cited on pages 5, 21, and 52.

Taylor, R.
1957. The problem of future contingencies. *The Philosophical Review*, 66(1):1–28. Cited on page 2.

Truszczynski, M.
2006. Strong and uniform equivalence of nonmonotonic theories - an algebraic approach. *Ann. Math. Artif. Intell.*, 48(3-4):245–265. Cited on pages 104 and 156.

van der Hoek, W. and E. Thijsse
2002. A general approach to multi-agent minimal knowledge: With tools and samples. *Stud Logica*, 72(1):61–84. Cited on pages 11 and 148.

van Emden, M. H. and R. A. Kowalski
1976. The semantics of predicate logic as a programming language. *J. ACM*, 23(4):733–742. Cited on page 21.

Van Gelder, A., K. A. Ross, and J. S. Schlipf
1991. The well-founded semantics for general logic programs. *J. ACM*, 38(3):620–650. Cited on page 29.

Van Hertum, P.
2016. *New Language Constructs and Inferences for the Knowledge Base Paradigm: A Business and Multi-agent Perspective ; Taaluitbreidingen en nieuwe inferenties voor het kennisbank paradigma vanuit een Business en Multi-Agent perspectief*. PhD thesis, Katholieke Universiteit Leuven, Belgium. Cited on pages 11, 106, and 149.

Van Hertum, P., M. Cramer, B. Bogaerts, and M. Denecker
2016. Distributed autoepistemic logic and its application to access control. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, S. Kambhampati, ed., Pp. 1286–1292. IJCAI/AAAI Press. Cited on pages 10 and 141.

Vanbesien, L.
2025. *Algebraic Formalizations of Constructive Knowledge*. Phd thesis, Katholieke Universiteit Leuven, Leuven, Belgium. Available at https://kuleuven.limo.libis.be/discovery/fulldisplay?docid=lirias4263634&context=SearchWebhook&vid=32KUL_KUL:Lirias&lang=en&search_scope=lirias_profile&adaptor=SearchWebhook&tab=LIRIAS&query=any,contains,LIRIAS4263634&offset=0. Cited on pages 22, 26, 154, and 155.

Vanbesien, L., S. Pollaci, B. Bogaerts, and M. Denecker
2024. Towards a unifying view on monotone constructive definitions. In *Artificial Intelligence and Image Analysis - 18th International Symposium on Artificial Intelligence and Mathematics, ISAIM 2024, and 22nd International Workshop on Combinatorial Image Analysis, IWCIA 2024, Fort Lauderdale, FL, USA, January 8-10, 2024, Revised Selected Papers*, R. P. Barneva, V. E. Brimkov, C. Gentile, and A. Pacchiano, eds., Lecture Notes in Computer Science, Pp. 218–235. Springer. Cited on page 3.

Vennekens, J., D. Gilis, and M. Denecker
2006. Splitting an operator: Algebraic modularity results for logics with fixpoint semantics. *ACM Trans. Comput. Log.*, 7(4):765–797. Cited on pages 55, 104, and 155.

Vlaeminck, H., J. Vennekens, M. Bruynooghe, and M. Denecker
2012. Ordered epistemic logic: Semantics, complexity and applications. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10-14, 2012*, G. Brewka, T. Eiter, and S. A. McIlraith, eds. AAAI Press. Cited on pages 10 and 141.

Waaler, A.
2004. Consistency proofs for systems of multi-agent only knowing. In *Advances in Modal Logic 5, papers from the fifth conference on "Advances in Modal logic," held in Manchester, UK, 9-11 September 2004*, R. A. Schmidt, I. Pratt-Hartmann, M. Reynolds, and H. Wansing, eds., Pp. 347–366. King's College Publications. Cited on pages 11 and 148.

Waaler, A. and B. Solhaug
2005. Semantics for multi-agent only knowing: extended abstract. In *Proceedings of the 10th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-2005), Singapore, June 10-12, 2005*, R. van der Meyden, ed., Pp. 109–125. National University of Singapore. Cited on pages 11, 106, and 148.

Walega, P. A., D. J. T. Cucala, B. C. Grau, and E. V. Kostylev
2024. The stable model semantics of Datalog with metric temporal operators. *Theory Pract. Log. Program.*, 24(1):22–56. Cited on pages 7, 34, 35, 36, 48, 52, 55, 153, and 156.

Walega, P. A., M. Kaminski, and B. C. Grau
2019. Reasoning over streaming data in metric temporal Datalog. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, Pp. 3092–3099. AAAI Press. Cited on pages 6 and 33.

Walega, P. A., M. Kaminski, D. Wang, and B. C. Grau
2023. Stream reasoning with DatalogMTL. *J. Web Semant.*, 76:100776. Cited on pages 6 and 33.

Whaley, J. and M. S. Lam
2004. Cloning-based context-sensitive pointer alias analysis using binary decision diagrams. In *Proceedings of the ACM SIGPLAN 2004 Conference on Programming Language Design and Implementation 2004, Washington, DC, USA, June 9-11, 2004*, W. W. Pugh and C. Chambers, eds., Pp. 131–144. ACM. Cited on page 6.

List of Publications

Bogaerts, B., M. Calautti, A. Pieris, S. Pollaci, and R. V. den Eede
2026. Why(-not)-provenance for datalog with negation. In *Proceedings of the 23rd International Conference on Principles of Knowledge Representation and Reasoning, KR 2026, Lisbon, Portugal, July 20-23, 2026*, R. Wassermann, M. Mugnier, and F. Baader, eds.

Bogaerts, B., A. Charalambidis, G. Chatziagapis, B. Kostopoulos, S. Pollaci, and P. Rondogiannis
2024. The stable model semantics for higher-order logic programming. *Theory Pract. Log. Program.*, 24(4):737–754.

Cramer, M., S. Pollaci, and B. Bogaerts
2023a. Mathematical foundations for joining only knowing and common knowledge. In *Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning, KR 2023, Rhodes, Greece, September 2-8, 2023*, P. Marquis, T. C. Son, and G. Kern-Isberner, eds., Pp. 167–177.

Cramer, M., S. Pollaci, and B. Bogaerts
2023b. Mathematical foundations for joining only knowing and common knowledge (extended version). *CoRR*, abs/2306.03267.

Pollaci, S.
2023. Spurious valleys and clustering behavior of neural networks. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, eds., Proceedings of Machine Learning Research, Pp. 28079–28099. PMLR.

Pollaci, S.
2025. Fixpoint semantics for DatalogMTL with negation. In *Proceedings 41st International Conference on Logic Programming, ICLP 2025, Rende, Italy, 12-19th September 2025*, M. Gebser, D. Inclezan, F. Ricca, M. Carro, and M. Truszczynski, eds., volume 439 of *EPTCS*, Pp. 263–277.

Pollaci, S., B. Kostopoulos, M. Denecker, and B. Bogaerts
2024. A category-theoretic perspective on higher-order approximation fixpoint theory. In *Logic Programming and Nonmonotonic Reasoning - 17th International Conference, LPNMR 2024, Dallas, TX, USA, October 11-14, 2024, Proceedings*, C. Dodaro, G. Gupta, and M. V. Martinez, eds., volume 15245 of *Lecture Notes in Computer Science*, Pp. 317–330. Springer.

Pollaci, S., B. Kostopoulos, M. Denecker, and B. Bogaerts
2025. A category-theoretic perspective on higher-order approximation fixpoint theory. *Theory and Practice of Logic Programming*, P. 1–34.

Röpke, W., S. Pollaci, B. Vandenbogaerde, J. Li, and Y. Coppens
2022. Multi-objective scheduling for agricultural interventions. In *Bnaic / Benelearn*.

Vanbesien, L., S. Pollaci, B. Bogaerts, and M. Denecker
2024. Towards a unifying view on monotone constructive definitions. In *Artificial Intelligence and Image Analysis - 18th International Symposium on Artificial Intelligence and Mathematics, ISAIM 2024, and 22nd International Workshop on Combinatorial Image Analysis, IWCIA 2024, Fort Lauderdale, FL, USA, January 8-10, 2024, Revised Selected Papers*, R. P. Barneva, V. E. Brimkov, C. Gentile, and A. Pacchiano, eds., Lecture Notes in Computer Science, Pp. 218–235. Springer.