

Proceedings of the 8th Scandinavian Logic Symposium

Jørgensen, Klaus Frovin; Blackburn, Patrick Rowan; Jones, Neil Deaton; Palmgren, Erik

Publication date:
2012

Document Version
Early version, also known as pre-print

Citation for published version (APA):
Jørgensen, K. F., Blackburn, P. R., Jones, N. D., & Palmgren, E. (Eds.) (2012). *Proceedings of the 8th Scandinavian Logic Symposium*. Roskilde Universitet.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain.
- You may freely distribute the URL identifying the publication in the public portal.

Take down policy

If you believe that this document breaches copyright please contact rucforsk@kb.dk providing details, and we will remove access to the work immediately and investigate your claim.

8th Scandinavian Logic Symposium



20-21 August 2012, Roskilde University, Denmark

Abstracts:

Patrick Blackburn
Klaus Frovin Jørgensen
Neil Jones
Erik Palmgren



Microsoft®

"Microsoft is a trademark of the Microsoft group of companies and is used under license from Microsoft."

Content

Ove Ahlman / A zero-one law for l -colourable structures with a vectorspace pregeometry	1
Mai Ajspur and John P. Gallagher / Towards Abstract Interpretation of Epistemic Logic	3
Mai Ajspur and Valentin Goranko / Tableaux-based decision method for single-agent linear time synchronous temporal epistemic logics with interacting time and knowledge	6
Staffan Angere / What is Semantics (really) about?.....	9
Philippe Balbiani / Measuring the distance between modal formulas	12
Benno van den Berg / The Herbrand Topos	15
Denis Bonnay and Frederik Engström / Invariance and definability, with or without equality	18
Eyvind Briseid / On a functional interpretation for nonstandard arithmetic	21
Henning Christiansen / An exposition of abductive reasoning through logic programming with constraints	23
Pavlos Eirinakis, Salvatore Ruggieri, K. Subramani and Piotr Wojciechowski / Quantified Linear Programs and Quantified Linear Implications	26
Sebastian Enqvist / A Lindström theorem for some normal modal logics	29
Camillo Fiorentini / Terminating sequent calculi for proving and refuting formulas in $S4$	32
Henrik Forssell / Recovering theories from their models	35
Pietro Galliani / The Doxastic Interpretation of Team Semantics	38
Valentin Goranko and Ruaan Kellerman / First-order theories of classes of trees associated with classes of linear orders	40
Jens Ulrik Hansen / Modeling pluralistic ignorance in epistemic logic	43
Casper Storm Hansen / Brouwer's Free Choice Sequences and Bivalence	46
Lauri Hella / Weak Models of Distributed Computing and Modal Logic	49
Thomas Hildebrandt / Dynamic Condition Response Graphs	52
Åsa Hirvonen / Generalized Isomorphisms in metric models theory	55
Tudor Jebelean / Analysis and Synthesis of Algorithms in Theorema	56
Andreas Schmidt Jensen and Jørgen Villadsen / Paraconsistent Computational Logic	59
Juliette Kennedy / Change the logic, change the meaning? Logical pluralism in the case of set theory	62
Vera Koponen / Logical limit laws of graphs subject to constraints	63
Antti Kuusisto, Jeremy Meyers and Jonni Virtema / Undecidable First-Order Theories of Affine Geometries	66

Kim Guldstrand Larsen, Radu Mardare and Prakash Panangaden / Stone duality for Markov processes	69
François Lepage / Intuitionistic logic and partial probability functions	71
Roussanka Loukanova / Algorithmic Intensionality in Type Theory of Acyclic Recursion and Underspecification	74
Kerkko Luosto / Unary quantifiers on finite and infinite structures	77
Mathias Winther Madsen / Multiplayer Game Semantics and Non-Polemic Conversation	78
María Manzano and Enrique Alonso / Henkin on Completeness	81
María Manzano and Manuel Crescencio Moreno / All identicals are equal, but some equals are more equal than others	84
Rasmus Ejlers Møgelberg / The enriched effect calculus – a calculus for linear usage of effects	87
Larry Moss / New Logical Systems Based on Natural Language	90
Peter Øhrstrøm / A.N. Prior's Contributions to the Rise of Temporal Logic in the 1950s and 1960s: New Insights Based on the Study of Priors Nachlass	93
Ulf Persson / Mathematics and Logic	95
Rasmus Rendsvig / Qualitative Modeling of Informational Cascades	97
Gemma Robles / Two-valued semantics for Łukasiewicz's 3-valued logic L_3	100
Olivier Roy and Ole Thomassen Hjortland / Dynamic Consequences for Soft Information	103
Francisco Salto, Gemma Robles and José M. Méndez / Deep relevant logics not included in relevant logic R	107
Dimiter Vakarelov / Dynamic modalities	110
Bartosz Więckowski / Remarks on proof-theoretic semantics of intensional transitive verbs	113

A zero-one law for l -colourable structures with a vectorspace pregeometry

Ove Ahlman, Uppsala University

1 Introduction

Model theory is the study of abstract mathematical structures, their formal languages and their theories. This talk is based on an article (not submitted yet) by the speaker and Koponen [1], and will be focused on finite models and what happens with them when they get arbitrary large. Especially we consider formulas which are satisfied in almost all structures or in almost no structures. Glebskii et. al. [3] and Fagin [2] answered this independently of each other for sets of finite relational structures by giving a so called 0 – 1 law for the uniform probability measure. If for each $n \in \mathbb{N}$, \mathbf{K}_n is a set of L -structures of size n , we say that $\mathbf{K} = \bigcup_{n=1}^{\infty} \mathbf{K}_n$ has a 0 – 1 law for the probability measure μ_n defined on formulas on \mathbf{K}_n , if for each $\varphi \in L$

$$\lim_{n \rightarrow \infty} \mu_n(\varphi) = 1 \quad \text{or} \quad \lim_{n \rightarrow \infty} \mu_n(\varphi) = 0.$$

The 0 – 1 law which Fagin [2] and Glebskii et. al. [3] proved, considered all finite structures over a certain relational language, so researchers asked themselves how we could restrict the sets of structures in different ways and still have a 0 – 1 law. In this talk we consider, for fixed $l \geq 2$, l -colourable structures. That is, we consider structures whose universe can be partitioned into l parts in such a way that every relationship of the structure intersects at least two parts. Kolaitis, Prömel and Rothschild [5] showed, as a part of their proof that K_{l+1} -free graphs ($l \geq 2$) has a 0 – 1 law for the uniform probability measure, that a 0 – 1 law holds for l -colourable graphs. The question may arise if such a 0 – 1 law is possible to generalise to any l -colourable structures. In the general case we may have relation symbols of higher arity than 2 in the formal language and then there are two natural ways of generalising l -colourings and l -colourability; the “strong” and the “weak” versions of l -colourings.

Koponen [6] showed that both strongly and weakly l -colourable structures have a 0 – 1 law for both the uniform probability measure and for the dimension conditional measure (defined in [6]). A consequence is that if you have sets of L -structures \mathbf{K}_n , $n = 1, 2, 3, \dots$ where each $\mathcal{M} \in \mathbf{K}_n$ has universe $\{1, \dots, n\}$ and a) each l -colourable L -structure with universe $\{1, \dots, n\}$ is in \mathbf{K}_n and b) “almost all” $\mathcal{M} \in \mathbf{K}_n$ are l -colourable (for big n), then $\mathbf{K} = \bigcup_{n=1}^{\infty} \mathbf{K}_n$ has a 0 – 1 law. In [7], Schacht and Person let \mathbf{K}_n be the set of all 3-hypergraphs without Fano planes and node-set $\{1, \dots, n\}$, and show that almost all such hypergraphs are 2-colourable. Since each 2-colourable 3-hypergraph is missing a Fano plane it follows that \mathbf{K} in this case has a 0 – 1 law.

One of the most fundamental and important mathematical structures are vector spaces (as well as affine and projective spaces) which in turn induce so called pregeometries. Pregeometries play an important part in model theory. It is therefore natural to study sets \mathbf{K}_n , $n = 1, 2, \dots$ of L -structures (for some fixed language L) which have an underlying pregeometry, definable by L -formulas. In

particular, one may consider structures with an underlying pregeometry that, in addition, have an l -colouring which respects the pregeometry.

In this talk we explore strongly and weakly l -colourable L -structures whose underlying pregeometry is a vector space (of finite dimension) over a fixed finite field. We will show that both strongly and weakly l -colourable L -structures have a 0 – 1 law for the “dimension conditional” probability measure, which generalises Theorem 9.1 in [6]. The dimension conditional measure has a natural interpretation as a process where you first randomly choose an l -colouring c on each finite dimensional vector space, then randomly choose relations on the 1-dimensional subsets, then on the 2-dimensional subsets (among those possibilities for which c is still an l -colouring) etc. for each r such that some relation symbol has at least the arity $d \geq r$. The proof idea is to define certain “extension axioms” and to show that each such almost surely is true in an l -colourable structure with big enough dimension. To do this we need a formula $\xi(x, y)$, such that with probability approaching one as the dimension tends to infinity, two elements a and b have the same colour if and only if $\xi(a, b)$ holds in the given structure. Moreover, it is essential that $\xi(x, y)$ does not explicitly mention the colours; it only speaks about the relations of the structure and the pregeometry. In the case of strong l -colourings, this will be done in an explicit way. While when we speak of weak l -colourings, the strong colouring method doesn’t work. Instead we seek aid in a result from Ramsey theory and a theorem by Graham, Leeb and Rothschild [4] which is about colouring vector spaces over an arbitrary finite field. This result shows that a formula ξ , as we said we needed above, exists but without exactly mentioning what ξ looks like.

References

- [1] O. Ahlman, V. Koponen, Random l -colourable structures with a pregeometry, Preprint.
- [2] R. Fagin, Probabilities on finite model theory, J. Symbolic Logic 41 (1976), no. 1, 55-58.
- [3] Y. V. Glebskii, D. I. Kogan, M.I. Liogonkii, V.A. Talanov, Volume and fraction of Satisfiability of formulas of the lower predicate calculus, Kibernetika Vol. 2 (1969) 17-27.
- [4] R. L. Graham, K. Leeb, B. L. Rothschild, Ramsey’s theorem for a class of categories, Advances in Mathematics, Vol. 8 (1972) 417-433.
- [5] Ph. G. Kolaitis, H. J. Prömel, B. L. Rothschild, K_{l+1} -free graphs: Asymptotic structure and a 0-1 law, Transactions of the American Mathematical Society, volume 303, number 2, Oct 1987, 637-671.
- [6] V. Koponen, Asymptotic probabilities of extension properties and random l -colourable structures, Annals of Pure and Applied Logic, Vol. 163 (2012) 391-438.
- [7] Y. Person, M. Schacht, Almost all hypergraphs without Fano planes are bipartite, SODA ’09 Proceedings of the twentieth annual ACM-SIAM Symposium on discrete mathematics, Society for industrial and applied mathematics, Philadelphia, 2009.

Towards Abstract Interpretation of Epistemic Logic

Mai Ajspur and John P. Gallagher

CBIT, Building 43.2, Roskilde University, 4000 Roskilde, Denmark
{ajspur,jpg}@ruc.dk

Abstract. The model-checking problem is to decide, given a formula ϕ and an interpretation M , whether M satisfies ϕ , written $M \models \phi$. Model-checking algorithms for temporal logics were initially developed with finite models (such as models of hardware) in mind so that $M \models \phi$ is decidable. As interest grew in model-checking infinite systems, other approaches were developed based on approximating the model-checking algorithm so that it still terminates with some useful output.

In this work we present a model-checking algorithm for a multiagent epistemic logic containing operators for common and distributed knowledge. The model-checker is developed as a function directly from the semantics of the logic, in a style that could be applied straightforwardly to derive model-checkers for other logics. Secondly, we consider how to abstract the model-checker using abstract interpretation, yielding a procedure applicable to infinite models. The abstract model-checker allows model-checking with infinite-state models. When applied to the problem of whether $M \models \phi$, it terminates and returns the set of states in M at which ϕ might hold. If the set is empty, then M definitely does not satisfy ϕ , while if the set is non-empty then M possibly satisfies ϕ .

1 Syntax and semantics of the logic CMAEL(CD)

We consider the logic **CMAEL(CD)** [1, 7] whose formulas $\phi \in \Phi$ are defined by the following grammar.

$$\varphi ::= p \mid \neg\varphi \mid (\varphi_1 \wedge \varphi_2) \mid \mathbf{D}_A\varphi \mid \mathbf{C}_A\varphi.$$

The variable p ranges over the set **AP** of *atomic propositions*, typically denoted by p, q, r, \dots ; the variable A ranges over the set of *coalitions* $\mathcal{P}^+(\Sigma)$, which is the set of non-empty subsets of Σ , where Σ is a finite, non-empty set of (names for) *agents*, typically denoted by a, b, \dots . The epistemic operators \mathbf{D}_A and \mathbf{C}_A are read as *it is distributed knowledge among A that ...* and *it is common knowledge among A that ...* respectively. When A is a singleton $\{a\}$ we often write it as a subscript a instead of $\{a\}$, for example \mathbf{D}_a instead of $\mathbf{D}_{\{a\}}$.

The semantics of **CMAEL(CD)** is given in terms of coalitional multiagent epistemic models (CMAEMs). A CMAEM is a tuple $(\Sigma, S, \{\mathcal{R}_A^D\}_{A \in \mathcal{P}^+(\Sigma)}, \{\mathcal{R}_A^C\}_{A \in \mathcal{P}^+(\Sigma)}, L)$,

1. Σ is a finite, non-empty set of *agents*;
2. $S \neq \emptyset$ is a set of *states*;
3. for every $A \in \mathcal{P}^+(\Sigma)$, \mathcal{R}_A^D is an equivalence relation on S , satisfying the condition $\mathcal{R}_A^D = \bigcap_{a \in A} \mathcal{R}_a^D$;
4. for every $A \in \mathcal{P}^+(\Sigma)$, \mathcal{R}_A^C is the transitive closure of $\bigcup_{a \in A} \mathcal{R}_a^D$;
5. $L : S \mapsto \mathcal{P}(\mathbf{AP})$ is a *labelling function*, assigning to every state s the set $L(s)$ of atomic propositions true at s .

Let S be a set. We define functions $pre : ((S \times S) \times \mathcal{P}(S)) \rightarrow \mathcal{P}(S)$ and $\widetilde{pre} : ((S \times S) \times \mathcal{P}(S)) \rightarrow \mathcal{P}(S)$.

- $pre(\mathcal{R})(X) = \{s \mid \exists s' \in X : (s, s') \in \mathcal{R}\}$ returns the set of states having at least one of their successors (in relation \mathcal{R}) in the set $X \subseteq S$;

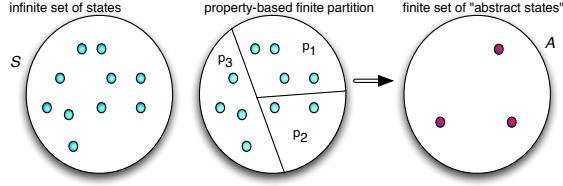


Fig. 1. Property Based Abstraction

– $\widehat{pre}(\mathcal{R})(X) = S \setminus pre(\mathcal{R})(S \setminus X)$ returns the set of states all of whose successors are in X . The functions pre and \widehat{pre} are defined by several authors (e.g. [6, 9]) and are also used with other names by other authors (e.g. they are called pre_{\exists} and pre_{\forall} by Huth and Ryan [8]).

*Semantic Function for **CMAEL(CD)**.* Let M be a CMAEM with states S ; the following function $\llbracket \cdot \rrbracket_M : \Phi \rightarrow \mathcal{P}(S)$ evaluates to the set of states of M where ϕ is true.

$$\begin{aligned} \llbracket p \rrbracket_M &= \{s \mid p \in L(s)\} & \llbracket \neg \phi \rrbracket_M &= S \setminus \llbracket \phi \rrbracket_M & \llbracket \phi_1 \wedge \phi_2 \rrbracket_M &= \llbracket \phi_1 \rrbracket_M \cap \llbracket \phi_2 \rrbracket_M \\ \llbracket \mathbf{D}_A \phi \rrbracket_M &= \widehat{pre}(\mathcal{R}_A^D)(\llbracket \phi \rrbracket_M) & \llbracket \mathbf{C}_A \phi \rrbracket_M &= \widehat{pre}(\mathcal{R}_A^C)(\llbracket \phi \rrbracket_M) \end{aligned}$$

(The set complement operator can be eliminated to avoid technical problems later when abstracting the function.) This is closely related to the standard semantic relation $M, s \models \phi$ (ϕ holds at state s in M), which can be rewritten as $s \in \llbracket \phi \rrbracket_M$. Note that given a model M and formula ϕ , the calculation of the value of $\llbracket \phi \rrbracket_M$ is simple and does not involve the computational problems associated with the proof theory of distributed and common knowledge.

1.1 Abstract Interpretation of CMAEL(CD)

What does it mean to perform “abstract model checking”? Informally, we check the satisfiability of a formula in a possibly infinite model using partial knowledge of the model. The abstract interpretation framework [5] ensures that the result of the check is safe, in that checking returns false only when the formula is not satisfied by the model. A typical abstraction is based on a finite set of properties of interest $\{p_1, \dots, p_k\}$ (see Figure 1), for example those atomic propositions appearing in the formula to be checked.¹ Suppose we have a model whose set of states S is infinite and in every state in S , exactly one p_i holds. Then the finite partition $A = \{d_1, \dots, d_k\}$ is defined such that $d_i = \{s \in S \mid p_i \in L(s)\}$. An abstract interpretation can be constructed from lattices $\langle \mathcal{P}(S), \subseteq \rangle$ and $\langle \mathcal{P}(A), \subseteq \rangle$ called the “concrete” and “abstract” domain respectively, and a *Galois connection* relating them. The Galois connection consists of monotonic functions $\alpha : \mathcal{P}(C) \rightarrow \mathcal{P}(A)$ and $\gamma : \mathcal{P}(A) \rightarrow \mathcal{P}(C)$ such that $\forall c \in \mathcal{P}(C), a \in \mathcal{P}(A), \alpha(c) \subseteq a \Leftrightarrow c \subseteq \gamma(a)$. In property-based abstractions, α and γ are defined as $\alpha(X) = \{d_i \mid d_i \cap X \neq \emptyset\}$ and $\gamma(Y) = \bigcup Y$. The concrete semantic function is $\llbracket \cdot \rrbracket_M : \Phi \rightarrow \mathcal{P}(S)$ defined above. Given these components, the framework of abstract interpretation shows how to derive an abstract semantic function $\llbracket \cdot \rrbracket_M^\# : \Phi \rightarrow \mathcal{P}(A)$ systematically from $\llbracket \cdot \rrbracket_M$ such that $\llbracket \phi \rrbracket_M \subseteq \gamma(\llbracket \phi \rrbracket_M^\#)$, for all $\phi \in \Phi$ (or equivalently, $\alpha(\llbracket \phi \rrbracket_M) \subseteq \llbracket \phi \rrbracket_M^\#$). Thus in property-based abstract interpretation, the abstract semantic function returns a set of partitions (an element of $\mathcal{P}(A)$). The union of this set of partitions is a superset of the set of concrete states returned by the concrete semantic function. In particular, if $\llbracket \phi \rrbracket_M^\#$ is empty for some ϕ , then the result of the concrete computation $\llbracket \phi \rrbracket_M$ is also empty.

¹ Abstract interpretation is not limited to this kind of abstraction.

Remarks. In some approaches to abstract model checking [3, 4], a partition of the set of states is used to induce “abstract relations” in an “abstract model”. This is not our approach, since it cannot simultaneously approximate both a formula and its negation. By contrast, the abstract semantic function shown below always returns an over-approximation of the set of states where any given formula holds. Other authors have developed “dual” approximations based on abstract relations to overcome the limitations of abstract models, but the framework of abstract interpretation offers a more direct solution, and was previously applied successfully to abstract model checking for CTL [2].

1.2 Abstract Semantic Function

The function $\llbracket \cdot \rrbracket_M^\sharp : \Phi \rightarrow \mathcal{P}(A)$ is defined systematically from the concrete function $\llbracket \phi \rrbracket_M : \Phi \rightarrow \mathcal{P}(S)$ and the functions α and γ . We show it below. The transformation consists only of applying α to the base cases $\{s \mid p \in L(s)\}$ and $\{s \mid p \notin L(s)\}$ and replacing $pre(\cdot)(\cdot)$ (resp. $\widehat{pre}(\cdot)(\cdot)$) by $\alpha(pre(\cdot)(\gamma(\cdot)))$ (resp. $\alpha(\widehat{pre}(\cdot)(\gamma(\cdot)))$).

$$\begin{array}{ll} \llbracket p \rrbracket_M^\sharp &= \alpha(\{s \mid p \in L(s)\}) & \llbracket \neg p \rrbracket_M^\sharp &= \alpha(\{s \mid p \notin L(s)\}) \\ \llbracket \phi_1 \wedge \phi_2 \rrbracket_M^\sharp &= \llbracket \phi_1 \rrbracket_M^\sharp \cap \llbracket \phi_2 \rrbracket_M^\sharp & \llbracket \neg(\phi_1 \wedge \phi_2) \rrbracket_M^\sharp &= \llbracket \neg \phi_1 \rrbracket_M^\sharp \cup \llbracket \neg \phi_2 \rrbracket_M^\sharp \\ \llbracket \mathbf{D}_A \phi \rrbracket_M^\sharp &= \alpha(\widehat{pre}(\mathcal{R}_A^D)(\gamma(\llbracket \phi \rrbracket_M^\sharp))) & \llbracket \neg(\mathbf{D}_A \phi) \rrbracket_M^\sharp &= \alpha(pre(\mathcal{R}_A^D)(\gamma(\llbracket \neg \phi \rrbracket_M^\sharp))) \\ \llbracket \mathbf{C}_A \phi \rrbracket_M^\sharp &= \alpha(\widehat{pre}(\mathcal{R}_A^C)(\gamma(\llbracket \phi \rrbracket_M^\sharp))) & \llbracket \neg(\mathbf{C}_A \phi) \rrbracket_M^\sharp &= \alpha(pre(\mathcal{R}_A^C)(\gamma(\llbracket \neg \phi \rrbracket_M^\sharp))) \\ & & \llbracket \neg \neg \phi \rrbracket_M^\sharp &= \llbracket \phi \rrbracket_M^\sharp \end{array}$$

Proposition 1. For all formulas $\phi \in \Phi$, and CMAEM M , $\llbracket \phi \rrbracket_M \subseteq \gamma(\llbracket \phi \rrbracket_M^\sharp)$.

Proof. Proof is by structural induction on the formula ϕ and uses the properties of Galois connections and monotonic functions.

References

1. M. Ajspur, V. Goranko, and D. Shkatov. Tableau-based decision procedure. *CoRR*, abs/1201.5346, 2012.
2. G. Banda and J. P. Gallagher. Constraint-based abstract semantics for temporal logic: A direct approach to design and implementation. In E. M. Clarke and A. Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning - 16th International Conference, LPAR-16, Dakar, Senegal, April 25-May 1, 2010, Revised Selected Papers*, volume 6355 of *Lecture Notes in Computer Science*, pages 27–45. Springer, 2010.
3. E. M. Clarke, O. Grumberg, and D. E. Long. Model checking and abstraction. In *Conference Record of the Nineteenth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Albuquerque, New Mexico, USA*, pages 342–354, 1992.
4. E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. MIT Press, 2000.
5. P. Cousot and R. Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proceedings of the 4th ACM Symposium on Principles of Programming Languages, Los Angeles*, pages 238–252, 1977.
6. P. Cousot and R. Cousot. Temporal abstract interpretation. In *Proceedings of the 27th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, January 19 - 21 2000, Boston, Mass. USA*, pages 12–25, 2000.
7. V. Goranko and D. Shkatov. Tableau-based procedure for deciding satisfiability in the full coalitional multiagent epistemic logic. *CoRR*, abs/0902.2125, 2009.
8. M. R. A. Huth and M. D. Ryan. *Logic in Computer Science: Modelling and reasoning about systems*. Cambridge University Press, 2000.
9. P. Kelb. Model checking and abstraction: A framework preserving both truth and failure information. Technical report, Carl von Ossietzky Univ. of Oldenburg, Oldenburg, Germany, 1994.

Tableaux-based decision method for single-agent linear time synchronous temporal epistemic logics with interacting time and knowledge

(Extended abstract)

Mai Ajspur¹ and Valentin Goranko²

¹ Roskilde University, ajspur@ruc.dk,

² Technical University of Denmark, vfgo@imm.dtu.dk

Knowledge and time are among the most important aspects of agency. Various *temporal-epistemic logics*, proposed as logical frameworks for reasoning about these aspects of agents and multi-agent systems were actively studied in a number of publications during the 1980's, eventually summarized and uniformly presented in a comprehensive study by Halpern and Vardi [3]. In [3], the authors considered several essential characteristics of temporal-epistemic logics: *one vs. several agents*, *synchrony vs. asynchrony*, *(no) learning*, *(no) forgetting (aka, perfect recall or no recall)*, *linear vs. branching time*, and existence (or not) of a *unique initial state*. Based on these, they identified and analyzed a total of 96 temporal-epistemic logics and obtained lower bounds for the complexity of a satisfiability problem in them. In [4] matching upper bounds were claimed for all, and established for most of these logics. It turned out that most of the logics that involve more than one agents, whose *knowledge interacts with time* (e.g., who do not learn or do not forget) – are undecidable (with common knowledge), or decidable but with non-elementary time lower bound (without common knowledge). Even in the single-agent case the interaction between knowledge and time proved to be quite costly, pushing the complexities of deciding satisfiability up to EXPSpace and 2EXPTIME.

In this work we develop tableau-based procedure for deciding satisfiability in single-agent synchronous temporal-epistemic logics with interactions between time and knowledge, by essentially expanding and adapting the incremental tableau construction, described in [2] for multi-agent temporal-epistemic logics with common and distributed knowledge, but with no interactions between time and knowledge (other than synchrony).

Our method is theoretically optimal and practically implementable (modulo the established complexities, of course), unlike the theoretically optimal decision method by Halpern and Vardi in [4], based on essentially combinatorial estimates of the size 'small' tree-like models satisfying models. We note, that a non-optimal (2EXPSpace) tableau-based method for some of the logics covered here, requiring pre-processing of the input formulas into a special clausal normal forms, was developed in [1].

1 The Single-agent Linear Time Temporal Epistemic Logic $\text{TEL}^1(\text{LT})$

We only give the very basic notions here. For further details the reader is referred to [3], [4], [2]. The language of $\text{TEL}^1(\text{LT})$ contains a set AP of atomic propositions, the Booleans \neg ("not") and \wedge ("and"), the temporal operators \mathcal{X} ("next") and \mathcal{U} ("until") of the logic LTL, as well as the epistemic operator \mathbf{K} . The formulas of $\text{TEL}^1(\text{LT})$ are defined as follows:

$$\varphi := p \mid \neg\varphi \mid (\varphi_1 \wedge \varphi_2) \mid \mathcal{X}\varphi \mid (\varphi_1 \mathcal{U} \varphi_2) \mid \mathbf{K}\varphi$$

where p ranges over AP . All other standard Boolean and temporal connectives can be defined as usual. Formulas of the type $\mathbf{K}\varphi$ or their negations will be called *knowledge formulas*.

Temporal-epistemic frames and models A (*single-agent*) *temporal-epistemic frame* (TEF) is a tuple $\mathfrak{S} = (S, R, \mathcal{R})$, where $S \neq \emptyset$ is a set of *states*; $R \subseteq S^{\mathbb{N}}$ is a non-empty set of *runs*; and $\mathcal{R} \subseteq (R \times \mathbb{N})^2$ is an equivalence relation, representing the epistemic uncertainty of the agent. A *temporal-epistemic model* (TEM) is a pair $\mathcal{M} = (\mathfrak{S}, L)$, where \mathfrak{S} is a TEF and $L : R \times \mathbb{N} \rightarrow \mathcal{P}(\text{AP})$ is a *labeling function*. We denote $P(\mathfrak{S}) := R \times \mathbb{N}$. The elements of $(r, n) \in P(\mathfrak{S})$ are called *points*.

Truth and satisfiability Truth of formulas at a point of a TEM is defined recursively as usual, by combining the semantics for LTL and that of the standard epistemic logic:

- $\mathcal{M}, (r, n) \models \mathcal{X}\varphi$ iff $\mathcal{M}, (r, n+1) \models \varphi$;
- $\mathcal{M}, (r, n) \models \varphi \mathcal{U}\psi$ iff $\mathcal{M}, (r, i) \models \psi$ for some $i \geq n$ such that $\mathcal{M}, (r, j) \models \varphi$ for every $n \leq j < i$;
- $\mathcal{M}, (r, n) \models \mathbf{K}\varphi$ iff $\mathcal{M}, (r', n') \models \varphi$ for every $((r, n), (r', n')) \in \mathcal{R}$;

A formula φ is *satisfiable* (resp., *valid*) if $\mathcal{M}, (r, n) \models \varphi$ for some (resp., every) TEM \mathcal{M} and a point (r, n) in it. Satisfiability and validity in a class of models is defined likewise.

Properties of TEF and TEM A TEF $\mathfrak{G} = (S, R, \mathcal{R})$ has the property of:

- **Indistinguishable_Initial_States** (iis), if for all runs $r, r' \in R$, $((r, 0), (r', 0)) \in \mathcal{R}$.
- **No_Learning** (nol), if whenever $((r, n), (r', n')) \in \mathcal{R}$, for every $k \geq n$ there exists a $k' \geq n'$ such that $((r, k), (r', k')) \in \mathcal{R}$ (the agent does not learn over time);
- **No_Forgetting** (nof), if whenever $((r, n), (r', n')) \in \mathcal{R}$, for every $0 \leq k \leq n$ there exists a $0 \leq k' \leq n'$ such that $((r, k), (r', k')) \in \mathcal{R}$ (the agent does not forget over time);
- **Synchrony** (sync), if $((r, n), (r', n')) \in \mathcal{R}$ implies $n = n'$ (the agent can perceive time).

For any $X \in \{\text{nol}, \text{nof}, \text{sync}, \text{iis}\}$, \mathcal{M} has the property X if \mathfrak{G} does so. For $X \subseteq \{\text{nol}, \text{nof}, \text{sync}, \text{iis}\}$, we denote the classes of all TEMs satisfying the properties in X by TEM_X . We then denote the extension of the logic $\text{TEL}^1(\text{LT})$ with semantics restricted to the class TEM_X by $\text{TEL}^1(\text{LT})_X$.

In this paper we will focus on $\text{TEL}^1(\text{LT})_X$, where $\text{sync} \in X$ and either $\text{nol} \in X$ or $\text{nof} \in X$.

Temporal Epistemic Hintikka Structures Even though we are ultimately interested in testing formulas of $\text{TEL}^1(\text{LT})$ for satisfiability in a TEM, the tableau procedure we will present here tests for satisfiability in a more general kind of semantic structures, namely *temporal epistemic Hintikka structures* (TEHS). A Hintikka structure for a formula θ contains just as much semantic information about the satisfying model of θ as it is necessary. It can be shown that any $\text{TEL}^1(\text{LT})$ -formula is satisfiable in a TEM iff it is satisfiable in a TEHS.

2 Tableaux for synchronous $\text{TEL}^1(\text{LT})$ with interaction conditions

Overview of the tableau procedure for $\text{TEL}^1(\text{LT})_\emptyset$ The tableaux procedure for $\text{TEL}^1(\text{LT})_\emptyset$ as developed for the multi-agent case in [2] is used here as a starting point for the procedure for $\text{TEL}^1(\text{LT})_X$ where $X \neq \emptyset$. The procedure for $\text{TEL}^1(\text{LT})_\emptyset$ constructs a directed graph \mathcal{T}^θ (called itself a *tableau*); its nodes are either *states* or *prestates*, while there are three types of directed edges between its nodes, representing temporal, epistemic, or label-expansion relations. States are labelled with fully expanded sets of formulas and represent points of a TEHS (eventually points of a model), while prestates (labelled with sets of formulas) play a temporary role in the construction of \mathcal{T}^θ . The procedure consists of three major phases: First prestates, states and relations are added to the tableau, by alternating between constructing epistemic and temporal successor prestates of a given state, and expanding a given prestate Γ into fully expanded sets ($\text{states}(\Gamma)$). Secondly, all prestates are eliminated from the tableau under construction, and accordingly edges from and to the prestates are redirected. In the last phase, all ‘bad’ states and relations are removed in successive steps; ‘bad’ states cannot be satisfied in a TEHS, because they either lack necessary successors (epistemic or temporal) or they contain unrealized eventualities. The resulting graph is the *final tableau* \mathcal{T}^θ for the input formula θ . If some state Δ of \mathcal{T}^θ contains θ , the procedure declares θ satisfiable and a TEHS satisfying θ can be extracted from it; otherwise it declares it unsatisfiable.

Complications arising with interacting temporal and epistemic operators In the tableau construction described above, the label of any temporal successor-prestate Γ for a state Δ only depends on formulas that come directly from Δ . When the logic has time-knowledge interaction, e.g. **nol**, this is no longer the case since formulas coming from other states that are epistemically related to Δ will be relevant for Γ , too. When the logic has **nof** and a state is epistemically related to another state, then both states need to have predecessor-states which are also epistemically related. Therefore, the procedure has to create enough states at any temporal layer, so that the states needed in the next temporal layer have predecessor states.

Construction of the tableau for $TEL^1(LT)_X$ In order to deal with these complications, the tableau procedure presented here will act *not on single states* but on special kinds of sets of states representing possible epistemic clusters, which we will call *bubbles*. The purpose of the tableau is to eventually create infinite paths of bubbles from which a satisfying TEM can be constructed. The bubbles will correspond to be the equivalence classes (w.r.t. \mathcal{R}) of the TEM.

For the logic $TEL^1(LT)_X$ where $\text{sync}, \text{nof} \in X$ or $\text{sync}, \text{no} \in X$, the procedure is again split into three phases. First, the procedure constructs all necessary bubbles and pre-bubbles, by alternately applying special expansion procedures that constructs the necessary temporal successor pre-bubbles of a bubble, and expand pre-bubbles into bubbles, respectively. The procedure does not produce epistemic successor pre-bubbles of a bubble, since the bubbles in themselves contain enough epistemic alternatives for any state in it. Secondly, the procedure removes all pre-bubbles and redirects the incoming and outgoing arrows to and from prebubbles. Finally, the procedure build the final tableau by repeatedly checking and eliminating ‘bad bubbles’ that lack necessary temporal successors or contain states with unrealized eventualities, until stabilization. The final tableau is open if the input formula belongs to a state in the label of a bubble in it. Otherwise it is closed and the input formula is declared unsatisfiable.

Theorem 1. *The tableaux procedure for each $TEL^1(LT)_X$ is sound and complete.*

Soundness of the tableaux method means that if the input formula θ is satisfiable, then the procedure will indeed produce an open tableau. The argument in a nutshell is that if θ is satisfiable, then there is also a satisfiable state Δ in $\text{states}(\{\theta\})$, and the pre-bubble $\{\Delta\}$ will be expanded to a number of bubbles. At least one of them ‘survives’ in the final tableau.

For completeness, we argue that if the procedure returns an open tableau for the input formula θ , then we can construct a TEHS from this open tableau. This TEHS is eventually transformed into a TEM with the required time-knowledge interaction properties and satisfying θ . As mentioned, the tableau method ensures the existence of some infinite paths of bubbles. These bubbles will correspond to equivalence classes in the TEM, while the runs successively will pass through states in successive bubbles in the paths.

Theorem 2. *The tableaux procedure for each $TEL^1(LT)_X$ runs in 2EXPTIME. Furthermore, for all $TEL^1(LT)_X$ where $\text{nof} \in X$, the tableaux-procedure can be modified to work in EXPSPACE.*

We note that the number of possible (different) states in the tableaux, $\#sts$, is exponential in the length of the input formula θ , which means that the possible number of bubbles in the tableau is $2^{\#sts}$, i.e. double exponential in the length of θ .

The 2EXPTIME upper bound is obtained by estimating the number of steps in every phase of the construction, which is polynomial in the number of possible bubbles in the tableau. The optimization to NEXPSpace=EXPSPACE is done by guessing and verifying on-the-fly a satisfying bubble-path in the open tableau. The argument is similar to the one proving PSPACE of the tableaux method for LTL, with an ‘exponential step’ added to it.

We claim that the method is amenable to cover extensions to asynchronous systems, multi-agent and branching-time temporal epistemic logics. These are in the future agenda of this project.

References

1. Clare Dixon, Cláudia Nalon, and Michael Fisher. Tableaux for logics of time and knowledge with interactions relating to synchrony. *Journal of Applied Non-Classical Logics*, 14(4):397–445, 2004.
2. Valentin Goranko and Dmitry Shkatov. Tableau-based decision procedure for full coalitional multiagent temporal-epistemic logic of linear time. In Decker, Sichman, Sierra, and Castelfranchi, editors, *Proc. of AAMAS’2009*, 2009.
3. Joseph Y. Halpern and Moshe Y. Vardi. The complexity of reasoning about knowledge and time I: Lower bounds. *Journal of Computer and System Sciences*, 38(1):195–237, 1989.
4. Joseph Y. Halpern and Moshe Y. Vardi. Reasoning about knowledge and time: Synchronous systems, 1989. IBM Research Report,.

What is Semantics (really) about?

Abstract for the Scandinavian Logic Symposium 2012

Staffan Angere
University of Bristol

This is a two-part talk. I will first make some general observations on the practice of doing semantics of formal languages. Here, I argue that semantic notions, such as consequence, objectual quantification, and (in)completeness, are reducible to syntactic concepts, and that these syntactic concepts actually give us a more enlightening view of logic than the one that is usually taught.

In the second part, I will describe how these remarks, and the picture of semantics they indicate, can be put on a firm basis through use of the concept of monoidal 2-category. In the talk, I will presuppose basic knowledge of “plain” category theory, but no specific knowledge of monoidal categories or 2-categories.

I end with some remarks on further applications of the framework: it can, for instance, be used to clarify the distinction between universal quantification and conjunction, as well as the status of logical constants.

1 Some Observations on Semantics

Semantics, as it is usually taught, depends on associating mathematical structures with syntactically defined objects: in FOL, elements of a domain D with terms, relations on D with predicates, and sets of sequences of elements of D with formulae. Tarski’s recursive truth definition allows us to derive the third of these associations from the two first. It is used in rigorous metalogical proofs of standard results such as the Löwenheim-Skolem theorem and the completeness theorem.

But what is actually going on in such a proof? Do we, when proving completeness, look into the abstract realm of mathematics, and see that for every non-contradictory set of sentences, there is indeed a set which is its model? Certainly not—if the proof is valid, we could, in principle, have done it formally. We could derive the existence of a certain set (the model) from the axioms of ZFC.

Generalizing, my conjecture is that what is “really” going on when we do semantic reasoning in logic is that we translate a piece of the object language into a suitable metalanguage, and we then work in this metalanguage, and, if necessary, translate back in the end. As a conjecture, it may not seem very substantial, but I believe that, correctly interpreted, it implies many important points:

- *Objectual quantification* in the object language is (really) substitutional quantification in the metalanguage.
- *Identity* in the object language is (really) substitutability *salva veritate* in the metalanguage.
- *Logical consequence* in the object language is (really) derivability in the metalanguage.

- *Satisfaction* is reducible to truth in the metalanguage.

To substantiate these claims, we would naturally need to carry out the required reductions in detail. It will unfortunately be impossible to do so in a talk, but I will sketch how it may be accomplished, using a category-based formalism.

2 Paradigms: a Formal Semantics of Formal Languages

A *2-category* is a category which, in addition to the objects and the arrows, has a set of *2-cells* for each ordered pair of arrows of the same hom-set. These 2-cells describe *transformations between arrows*, and are required to fulfill certain commutativity conditions. The classical example is the 2-category of categories, with categories as objects, functors as arrows, and natural transformations as 2-cells.

A *monoidal* category is a category equipped with a binary operation \otimes on objects and arrows (and in the 2-category case, on 2-cells) which is bifunctorial (roughly, functorial in both arguments). For coherence and associativity, \otimes is required to fulfil a few conditions such as MacLane's *pentagon equation*. An example of a monoidal category is any category with products, in which we may take $A \otimes B$ to be any product of A and B , and $f \otimes g$, where $f : A \rightarrow C$ and $g : B \rightarrow D$, to be the unique morphism from $A \otimes B$ to $C \otimes D$ that this product determines. Monoidal categories are also very useful in physics, where objects are often taken to be Hilbert spaces, and \otimes the tensor product.

We can describe a *language* as a monoidal 2-category with (i) *types* as objects, (ii) *formation rules* as arrows, (iii) *inference rules* as 2-cells, and (iv) *ordered pairing* as \otimes . We require every language to have a type of *formulae* Ω , and a *terminal type* 1 (i.e. a terminal object in the category) so that we can represent individual well-formed formulae as arrows $1 \rightarrow \Omega$. A *formal* language is a language which is in an appropriate sense *free* as such a 2-category, i.e. it is generated from a graph, possibly together with a number of equations.

For instance, first-order logic can be usefully described as having three fundamental types, apart from Ω and 1: a type I of *individual constants*, a type V of *individual variables* and a type P of *one-place predicates*. It is useful to also have a type $T = I + V$, being the coproduct (direct sum) of I and V , which we will call the type of *terms*. We assume a countably infinite set of arrows $1 \rightarrow Var$, and countable sets of arrows $1 \rightarrow I$, and from 1 to products of P . We assume that \otimes , when applied to terms, is just a normal categorical product, so there will be formation rules to make arbitrary sequences of terms. On the other hand, $P \otimes P$ will in general not be $P \times P$, since there is no canonical way to make a $2n$ -predicate from an n -predicate. This is why we use a monoidal 2-category, rather than simply a 2-category with products.

We write X^n for the n -fold \otimes -product of X with itself. Important formation rules then go from $P^n \otimes T^n$ to Ω (n -predicate application), or from Ω or $\Omega \otimes \Omega$ to Ω (connectives), or from $\Omega \otimes V$ to Ω (quantifiers). The well-formed formulae are the arrows $1 \rightarrow \Omega$ that are freely generated this way. *Inference rules* can be defined on $\text{hom}(\Omega, \Omega)$ (or $\text{hom}(\Omega \otimes \Omega, \Omega)$ for ones like \wedge -introduction). If we

want to, it is easy to extend with inference rules on predicates, and also, if we are prepared to go to 3-categories, with structural rules on proofs as well.

Most important, for our purposes, are the interpretations of one language in another. These are given by monoidal 2-functors between languages. A *semantics of \mathcal{L} in \mathcal{L}'* is a class of monoidal 2-functors $F : \mathcal{L} \rightarrow \mathcal{L}'$. We write such a semantics as $\mathfrak{S}(\mathcal{L}, \mathcal{L}')$. Soundness will follow automatically from the requirement of 2-functoriality, while completeness, when it holds, consists in the satisfaction of certain *fullness* criteria. We let a *paradigm* be a sequence of languages $\mathcal{L}_0, \mathcal{L}_1, \dots$ together with semantics $\mathfrak{S}(\mathcal{L}_k, \mathcal{L}_{k+1})$ for all k .

An example of a paradigm with \mathcal{L}_0 as first-order logic is what we may call the *Quinean* one, where all \mathcal{L}_k are first-order languages, but \mathcal{L}_{k+1} contains more individual constants than \mathcal{L}_k . Another might be called *Tarskian*, in which the language of \mathcal{L}_k for all $k > 0$ is ZFC, plus a truth predicate and individual constants for formulae of \mathcal{L}_{k-1} . There are also many others available. A paradigm, roughly speaking, can be seen as a *way of doing logic*, just as a topos can be seen as a way of doing set theory.

In a paradigm, we can distinguish between *syntactic* and *semantic* inference rules. A syntactic inference rule of \mathcal{L}_k is, roughly, one which can be specified just from the structure of \mathcal{L}_k itself, while a semantic rule is one whose specification proceeds in terms of the structure of the image of \mathcal{L}_k in \mathcal{L}_{k+1} under the translations in $\mathfrak{S}(\mathcal{L}_k, \mathcal{L}_{k+1})$. For example, the conjunction rules are syntactic in this sense, while the quantification rules are more enlighteningly seen as semantic. In first-order logic, because of the completeness theorem, the distinction is may not seem very important, but it becomes crucial when we move to higher-order logics and other theories without complete proof systems.

3 Conclusions

The paradigm concept sketched in the last paragraph can be put to use in explaining how we can do, say, the first incompleteness proof rigorously, even if we cannot formalize it in a single language. It allows us to show how second-order ZFC can be true in a merely countably infinite universe, and how, even in a language with a finite domain, quantification is not reducible to conjunction (or disjunction). We can furthermore use it to throw light on which terms in a language \mathcal{L}_k are logical constants based on how they are transformed under its semantics, through the study of natural transformations between the functors in $\mathfrak{S}(\mathcal{L}_k, \mathcal{L}_{k+1})$.

From a nominalistic point of view, the paradigm concept might also be more palatable than the usual (set-theoretic) model concept. A category can very well consist of, say, concrete objects and physical processes. A paradigm consists of languages, and in many cases these can be taken to be finite, or merely potentially infinite. If we can do anything we can do in model theory in a paradigm instead, that could perhaps be used to show why logic, even in its semantic guise, is not dependent on the actual existence of sets.

Measuring the distance between modal formulas

Philippe Balbiani

Institut de recherche en informatique de Toulouse

Abstract

We propose to introduce a metric function into the canonical model of a normal logic by means of the concept of modal equivalence and we show how it can be used to introduce a metric function into the set of all modal formulas.

Introduction Metrics have been used in several places in the literature on the formal semantics of programming languages, for example in sequential programming [1] and in concurrent programming [2]. In trace-based models of computation, a natural definition of a distance has been considered, namely the classical Baire metric: two computations have a distance 2^{-i} when the first difference between them appears after i steps. Alternative approaches based on metric methods have also appeared from time to time in knowledge representation and reasoning, either for rationalizing voting rules [4], or for representing preferences in constraint satisfaction [5]. The common idea underlying many of these approaches consists in this: the distance between propositional formulas ϕ and ψ is the minimal cardinality of the difference sets between models of ϕ and models of ψ . Through temporal logics and description logics, modal logic is both concerned with problems in the semantics of programming languages and questions in knowledge representation. Yet, none of the above approaches seems to be fit for providing a measure of the distance between modal formulas: satisfiability of modal formulas is not invariant under operations that preserve traces of models whereas the cardinalities of the difference sets between models of modal formulas do not seem to reflect the distance between them. Seeing that modal formulas correspond to sets of maximal consistent theories, we introduce a metric function into the canonical model of a normal logic by means of the concept of modal equivalence and we use it to introduce a metric function into the set of all modal formulas. Our idea rests in this: a good metric function into the canonical model of a normal logic must reflect the fact that modal formulas are the only tools available when one needs to express the difference between maximal consistent theories.

Normal logics Let BV be a finite set of Boolean variables with typical members denoted p, q , etc. We define the set $FOR(BV)$ of all formulas by the rule $\phi ::= p \mid \perp \mid \neg\phi \mid (\phi \vee \psi) \mid \Box\phi$. The depth of ϕ , in symbols $dp(\phi)$, is inductively defined as usual. In particular: $dp(\Box\phi) = dp(\phi) + 1$. A set L of formulas is said to be a normal logic iff it satisfies the following conditions: L contains all propositional tautologies;

L contains all formulas of the form $\Box(\phi \rightarrow \psi) \rightarrow (\Box\phi \rightarrow \Box\psi)$; L is closed under the rule of modus ponens (given ϕ and $\phi \rightarrow \psi$, prove ψ); L is closed under the rule of generalization (given ϕ , prove $\Box\phi$). Let K be the least normal logic. Until the end of this paper, L will denote a consistent normal logic. A set Γ of formulas is said to be an L -theory iff it satisfies the following conditions: Γ contains L ; Γ is closed under the rule of modus ponens. The set of all L -consistent formulas will be denoted Con_L . An L -theory Γ is said to be maximal iff for all formulas ϕ , either $\phi \in \Gamma$, or $\neg\phi \in \Gamma$. Let W_L be the set of all maximal consistent L -theories. Let $[\cdot]_L: \phi \mapsto [\phi]_L$ be the mapping defined by $[\phi]_L = \{\Gamma: \Gamma \in W_L \text{ is such that } \phi \in \Gamma\}$ for every formula ϕ .

Stone's topology In his fundamental paper, "Applications of the theory of Boolean rings to general topology", Stone [6] has showed how to construct a topology in the set of all maximal ideals of a Boolean ring. Within the context of W_L , a topology τ_L of a similar nature can also be constructed, generated by a basis consisting of all sets of the form $[\phi]_L$ where $\phi \in FOR(BV)$. The following results are fundamental: (W_L, τ_L) is Hausdorff; (W_L, τ_L) is regular; (W_L, τ_L) is second-countable. Hence, (W_L, τ_L) is homeomorphic to a metric space.

Metric based on modal equivalence For all $i \in \mathbb{N}$, let $\equiv_{L,i}$ be the equivalence relation on W_L defined as follows: $\Gamma \equiv_{L,i} \Delta$ iff $\Gamma \cap \{\phi: \phi \in FOR(BV) \text{ is such that } dp(\phi) \leq i\} = \Delta \cap \{\phi: \phi \in FOR(BV) \text{ is such that } dp(\phi) \leq i\}$. About introducing a metric function into W_L by means of the concept of modal equivalence, we now define a function $\delta_L: W_L \times W_L \mapsto \mathbb{R}^+$ as follows: $\delta_L(\Gamma, \Delta) = \sup\{2^{-i}: i \in \mathbb{N} \text{ is such that } \Gamma \not\equiv_{L,i} \Delta\}$. Remark that for all $\Gamma, \Delta \in W_L$, $\delta_L(\Gamma, \Delta) < 2^{-i}$ iff $\Gamma \equiv_{L,i} \Delta$. Obviously, δ_L is an ultrametric function on W_L . Let τ_{δ_L} be the topology on W_L induced by δ_L . One can demonstrate that: for all $\phi \in FOR(BV)$, $[\phi]_L \in \tau_{\delta_L}$; for all $\Gamma \in W_L$ and for all $r \in \mathbb{R}^{+*}$, $B_{\delta_L}(\Gamma, r) \in \tau_L$. These considerations prove that the topologies τ_L and τ_{δ_L} are equal. The principal advantage in a metric such as δ_L consists in this: the distance between two given maximal consistent L -theories only depends on the depth of the simplest formulas that distinguish them.

Proposition 1 (1) (W_L, δ_L) is complete; (2) (W_L, δ_L) is totally bounded; (3) (W_L, τ_L) is compact; (4) (W_L, τ_L) is separable.

All nonempty closed sets in W_L are δ_L -bounded. That is why we can introduce, between such sets, the Hausdorff metric corresponding to δ_L .

Distance between sets Let \mathcal{C}_L be the set of all nonempty closed sets in W_L . For $A, B \in \mathcal{C}_L$, define $\vec{\delta}_L(A, B) = \sup\{\delta_L(\Gamma, B): \Gamma \in A\}$ where $\delta_L(\Gamma, B) = \inf\{\delta_L(\Gamma, \Delta): \Delta \in B\}$. About introducing a metric function into \mathcal{C}_L , we now define the Hausdorff function $\ddot{\delta}_L: \mathcal{C}_L \times \mathcal{C}_L \mapsto \mathbb{R}^+$ as follows: $\ddot{\delta}_L(A, B) = \max\{\vec{\delta}_L(A, B), \vec{\delta}_L(B, A)\}$. Remark that for all $A, B \in \mathcal{C}_L$, $\ddot{\delta}_L(A, B) < 2^{-i}$ iff for all $\Gamma \in A$, there exists $\Delta \in B$ such that $\Gamma \equiv_{L,i} \Delta$ and for all $\Delta \in B$, there exists $\Gamma \in A$ such that $\Gamma \equiv_{L,i} \Delta$. Obviously, $\ddot{\delta}_L$ is an ultrametric function on \mathcal{C}_L . Let $\tilde{\tau}_L$ be the topology on \mathcal{C}_L induced by $\ddot{\delta}_L$.

Proposition 2 (1) $(\mathcal{C}_L, \ddot{\delta}_L)$ is complete; (2) $(\mathcal{C}_L, \ddot{\delta}_L)$ is totally bounded; (3) $(\mathcal{C}_L, \ddot{\tau}_L)$ is compact; (4) $(\mathcal{C}_L, \ddot{\tau}_L)$ is separable.

Since $[\phi]_L \in \mathcal{C}_L$ for every $\phi \in Con_L$, then we can introduce, between formulas, a distance corresponding to δ_L .

Distance between formulas Let \cong_L be the equivalence relation on Con_L defined as follows: $\phi \cong_L \psi$ iff $[\phi]_L = [\psi]_L$. Let $\phi \in Con_L$. The set of all formulas in Con_L equivalent to ϕ modulo \cong_L , in symbols $\|\phi\|_L$, is called the equivalence class modulo \cong_L with ϕ as its representative. The set of all equivalence classes of Con_L modulo \cong_L , in symbols Con_L / \cong_L , is called the quotient set of Con_L modulo \cong_L . About introducing a metric function into Con_L / \cong_L , we now define the Hausdorff-like function $\hat{\delta}_L: Con_L / \cong_L \times Con_L / \cong_L \rightarrow \mathbf{R}^+$ as follows: $\hat{\delta}_L(\|\phi\|_L, \|\psi\|_L) = \ddot{\delta}_L([\phi]_L, [\psi]_L)$. Remark that for all $\phi, \psi \in Con_L$, $\hat{\delta}_L(\|\phi\|_L, \|\psi\|_L) < 2^{-i}$ iff for all $\Gamma \in [\phi]_L$, there exists $\Delta \in [\psi]_L$ such that $\Gamma \equiv_{L,i} \Delta$ and for all $\Delta \in [\psi]_L$, there exists $\Gamma \in [\phi]_L$ such that $\Gamma \equiv_{L,i} \Delta$. Obviously, $\hat{\delta}_L$ is an ultrametric function on Con_L / \cong_L . Let $\hat{\tau}_L$ be the topology on Con_L / \cong_L induced by $\hat{\delta}_L$.

Proposition 3 (1) $(Con_K / \cong_K, \hat{\delta}_K)$ is not complete; (2) $(Con_L / \cong_L, \hat{\delta}_L)$ is totally bounded; (3) $(Con_K / \cong_K, \hat{\tau}_K)$ is not compact; (4) $(Con_L / \cong_L, \hat{\tau}_L)$ is separable.

References

- [1] De Bakker, J., de Vink, E. Control Flow Semantics. MIT Press (1996).
- [2] Van Breugel, F. An introduction to metric semantics: operational and denotational models for programming and specification languages. Theoretical Computer Science **258** (2001) 1–98.
- [3] Dugundji, J. Topology. Allyn and Bacon (1966).
- [4] Elkind, E., Faliszewski, P., Slinko, A. On the role of distances in defining voting rules. In van der Hoek W., Kaminka, G., Lespérance, Y., Luck, M., Sen, S. (editors): Proceedings of 9th International Conference on Autonomous Agents and Multiagent Systems. International Foundation for Autonomous Agents and Multiagent Systems (2010) 375–382.
- [5] Hebrard, E., O’Sullivan, B., Walsh, T. Distance constraints in constraint satisfaction. In Sangal, R., Mehta, H., Bagga, R. (editors): Proceedings of the Twentieth International Joint Conference on Artificial Intelligence. Morgan Kaufmann (2007) 106–111.
- [6] Stone, M. Applications of the theory of Boolean rings to general topology. Transactions of the American Mathematical Society **41** (1937) 375–481.

The Herbrand Topos

Benno van den Berg *

June 22, 2012

The aim of my talk will be to introduce a new topos, which we will call the Herbrand topos and is inspired by earlier joint work with Eyvind Briseid and Pavol Safarik (see [1]). In [2] we hit upon a new realizability interpretation in an attempt to find computational content in arguments performed in nonstandard analysis. This new interpretation, which was a variant of modified realizability, was dubbed Herbrand realizability. Although our investigations in [2] were entirely proof-theoretic, it is also possible to explain Herbrand realizability in semantic terms.

To develop this semantics we use topos theory (for which see [9, 6, 7]). This choice was motivated by the fact that the notion of a topos is the most comprehensive notion of model for a constructive system we have available, incorporating topological, sheaf and Kripke models, as well as various realizability and functional interpretations. In addition, it shows that these interpretations can be made to work for full higher-order arithmetic. The starting point for this paper was the theory of realizability toposes (starting with [5] and surveyed in [12]): indeed, the topos most closely related to the topos we will introduce here is the modified realizability topos (for which, see [11, 12]).

In order to arrive at the modified realizability topos, one has to abstract considerably from Kreisel's original definition [8]. First of all, one fixes the hereditarily effective operations (HEO) as a model of Gödel's T . Then a type gets identified with a certain inhabited set of codes and a set of realizers of that type will simply be subset of that set. The step that Grayson took in [4] was to take as truth values any pair (A_0, A_1) where A_0 and A_1 are two sets of codes (often called the actual realizers and the potential realizers, respectively) with $A_0 \subseteq A_1$ and A_1 containing a fixed element. One can build a tripos around such pairs and in the associated topos the finite types will be interpreted as the hereditarily effective operations.

In order to define the Herbrand topos, we make a similar move. The idea from [2] was that in order to realize

$$(\exists n \in \mathbb{N}) \varphi(n)$$

*Mathematisch Instituut, Universiteit Utrecht, PO. Box 80010, 3508 TA Utrecht, the Netherlands. Email: B.vandenBerg1@uu.nl. Supported by the Netherlands Organisation for Scientific Research.

it suffices to supply a finite list of natural numbers (n_1, \dots, n_k) such that $\varphi(n_i)$ is realized for some $i \leq k$. Abstracting away from the details, this means that potential realizers are finite list of natural numbers, while the actual realizers are those finite lists (n_1, \dots, n_k) which contain an n_i which works (this is similar to the idea of Herbrand disjunctions in proof theory; hence the name). Abstracting even further, we say that truth values in the Herbrand topos are pairs of sets of codes (A_0, A_1) such that A_0 consists of finite sequences all whose elements belong to A_1 and which is closed upwards (by this we mean that it is closed under supersets, if we regard finite sequences as representatives for their set of components). We will show that on the basis of these pairs one can construct a tripos, whose associated topos we will call the Herbrand topos.

The Herbrand topos turns out to have several features in common with other realizability toposes. It has an interesting subcategory consisting of the $\neg\neg$ -separated objects (we will call these the Herbrand assemblies) and the category of sets is included as a subtopos via the $\neg\neg$ -topology. What is very unusual, however, is that this inclusion functor, which we will call ∇ , preserves and reflects the validity of first-order logic; in fact, ∇ preserves and reflects the structure of a locally cartesian closed pretopos. In particular, $\nabla 2 = 2$ in the Herbrand topos.

This is a striking illustration of the fact that in the Herbrand topos disjunction has essentially no constructive content. Indeed, in order for a disjunction $\varphi \vee \psi$ to be realized it is sufficient that one of the two disjuncts is realized; but a realizer for $\varphi \vee \psi$ need not say which disjunct it is that is actually realized. This fact explains many of the features of the Herbrand topos: why it believes in the law of excluded middle for Π_1^0 -formulas, and why it does not believe in Church's thesis or in continuity principles.

However, arithmetic in the Herbrand topos is not classical. This is due to the fact that existential quantifiers still have some constructive content. Admittedly, this content is less than is usually the case, but it is still strong enough to rule out Markov's principle.

Finally, there are two other properties of the Herbrand topos which are worth mentioning. First of all, because ∇ preserves and reflects first-order logic, ∇A will be a nonstandard model of arithmetic in the Herbrand topos for any nonstandard model A (actually, it will also be a nonstandard model when A is the standard model \mathbb{N}). This is interesting, because realizability toposes are unfavourable terrain for nonstandard models of arithmetic (see [10]).

A proof-theoretic feature of the Herbrand topos which is worth stressing is that in it the Fan Theorem holds; for this it is not necessary to assume its validity in the metatheory. The proof should look very familiar to anyone who is aware of the bounded modified realizability interpretation and its properties (for which, see [3]).

References

- [1] B. van den Berg – *The Herbrand Topos*. Preprint available as arXiv:1112.3837, 2012.
- [2] B. van den Berg, E. Briseid and P. Safarik – A functional interpretation for nonstandard arithmetic. Accepted for publication in the *Annals of Pure and Applied Logic*, 2012.
- [3] F. Ferreira and A. Nunes – Bounded modified realizability. *J. Symbolic Logic* 71 (1), 2006, pp. 329–346.
- [4] R. J. Grayson – *Modified realisability toposes*. Handwritten notes from Münster university, 1981,
- [5] J. M. E. Hyland – The effective topos. In: *The L.E.J. Brouwer Centenary Symposium (Noordwijkerhout, 1981)*. Stud. Logic Foundations Math. 110, pp. 165–216. North-Holland, Amsterdam, 1982.
- [6] P. T. Johnstone – *Sketches of an elephant: a topos theory compendium. Volume 1*. Oxf. Logic Guides 43, Oxford University Press, New York, 2002.
- [7] P. T. Johnstone – *Sketches of an elephant: a topos theory compendium. Volume 2*. Oxf. Logic Guides 44, Oxford University Press, Oxford, 2002.
- [8] G. Kreisel – Interpretation of analysis by means of constructive functionals of finite types. In: *Constructivity in mathematics: Proceedings of the colloquium held at Amsterdam, 1957 (edited by A. Heyting)*. Studies in Logic and the Foundations of Mathematics, pp. 101–128. North-Holland, Amsterdam, 1959.
- [9] S. Mac Lane and I. Moerdijk – *Sheaves in geometry and logic – A first introduction to topos theory*. Universitext, Springer-Verlag, New York, 1992.
- [10] D. C. McCarty – Variations on a thesis: intuitionism and computability. *Notre Dame J. Formal Logic* 28 (4), 1987, pp. 536–580.
- [11] J. van Oosten – The modified realizability topos. *J. Pure Appl. Algebra* 116 (1-3), 1997, pp. 273–289.
- [12] J. van Oosten – *Realizability – An Introduction to its Categorical Side*. Studies in Logic 152, Elsevier, Amsterdam, 2008.

INVARIANCE AND DEFINABILITY, WITH OR WITHOUT EQUALITY

DENIS BONNAY AND FREDRIK ENGSTRÖM

Invariance under permutation has been used as a logicality criterion in the context of the semantic definition of logical consequence. By logicality criterion, we mean a characterization of the kind of interpreted symbols that should be used as logical constants in the Tarskian definition. In this perspective, invariance under permutation is taken as the formal output of a conceptual analysis of logic. Thus, it has been proposed as a mathematical counterpart to the generality of logic (by Tarski himself, [12]) and to its purely formal nature (by Sher [11] and MacFarlane [8]). The issues we will be concerned with are the characterization of permutation invariant operations by McGee in [9] and Feferman's question in [3] about the characterization of homomorphism invariant operations. The unified perspective will be based on Krasner's much earlier work ([5], [6]) and his so-called abstract Galois theory.

Given a fixed domain, Krasner establishes a general correspondence between classes of relations (of infinite arity) closed under definability in $\mathcal{L}_{\infty\infty}$ and subgroups of the permutation group. McGee shows that the classes of relations and quantifiers which are invariant under all permutations is precisely the class of relations and quantifiers which are definable in pure $\mathcal{L}_{\infty\infty}$. It appears that McGee's theorem is nothing but a special case of Krasner's correspondence when second-order relations, which can be thought of as quantifiers, are allowed for. Feferman shows that an operation is definable in first-order logic without equality just in case it is definable in the λ -calculus from homomorphism invariant *monadic* quantifiers and asks whether "there is a natural characterization of the homomorphisms invariant propositional operations in general, in terms of logics extending the predicate calculus" ([3], p. 47). This suggests a further generalization of Krasner's correspondence to the equality-free version of $\mathcal{L}_{\infty\infty}$.

Krasner's abstract Galois theory. To begin with, we shall recall the fundamentals of Krasner's *abstract* Galois theory, following in particular Poizat [10]. Consider a domain Ω , G the full permutation group on Ω and a set \mathcal{R} of relations on Ω ; these relations may include infinite arity relations regarded as subsets of Ω^α where α is some ordinal number. We now define the following pair of mappings:

$$\begin{aligned} \text{Inv}(H) &= \{ R \subseteq \Omega^\alpha \mid hR = R \text{ for all } h \in H, \alpha \leq |\Omega| \} \\ \text{Aut}(\mathcal{R}) &= \{ g \in G \mid gR = R \text{ for all } R \in \mathcal{R} \} \end{aligned}$$

The logic $\mathcal{L}_{\infty\infty}$ is the infinitary generalization of the predicate calculus where formulas are built by means of arbitrarily long conjunctions and disjunctions and by means of arbitrarily long universal and existential quantifiers sequences (see e.g. [4]). The $\mathcal{L}_{\infty\infty}$ -closure of \mathcal{R} is the set of relations definable in $\mathcal{L}_{\infty\infty}$ from relations in \mathcal{R} . The following Theorem is essentially¹ shown by Krasner in [5]:

Theorem 1. *Let Ω and G be as above, $H \subseteq G$ any set of permutations and \mathcal{R} any set of relations on Ω of arities at most $|\Omega|$.*

- (1) *$\text{Inv}(\text{Aut}(\mathcal{R}))$ is the $\mathcal{L}_{\infty\infty}$ -closure of \mathcal{R} .*
- (2) *$\text{Aut}(\text{Inv}(H))$ is the smallest subgroup of G including H .*

Thus, there is a one-to-one correspondence between the subgroups of the full permutation group G of Ω and the sets of relations closed under definability in $\mathcal{L}_{\infty\infty}$.

Both authors were partially supported by the EUROCORE LogICCC LINT program and the Swedish Research Council.

¹The qualification 'essentially' is due to the fact that Krasner thinks directly in terms of closure under 'logical' operations on relations rather in terms of definability in a formal language.

Second-order relations, $\mathcal{L}_{\infty\infty}$ and Krasner's correspondence. We shall now extend Krasner's correspondence to second-order operations, in order to account for quantifier extensions, which have been the traditional focus of the debates regarding logicity and invariance. In this subsection, we state the corresponding generalization of Theorem 1 and prove its first part.

A finite second-order relation Q of type (i_1, \dots, i_k) on Ω is a subset of $\mathcal{P}(\Omega^{i_1}) \times \dots \times \mathcal{P}(\Omega^{i_k})$ for finite k and finite i_1, \dots, i_k . A second-order structure \mathcal{Q} on a domain Ω is a set of finite first-order and second-order relations on Ω . A permutation g on Ω preserves a second-order relation Q of type (i_1, \dots, i_k) if $(R^{i_1}, \dots, R^{i_k}) \in Q$ iff $(gR^{i_1}, \dots, gR^{i_k}) \in Q$ where R^{i_j} , $1 \leq j \leq k$, are first-order relations of arities i_j . The mappings Aut and Inv admit a straightforward generalization to the present setting: $\text{Aut}(\mathcal{Q})$ is the group of permutations which preserve all first-order and second-order relations in \mathcal{Q} , and $\text{Inv}(H)$, for $H \subseteq G$, is the set of first-order and second-order relations which are preserved by all permutations in H .

Given a second-order structure \mathcal{Q} , $\mathcal{L}_{\infty\infty}(\mathcal{Q})$ is an interpreted language in the logic $\mathcal{L}_{\infty\infty}$: it is the language whose signature matches the structure \mathcal{Q} and whose predicate and quantifier symbols are interpreted by the relations in \mathcal{Q} . The syntax and semantics for symbols interpreted by second-order relations is familiar from generalized quantifier theory (see [7]).

We shall need to consider definability in $\mathcal{L}_{\infty\infty}(\mathcal{Q})$ for a given \mathcal{Q} . Without loss of generality, let Q be a second-order relation of type (2). We say that Q is definable in $\mathcal{L}_{\infty\infty}(\mathcal{Q})$ iff there is a sentence $\phi_Q(\bar{R})$ in $\mathcal{L}_{\infty\infty}(\mathcal{Q})$ expanded with a binary predicate symbol \bar{R} such that

$$\mathcal{Q}, R \models \phi_Q(\bar{R}) \text{ iff } R \in Q$$

where R is a binary first-order relation on Ω interpreting \bar{R} . We can now state the generalization of Theorem 1 to second-order structures and automorphism groups thereof:

Theorem 2. *Let Ω be a domain and G the symmetric group on Ω , $H \subseteq G$ any set of permutations and \mathcal{Q} a second-order structure on Ω .*

- (1) $Q \in \text{Inv}(\text{Aut}(\mathcal{Q}))$ iff Q is definable in $\mathcal{L}_{\infty\infty}(\mathcal{Q})$. The same holds for relations R .
- (2) $\text{Aut}(\text{Inv}(H))$ is the smallest subgroup of G including H .

Invariance under similarities. We now turn to the case of equality free logics and invariance under similarity relations. This was first proposed by Casanovas in [2] and [1], and also by Feferman in [3]. In these works similarity invariance *between domains* is analyzed and several different equivalences are shown for different invariance criteria of this kind. In this paper we are interested in fixing one domain and obtaining a correspondence between invariance and definability, not very different from the results above.

A binary relation π is a similarity relation from A to B if for all $a \in A$ there is $b \in B$ such that $a \pi b$ and for all $b \in B$ there is $a \in A$ such that $a \pi b$. In other words $\pi \subseteq A \times B$ is a similarity relation iff $\text{dom}(\pi) = A$ and $\text{rng}(\pi) = B$. When $A = B$ we say that π is a similarity relation on A .

Given a similarity relation π from A to B and relations $R \subseteq A^k$, $S \subseteq B^k$ we define $R \pi S$ iff for every $\bar{a} \pi \bar{b}$ we have $\bar{a} \in R$ iff $\bar{b} \in S$. A relation $R \subseteq \Omega^k$ is invariant under the similarity relation π on Ω if for all $\bar{a} \pi \bar{b}$ we have $\bar{a} \in R$ iff $\bar{b} \in R$, in other words R is invariant under π iff $R \pi R$.

Given a quantifier Q on Ω we say that Q is invariant under π if for all relations $R_1, \dots, R_k, S_1, \dots, S_k$ on Ω such that $R_i \pi S_i$ we have $\langle R_1, \dots, R_k \rangle \in Q$ iff $\langle S_1, \dots, S_k \rangle \in Q$.

We define two mappings: \cdot/\sim and \cup which will operate on several different domains, but in a similar way. Let \sim be an equivalence relation on Ω and $[a] = \{b \mid a \sim b\}$ be the equivalence class of a . Given $R \subseteq \Omega^k$ we define the relation R/\sim on Ω/\sim by

$$R/\sim = \{ \langle [a_1], \dots, [a_k] \rangle \mid \langle a_1, \dots, a_k \rangle \in R \}.$$

If $S \subseteq (\Omega/\sim)^k$ then

$$\cup S = \{ \langle a_1, \dots, a_k \rangle \mid \langle [a_1], \dots, [a_k] \rangle \in S \}.$$

Given a quantifier Q on Ω , the quantifier Q/\sim on Ω/\sim is defined by

$$Q/\sim = \{ \langle R_1, \dots, R_k \rangle \mid \langle \cup R_1, \dots, \cup R_k \rangle \in Q \},$$

and given a quantifier Q on Ω/\sim , the quantifier $\cup Q$ on Ω is defined by

$$\cup Q = \{ \langle \cup R_1, \dots, \cup R_k \rangle \mid \langle R_1, \dots, R_k \rangle \in Q \}.$$

Invariance is now parametrized by the equivalence relations \sim we are considering. We say that $R \subseteq \Omega^k$ respects \sim if $\cup(R/\sim) = R$. Given a similarity π , a quantifier Q is \sim -invariant under π if for any $\bar{R}, \bar{S} \subseteq \Omega^k$ respecting \sim if $\bar{R} \pi \bar{S}$ then $\bar{R} \in Q$ iff $\bar{S} \in Q$. A relation is \sim -invariant under π if it is invariant under π .

A set of operations \mathcal{Q} generates an equivalence relation $\sim_{\mathcal{Q}}$ corresponding to definability in $\mathcal{L}_{\infty, \infty}^-(\mathcal{Q})$, that is $a \sim_{\mathcal{Q}} b$ iff

$$\bigwedge_{\phi \in \mathcal{L}_{\infty, \infty}^-(\mathcal{Q})} \forall \bar{x} (\phi(a, \bar{x}) \leftrightarrow \phi(b, \bar{x})).$$

Also, a set of similarities Π gives us an equivalence relation by the following condition:

$$a \approx_{\Pi} b \text{ iff for all } \bar{c} \in \Omega^k \text{ there exists } \exists \pi \in \Pi \text{ such that } a, \bar{c} \pi b, \bar{c}.$$

We need some definitions to state the main theorem of invariance under similarities.

Definition 3. Let $\text{Inv}(\Pi)$ be the set of all relations R and quantifiers Q on Ω which are \approx_{Π} -invariant under all similarities in Π . $\text{Sim}(\mathcal{Q})$ is the set of similarities π such that all relations and quantifiers in \mathcal{Q} are $\sim_{\mathcal{Q}}$ -invariant under π .

Definition 4. • A similarity π is *identity-like* (with respect to Π) if $\pi \subseteq \approx_{\Pi}$.
 • A set Π of similarities is *saturated* if it includes all identity-like similarities.
 • Π is a *monoid with involution* if it is closed under composition and taking converses.
 • Π is *full* if it is a saturated monoid with involution closed under taking subsimilarities, i.e., such that if $\pi \in \Pi$ and $\pi' \subseteq \pi$ is a similarity then $\pi' \in \Pi$.

We are now ready to state the main theorem, generalizing Theorem 2 to the case of an equality free setup.

Theorem 5. Let \mathcal{Q} be a set of operators and Π a set of similarities, then

- (1) $Q \in \text{Inv}(\text{Sim}(\mathcal{Q}))$ iff $\cup(Q/\sim_{\mathcal{Q}})$ is definable in $\mathcal{L}_{\infty, \infty}^-(\mathcal{Q})$.
- (2) $R \in \text{Inv}(\text{Sim}(\mathcal{Q}))$ iff R is definable in $\mathcal{L}_{\infty, \infty}^-(\mathcal{Q})$.
- (3) $\text{Sim}(\text{Inv}(\Pi))$ is the smallest full monoid including Π .

REFERENCES

- [1] E. Casanovas. Logical operations and invariance. *Journal of Philosophical Logic*, 36(1):33–60, 2007.
- [2] E. Casanovas, P. Dellunde, and R. Jansana. On elementary equivalence for equality-free logic. *Notre Dame Journal of Formal Logic*, 37(3):506–522, 1996.
- [3] Solomon Feferman. Logic, logics, and logicism. *Notre Dame Journal of Formal Logic*, 40(1):31–54, 1999.
- [4] Carol Karp. *Languages with Expressions of Infinite Length*. North-Holland, Amsterdam, 1964.
- [5] Marc Krasner. Une généralisation de la notion de corps. *Journal de mathématiques pures et appliquées*, 17(3-4):367–385, 1938.
- [6] Marc Krasner. Généralisation abstraite de la théorie de galois. In *Colloque d'algèbre et de théorie des nombres*, pages 163–168. Éditions du Centre National de la Recherche Scientifique, 1950.
- [7] Per Lindström. First-order predicate logic with generalized quantifiers. *Theoria*, 32:165–171, 1966.
- [8] John MacFarlane. *What Does it Mean to Say that Logic is Formal*. PhD thesis, University of Pittsburgh, 2000.
- [9] Vann McGee. Logical operations. *Journal of Philosophical Logic*, 25:567–580, 1996.
- [10] Bruno Poizat. La théorie de galois abstraite. *Eleftheria*, 3:99–107, 1985.
- [11] Gila Sher. *The Bounds of Logic*. MIT Press, Cambridge, Mass., 1991.
- [12] Alfred Tarski. What are logical notions? *History and Philosophy of Logic*, 7:143–154, 1986.

DÉPARTEMENT DE PHILOSOPHIE, UNIVERSITÉ PARIS OUEST NANTERRE, 200, AVENUE DE LA RÉPUBLIQUE, 92001 NANTERRE CEDEX, FRANCE

E-mail address: denis.bonnay@u-paris10.fr

DEPARTMENT OF PHILOSOPHY, LINGUISTICS AND THEORY OF SCIENCE, UNIVERSITY OF GOTHENBURG, BOX 200, 405 30 GOTHENBURG, SWEDEN

E-mail address: fredrik.engstrom@gu.se

On a functional interpretation for nonstandard arithmetic

Eyvind Briseid*

(joint work with Benno van den Berg[†] and Pavol Safarik[‡][2])

June 1, 2012

We will present work dealing with proof-theoretic investigations of systems for non-standard analysis. This involves introducing constructive and classical systems for non-standard arithmetic and showing how variants of the functional interpretations due to Gödel and Shoenfield can be used to rewrite proofs performed in these systems into standard ones. There are two principal outcomes of this:

- Similarly to the case for the ordinary functional interpretation the rewriting algorithm allows term extraction for suitable classes of statements.
- In particular, the functional interpretations show that our nonstandard systems are conservative extensions of $\mathbf{E-HA}^\omega$ and $\mathbf{E-PA}^\omega$, strengthening earlier results by Moerdijk and Palmgren [7] and Avigad and Helzner [1].

A main source of inspiration for the present work comes from Nelson’s Internal Set Theory (IST), introduced in [8]¹. Nelson’s idea was to treat nonstandard analysis syntactically by adding a new unary predicate symbol st to \mathbf{ZFC} for “being standard”, and further adding three new axioms which govern the use of this new unary predicate symbol: Idealization, Standardization, and Transfer. The resulting system he called Internal Set Theory or IST. The main logical result about IST is that it is a conservative extension of \mathbf{ZFC} , so any theorem provable in IST which does not involve the st -predicate is provable in \mathbf{ZFC} as well.

The conservativity of IST over \mathbf{ZFC} was proved twice. In [8], Nelson gives a model-theoretic argument (which he attributes to Powell). Later [9] he proves the same result syntactically by providing a “reduction algorithm” (a rewriting algorithm) for converting proofs performed in IST to ordinary \mathbf{ZFC} -proofs. There is a remarkable similarity between his reduction algorithm and the Shoenfield interpretation [10]; this observation was the starting point for our work.

Rather than set theory we work with systems in higher types, such as extensional Heyting and Peano arithmetic in all finite types ($\mathbf{E-HA}^\omega$ and $\mathbf{E-PA}^\omega$), because in addition to conservation results we are interested in extracting terms from nonstandard proofs and “proof mining”. Proof mining is an area of applied logic where one uses proof-theoretic techniques to extract quantitative information (such as bounds on the growth rate of certain functions) from proofs in ordinary mathematics. In addition, such techniques

*Department of Mathematics, The University of Oslo, Postboks 1053, Blindern, 0316 Oslo. Email addresses: eyvindmb@math.uio.no, eyvindbriseid@gmail.com. Supported by the Research Council of Norway (Project 204762/V30).

[†]Mathematisch Instituut, Universiteit Utrecht, PO. Box 80010, 3508 TA Utrecht. Email address: B.vandenBerg1@uu.nl. Supported by Netherlands Organisation for Scientific Research (NWO project “The Model Theory of Constructive Proofs”).

[‡]Fachbereich Mathematik, Technische Universität Darmstadt, Schloßgartenstraße 7, 64289 Darmstadt. Email address: pavol.safarik@googlemail.com. Supported by the German Science Foundation (DFG project KO 1737/5-1).

¹This paper was reprinted with a foreword by G. F. Lawler in volume 48, no. 4, of the *Bulletin of the American Mathematical Society* in recognition of its status as a classic.

can reveal certain uniformities leading to new qualitative results as well. Functional interpretations are one of the main tools in proof mining (for an introduction to this part of applied proof theory, see [6]). To extract interesting bounds, however, it is important that the mathematical arguments one analyses can be performed in sufficiently weak systems: therefore one considers systems such as $\mathbf{E-HA}^\omega$ or $\mathbf{E-PA}^\omega$, or fragments thereof, rather than \mathbf{ZFC} . The reason for considering systems in higher types (rather than \mathbf{PA} , for instance) is not just because they are more expressive, but also because higher types are precisely what makes functional interpretations work.

We take as starting point $\mathbf{E-HA}^\omega$ and proceed in a similar way as Nelson: we add a new unary predicate st to its language (in fact, we will add unary predicates st^σ for every type σ) and add nonstandard axioms in the extended language. Our main result is the existence of an algorithm which rewrites proofs in this constructive nonstandard system to ordinary proofs performed in $\mathbf{E-HA}^\omega$. This algorithm is a functional interpretation in the style of Gödel, with features reminiscent of the Diller-Nahm [3] and the bounded functional interpretation [4] (the relation to the latter is especially close). Then by combining this rewriting algorithm with negative translation one obtains a Shoenfield-type functional interpretation for a nonstandard extension of $\mathbf{E-PA}^\omega$.

As mentioned above the existence of such a rewriting algorithm has two corollaries: first of all, it shows that the nonstandard systems we consider are conservative extensions of $\mathbf{E-HA}^\omega$ and $\mathbf{E-PA}^\omega$, respectively. Secondly, they show how one can extract terms in Gödel's \mathcal{T} (and hence computational content) from nonstandard proofs.

Ongoing work involves investigating proof-theoretically the role of saturation principles in nonstandard proofs. We have shown that constructively the countable saturation principle

$$\text{CSAT} : \quad \forall^{\text{st}} n^0 \exists y^\tau \Phi(n, y) \rightarrow \exists f^{0 \rightarrow \tau} \forall^{\text{st}} n^0 \Phi(n, f(n))$$

does not add any strength, whereas classically it makes the system much stronger. Presently, we are working on determining the exact strength over our classical nonstandard system.

References

- [1] J. Avigad and J. Helzner. Transfer principles in nonstandard intuitionistic arithmetic. *Arch. Math. Logic*, 41(6):581–602, 2002.
- [2] B. van den Berg, E. Briseid, and P. Safarik. A functional interpretation for nonstandard arithmetic. Preprint submitted for publication.
- [3] J. Diller and W. Nahm. Eine Variante zur Dialectica-Interpretation der Heyting-Arithmetik endlicher Typen. *Arch. Math. Logik Grundlagenforsch.*, 16:49–66, 1974.
- [4] F. Ferreira and P. Oliva. Bounded functional interpretation. *Ann. Pure Appl. Logic*, 135(1-3):73–112, 2005.
- [5] U. Kohlenbach. *Applied proof theory: proof interpretations and their use in mathematics*. Springer Monographs in Mathematics. Springer-Verlag, Berlin, 2008.
- [6] I. Moerdijk and E. Palmgren. Minimal models of Heyting arithmetic. *J. Symbolic Logic*, 62(4):1448–1460, 1997.
- [7] E. Nelson. Internal set theory: a new approach to nonstandard analysis. *Bull. Amer. Math. Soc.*, 83(6):1165–1198, 1977.
- [8] E. Nelson. The syntax of nonstandard analysis. *Ann. Pure Appl. Logic*, 38(2):123–134, 1988.
- [9] J.R. Shoenfield. *Mathematical logic*. Association for Symbolic Logic, Urbana, IL, 2001. Reprint of the 1973 second printing.

An exposition of abductive reasoning through logic programming with constraints

— *extended abstract* —

Henning Christiansen

Research group PLIS: Programming, Logic and Intelligent Systems
Department of Communication, Business and Information Technologies
Roskilde University, P.O.Box 260, DK-4000 Roskilde, Denmark
E-mail: henning@ruc.dk

1 Introduction

We demonstrate how a fairly powerful version of abductive reasoning can be expressed directly in the logic programming language Prolog, using its extension by Constraint Handling Rules as the engine to take care of abducible hypotheses.

Until the shift of the millennium, abduction in logic programming was realized through complex meta-interpreters written in Prolog, which may have led to a view of abduction as being something hairy, difficult stuff, far too inefficient for any realistic applications. There is, however, a relationship between the paradigm of constraint logic programming, which appeared in the late 1980es [6], and abduction which have not been fully recognized. The point is that constraint predicates behave in very much the same way as abducible predicates for abduction. Now, the extension of Prolog with Constraint Handling Rules in the 1990es [4] allows the programmer to introduce his or her own constraint solvers in a declarative fashion. Putting these two observation together opens up for a general approach to abduction in logic programming that we unfold in the following. The principle have been developed by the author over last decade or so, together with different coauthors, most importantly [1] and [3].

2 Prolog versus abductive logic programming

Prolog is a language capable of answering certain queries in a deductive fashion based on a closed worlds semantics: everything that is explicitly stated to be true is true, and so is everything that can be derived deductively from that; everything else is false. Consider the following program.

```
p:- a.  
p:- b.  
a.
```

Anticipating abduction, we may talk about *observations* rather than queries. Assume we have observed that **p** is the case, and we present this as a query **?-p** to the program above; Prolog gives the successful answer **yes** since the fact **a** is part of the program. Removing this fact and trying again, i.e., presenting the observation **p** to the program **{p:- a, p:- b}**, Prolog returns the laconic answer **no**, meaning that the observation **p** could not sensibly have been made from what is known.

A sort of abductive reasoning may arise when we go beyond Prolog and change the semantics for the predicates **a** and **b** into an open world view. This means to interpret their absence as facts in a program as “perhaps true” rather than “not true”. Then we might expect answers “yes, **p** is true, provided **a** is actually true” or, alternatively, the same for **b**.

An abductive logic program *P* usually depends on some designated abducible predicates *A*, such as **a** and **b** in the example above, plus so-called integrity constraints *IC* that most hold for the collection of abducible facts produced. An example of an integrity constraint may be “**a** and **c** cannot be the case at the same time”, under which the query **?-c, p** would yield only one explanation, namely **b**, and not **a**.

We will discuss integrity constraints in more detail and how to define them later; here we give a standard definition [7] of an abductive answer (or explanation) E to an observation Obs .

$$P \cup E \models Obs \quad (1)$$

$$P \cup E \cup IC \text{ is consistent} \quad (2)$$

3 Constraint logic programming as abduction

The constraints of a constraint logic program behave very much the same way as abducibles, in the sense that an answer may consist of a collection of special atoms encountered during the execution. Consider for example this program, in which the constraint predicate $\#>$ represents a greater-than relation between rational numbers.

```
p(X):- X #> 5, 10 #> X.
```

An answer to the query $p(N)$ may consist of the set of constraints $E = \{N \#> 5, 10 \#> N\}$, as a shorthand for an infinite set of ground answers of the form $N = \dots$.

We can consider such a constraint logic program as an abductive logic program, with constraint predicates viewed as abducible and taking the (knowledge embedded in the) constraint solver as integrity constraints. Without going into the formal details, we can argue that E (above) is an abductive answer – or explanation why $?-p(X)$ can hold – as follows.

- $p(X)$ would succeed in Prolog if the program is extended with two facts `const #> 5` and `10 #> const`, where `const` is an arbitrary constant symbol,
- $\exists N(N\#>5 \wedge 10\#>N)$ is consistent with a reasonable theory about $\#>$.

We shall avoid a comparison of the standard way of defining the semantics for constraint logic programming and the one that we have hinted with this discussion, and head to our main point, namely how an abductive logic program can be written using Prolog plus Constraint Handling Rules.

4 Constraint Handling Rules and abducible predicates

The language of Constraint Handling Rules (CHR) was introduced as an extension to Prolog in order to provide a white box approach to constraint solving, so that constraint solvers can be written in a declarative way. It is, for example, a standard exercise in CHR to write a constraint solver for the $\#>$ constraint considered above.

During execution of a program, the rules of CHR serve as rewrite rules over constraint stores, such that whenever a constraint is encountered, it is added to the constraint store, and these rules will incrementally check consistency, and perhaps simplify, the contents of the constraint store. Consider as examples the following two rules.¹

```
c1, c2 <=> c3, c4.
c5, c6 ==> c7.
```

The lefthand sides must consist of constraints (possibly parameterized by variables) and the righthand sides can be any Prolog executable term, including `fail`, but here we illustrate only constraints. The first rule, a so-called simplification rule, will replace the common occurrence of constraints `c1`, `c2` by `c3`, `c4`. The second, a propagation rule, adds new constraints, here `c7`, without removing those matched by the lefthand side. In the declarative semantics of CHR, these rules are considered as equivalence, resp. implication, as indicated by the chosen arrow symbol.

The answer to a successful query consists (in addition to possible variable substitutions) of the resulting contents of the constraints store. This is compatible with an open world understanding of constraint predicates: unless explicitly stated to be false, any collection of constraints can be assumed to be true. Here is a version of the most elaborate version of the program discussed in section 2 written in Prolog extended with CHR.

```
:- chr_constraint a,b,c.

a,c ==> fail.

p:- a.
p:- b.
```

¹ CHR rules are much more than illustrated here. Most importantly, rules can have guards.

The first line declares our abducibles as constraint predicates, the second is a CHR rule that defines the integrity constraint, and the remaining part consists of standard Prolog rules. Given the query $?-c, p$, exactly one answer is provided, namely the constraint b , that corresponds to the abductive explanation why the observation p can be true.

We refer to [2] for a precise formulation of the relationship between abductive logic programs and Prolog+CHR, including a correctness theorem. In plain word, the contribution is that we can write an abductive logic program with integrity constraints directly in Prolog+CHR in a most natural way without assuming any additional implementation code beneath the carpet. We may summarize the approach in the following translation of linguistic terms from the one domain to the other; no program transformation is needed.

Abductive logic programming	Constraint logic programming with CHR
Abductive logic programs	Prolog programs with a little bit of CHR
Abducible predicate	Constraint predicate
Integrity constraints	CHR Rules
Program rules	Program rules
Explanation	Final constraint store

It is, in fact, tempting and quite feasible, to have students to write programs in Prolog+CHR to perform task such as diagnosis and planning (as demonstrated in [2]), and then afterwards telling them that they have committed something as terrible as abduction. Among the additional advantages of the approach, we may list:

- due to the lack of any interpretational overhead, it appears to be the most efficient, known implementation of abductive logic programming,
- there is a full integration with any additional, predefined constraint solver that might be available,
- all facilities of the underlying Prolog system (including illogical ones) are available,
- it works smoothly together with Prolog’s definite clause grammars and provides, thus, an implementation of the principle of interpretation of abduction [5].

Everything comes with a price, and the price here is the lack of negation. We may partly simulate negation by additional constraints standing for negated ones and use integrity constraints such as $a(X), \text{not_}a(X) \Rightarrow \text{fail}$. Prolog’s negation as failure is of course available, but with the usual limitation that it serves as a test only, and no information can be exported from a negated call (specifically no negated constraints).

5 Further results

In other publications, not referred here due to lack of space, we have studied probabilistic abduction based on similar principles. The grammatical version has been applied in discourse analysis, among other things, for converting use case texts into UML diagrams. The approach has been applied successfully in teaching for different audiences, from computer science to linguist students.

References

1. S. Abdennadher and H. Christiansen. An experimental CLP platform for integrity constraints and abduction. In *Proceedings of FQAS2000, Flexible Query Answering Systems: Advances in Soft Computing series*, pages 141–152. Physica-Verlag (Springer), 2000.
2. H. Christiansen. Executable specifications for hypothesis-based reasoning with Prolog and Constraint Handling Rules. *J. Applied Logic*, 7(3):341–362, 2009.
3. H. Christiansen and V. Dahl. HYPROLOG: A new logic programming language with assumptions and abduction. In M. Gabbrielli and G. Gupta, editors, *ICLP*, volume 3668 of *Lecture Notes in Computer Science*, pages 159–173. Springer, 2005.
4. T. Frühwirth. *Constraint Handling Rules*. Cambridge University Press, Aug. 2009.
5. J. R. Hobbs, M. E. Stickel, D. E. Appelt, and P. A. Martin. Interpretation as abduction. *Artificial Intelligence*, 63(1-2):69–142, 1993.
6. J. Jaffar and J.-L. Lassez. Constraint logic programming. In *POPL, Conference Record of the Fourteenth Annual ACM Symposium on Principles of Programming Languages, Munich, Germany, January 21-23, 1987*, pages 111–119, 1987.
7. A. Kakas, R. Kowalski, and F. Toni. The role of abduction in logic programming. *Handbook of Logic in Artificial Intelligence and Logic Programming*, vol. 5, Gabbay, D.M, Hogger, C.J., Robinson, J.A., (eds.), Oxford University Press, pages 235–324, 1998.

Quantified Linear Programs and Quantified Linear Implications

Pavlos Eirinakis* Salvatore Ruggieri† K. Subramani* Piotr Wojciechowski*

Abstract

We provide a brief presentation of the recent developments in the areas of Quantified Linear Programming and Quantified Linear Implication.

I. INTRODUCTION

Quantified linear programming [1], [2] is the problem of checking whether a set of linear inequalities over the reals, i.e., a linear system, is satisfiable with respect to a given quantifier string. In quantified linear programming all variables are either existentially or universally quantified. Hence, it represents a generalization of linear programming, where all variables are existentially quantified. However, the alternation of quantifiers in the quantifier string makes deciding a *Quantified Linear Program* (QLP) a much more elaborate problem. QLPs represent a rich language that is ideal for expressing schedulability specifications in real-time scheduling [3]–[6].

By extending the quantification of variables to *implications* of two linear systems, we explore *Quantified Linear Implications* (QLIs) [7]. That is, QLIs correspond to inclusion queries of polyhedral solution sets of two linear systems with respect to a given quantifier string. We mention two application areas: Consider a scenario of real-time scheduling, where the dispatcher has already obtained a schedule (solution) but then some constraints are slightly altered. QLIs can be utilized to decide whether the dispatcher needs to recompute a solution or can still use the current one. Moreover, QLIs can be used to model reactive systems [8]–[10], where the values of the universally quantified variables represent the environmental input, while the values of the existentially quantified variables represent the system’s response.

In this paper, we provide a brief presentation of the recent developments in the areas of QLP and QLI. We discuss the computational complexities of various classes of these problems, while we also examine the relation between QLPs and QLIs using a 2-person game perspective.

II. QUANTIFIED LINEAR PROGRAMMING

Quantified linear programming extends linear programming by admitting arbitrary quantifications. In a QLP, variables of a linear system are either existentially or universally (with bounds) quantified:

$$\exists \mathbf{x}_1 \forall \mathbf{y}_1 \in [\mathbf{l}_1, \mathbf{u}_1] \dots \exists \mathbf{x}_n \forall \mathbf{y}_n \in [\mathbf{l}_n, \mathbf{u}_n] \quad \mathbf{A} \cdot \mathbf{x} + \mathbf{N} \cdot \mathbf{y} \leq \mathbf{b} \quad (1)$$

where $\mathbf{x}_1 \dots \mathbf{x}_n$ is a partition of \mathbf{x} with, possibly, \mathbf{x}_1 empty; $\mathbf{y}_1 \dots \mathbf{y}_n$ is a partition of \mathbf{y} with, possibly, \mathbf{y}_n empty; and $\mathbf{l}_i, \mathbf{u}_i$ are lower and upper bounds in \Re for \mathbf{y}_i , $i = 1, \dots, n$.

The Fourier-Motzkin existential quantifier elimination method and a universal quantifier elimination method have been employed to provide a method for deciding QLPs [2].

Theorem 2.1: The decision problem for a QLP of the form (1) is in **PSPACE**.

The special case of **E**-QLP problems, which are of the form $\exists \mathbf{y} \forall \mathbf{x} \in [\mathbf{l}, \mathbf{u}] \quad \mathbf{A} \mathbf{x} + \mathbf{N} \cdot \mathbf{y} \leq \mathbf{b}$, are solvable in polynomial time [2].

* LDCSEE, West Virginia University, Morgantown, WV. Email: {pavlos.eirinakis@mail,ksmani@csee,pwojciec@mix}.wvu.edu
This research was supported in part by the National Science Foundation through Award CCF-0827397 and Award CNS-0849735.

† Dipartimento di Informatica, Università di Pisa, Pisa, Italy. Email: ruggieri@di.unipi.it

Theorem 2.2: The decision problem for an **E-QLP** is in **P**.

Another special case that was characterized in [2] is the **F-QLP** problem, which corresponds to formulas of the form $\forall y \in [l, u] \exists x \ A \cdot x + N \cdot y \leq b$.

Theorem 2.3: The decision problem for an **F-QLP** is **coNP-complete**.

III. QUANTIFIED LINEAR IMPLICATION

Consider two linear systems $P_1 : A \cdot x \leq b$ and $P_2 : C \cdot x \leq d$. We say that P_1 is *included* in P_2 if every solution of P_1 is also a solution of P_2 . This holds if and only if the logic formula $\forall x [Ax \leq b \rightarrow C \cdot x \leq d]$ is true in the domain of the reals. Quantified Linear Implication extends the notion of inclusion to arbitrary quantifiers:

$$\exists x_1 \forall y_1 \dots \exists x_n \forall y_n [A \cdot x + N \cdot y \leq b \rightarrow C \cdot x + M \cdot y \leq d] \quad (2)$$

where $x_1 \dots x_n$ and $y_1 \dots y_n$ are partitions of x and y respectively, and where x_1 and/or y_n may be empty. We say that a **QLI** *holds* if it is true as a first-order formula over the domain of the reals. The *decision problem for a QLI* consists of checking whether it holds or not. The following result can be obtained through a reduction from the generic Q3SAT problem.

Theorem 3.1: The decision problem for a QLI of the form (2) is **PSPACE-hard**.

Let $Q(x, y)$ denote the quantifier string, namely $\exists x_1 \forall y_1 \dots \exists x_n \forall y_n$ in the QLI (2). A nomenclature is introduced in [7] to represent the classes of QLIs. Consider a triple $\langle A, Q, R \rangle$. Let A denote the number of quantifier alternations in the quantifier string $Q(x, y)$ and Q the first quantifier of $Q(x, y)$. Also, let R be an $(A + 1)$ -character string, specifying for each quantified set of variables in $Q(x, y)$ whether they appear on the **Left**, on the **Right**, or on **Both** sides of the implication. For instance, $\langle 1, \exists, \mathbf{LB} \rangle$ indicates a problem described by: $\exists x \forall y [A \cdot x + N \cdot y \leq b \rightarrow M \cdot y \leq d]$.

The following theorem presents the case of k alternations of quantifiers, with k odd. This result can be obtained through a reduction from the corresponding Q3SAT problem. Note that we write \mathbf{B}^{k+1} to denote the string $\underbrace{\mathbf{B} \dots \mathbf{B}}_{k+1}$.

Theorem 3.2: Problem $\langle k, \exists, \mathbf{B}^{k+1} \rangle$ with k odd is Σ_k^P -**hard**.

Similarly, the following can be obtained for k being even.

Theorem 3.3: Problem $\langle k, \forall, \mathbf{B}^{k+1} \rangle$ with k even is Π_k^P -**hard**.

Various classes of 0, 1, and 2-quantifier alternations have been examined in [7]. Here, we present QLIs with no quantifier alternations. The next result was obtained in [11] by reducing the problem to a finite number of linear programs, which are in **P** by [12].

Theorem 3.4: Problem $\langle 0, \forall, \mathbf{B} \rangle$ is in **P**.

The case of 0-quantifier alternation QLI starting with \exists follows trivially from Theorem 3.2 for $k = 0$.

Corollary 3.1: Problem $\langle 0, \exists, \mathbf{B} \rangle$ is in **P**.

IV. RELATION BETWEEN QLPS AND QLIS

A 2-person game semantics of QLP problems is presented in [2]. Such a game includes an existential player **X**, who chooses values for the existentially quantified variables, and a universal player **Y**, who chooses values for the universally quantified variables. **X** and **Y** make their choices according to the order of the variables in the quantifier string. If, at the end, the instantiated linear system in the QLP is true, then **X** wins the game (we say that **X** has a winning strategy). Otherwise, **Y** wins the game.

A 2-person game semantics can be given for QLIs as well. It also includes an existential player \mathbf{X} and a universal player \mathbf{Y} , who choose their moves according to the order of their corresponding variables in the quantifier string. In any game of this form, the goals of the players are the following: \mathbf{X} selects the values of the existentially quantified variables so as to violate the constraints in the Left-Hand Side (LHS) or to satisfy the constraints in the Right-Hand Side (RHS) of the implication. On the other hand, \mathbf{Y} selects the values of the universally quantified variables so as to satisfy the constraints of the LHS and on the same time to violate the constraints of the RHS of the implication. We say that \mathbf{X} *wins the game* if at the end of the game the LHS of the instantiated QLI is false or its RHS is true. Otherwise, we say that \mathbf{Y} wins the game (i.e., if the LHS is satisfied and the RHS is falsified). We say that \mathbf{X} *has a winning strategy* if it is possible for \mathbf{X} to win, i.e., if there is a sequence of moves such that \mathbf{X} wins the game. Otherwise, \mathbf{Y} has a winning strategy. The QLI holds precisely when \mathbf{X} has a winning strategy.

It is important to note that both games as described above are non-deterministic in nature, in that we have not specified *how* \mathbf{X} and \mathbf{Y} make their moves. It can be shown that the game semantics of QLIs are a conservative extension of the game semantics of QLPs. Consider a generic QLP as described by System (1). Now consider the following QLI:

$$\left. \begin{array}{l} \exists \mathbf{x}_1 \forall \mathbf{y}_1 \dots \exists \mathbf{x}_n \forall \mathbf{y}_n \\ \mathbf{l}_1 \leq \mathbf{y}_1 \leq \mathbf{u}_1 \\ \dots \\ \mathbf{l}_n \leq \mathbf{y}_n \leq \mathbf{u}_n \end{array} \right\} \rightarrow \mathbf{A} \cdot \mathbf{x} + \mathbf{N} \cdot \mathbf{y} \leq \mathbf{b} \quad (3)$$

where $\mathbf{l}_1, \dots, \mathbf{l}_n$ and $\mathbf{u}_1, \dots, \mathbf{u}_n$ are partitions of \mathbf{l} and \mathbf{u} and correspond to the lower and upper bounds respectively on the variables in $\mathbf{y}_1, \dots, \mathbf{y}_n$ of \mathbf{y} that appear in the quantifier string of System (1). The following theorem presents the relation between the two problems.

Theorem 4.1: The existential player has a winning strategy in System (3) if and only if the existential player has a winning strategy in System (1).

REFERENCES

- [1] K. Subramani, "An analysis of quantified linear programs," in *Proceedings of the 4th International Conference on Discrete Mathematics and Theoretical Computer Science (DMTCS)*, ser. Lecture Notes in Computer Science, C.S. Calude, et. al., Ed., vol. 2731. Springer-Verlag, July 2003, pp. 265–277.
- [2] —, "On a decision procedure for quantified linear programs," *Annals of Mathematics and Artificial Intelligence*, vol. 51, no. 1, pp. 55–77, 2007.
- [3] R. Gerber, W. Pugh, and M. Saksena, "Parametric dispatching of hard real-time tasks," *IEEE Transactions on Computing*, vol. 44, no. 3, pp. 471–479, 1995.
- [4] S. Choi and A. Agrawala, "Dynamic dispatching of cyclic real-time tasks with relative timing constraints," *Real-Time Systems*, vol. 19, no. 1, pp. 5–40, 2000.
- [5] K. Subramani, "An analysis of partially clairvoyant scheduling," *Journal of Mathematical Modelling and Algorithms*, vol. 2, no. 2, pp. 97–119, 2003.
- [6] —, "An analysis of totally clairvoyant scheduling," *Journal of Scheduling*, vol. 8, no. 2, pp. 113–133, 2005.
- [7] P. Eirinakis, S. Ruggieri, K. Subramani, and P. Wojciechowski, "Computational complexities of inclusion queries over polyhedral sets," in *International Symposium on Artificial Intelligence and Mathematics*, Fort Lauderdale, FL, 2012.
- [8] T. Koo, B. Sinopoli, A. Sangiovanni-Vincentelli, and S. Sastry, "A formal approach to reactive system design: unmanned aerial vehicle flight management system design example," in *Proceedings of the 1999 IEEE International Symposium on Computer Aided Control System Design*, 1999, pp. 522 – 527.
- [9] B. Pfitzmann and M. Waidner, "A model for asynchronous reactive systems and its application to secure message transmission," in *Proceedings of the 2001 IEEE Symposium on Security and Privacy*, 2001, pp. 184 – 200.
- [10] R. Hall, "Specification, validation, and synthesis of email agent controllers: A case study in function rich reactive system design," *Automated Software Engineering*, vol. 9, no. 3, pp. 233–261, 2002.
- [11] K. Subramani, "On the complexity of selected satisfiability and equivalence queries over boolean formulas and inclusion queries over hulls," *Journal of Applied Mathematics and Decision Sciences (JAMDS)*, vol. 2009, pp. 1–18, 2009.
- [12] L. G. Khachiyan, "A polynomial algorithm in linear programming," *Doklady Akademii Nauk SSSR*, vol. 224, pp. 1093–1096, 1979, english Translation: *Soviet Mathematics Doklady*, Volume 20, pp. 1093–1096.

A Lindström theorem for some normal modal logics

Sebastian Enqvist
Department of Philosophy
Lund University, Sweden
Sebastian.Enqvist@fil.lu.se

June 18, 2012

In this talk, I will address some problems in the abstract model theory for modal languages, especially concerning *characterizations* of modal languages. Here, the sort “characterization” referred to is of the same type as Per Lindström’s famous characterization of FOL, in terms of compactness and the Löwenheim-Skolem property [4]. Several Lindström-style characterizations of modal logics are already known. One of the most prominent results is the following, which is due to Johan van Benthem [6], and improves on an earlier characterization due to Maarten de Rijke [2]:

Theorem 1 (van Benthem, 2007). *Suppose an abstract modal logic \mathcal{L} is compact, invariant for bisimulation, and has the relativization property. Then \mathcal{L} is equivalent to basic modal logic.*

Here, an abstract modal logic is an extension of the basic modal logic \mathcal{ML} (for a given similarity type), satisfying certain constraints similar to the constraints on “abstract logics” as they appear in abstract model theory [1]. “Equivalent to \mathcal{ML} ” means that, for any \mathcal{L} -formula ϕ there is a corresponding \mathcal{ML} -formula that defines the same class of pointed models as ϕ . The relativization property alluded to in the theorem is an adapted version of the relativization property familiar from abstract model theory. It states that, for any ϕ and propositional variable p , there should be a formula $Rel(\phi, p)$ such that $(M, u) \models Rel(\phi, p)$ if and only if $(M \upharpoonright p, u) \models \phi$, where $M \upharpoonright p$ is the submodel of M generated by the set of nodes in M at which p is true.

Several variations of this result have been proved. Alexander Kurz and Yde Venema have obtained Lindström theorems in the general context of *coalgebraic modal logic* [3]. Similar results have been proved for modal logic with a global modality and the guarded fragment by Martin Otto and Robert Piro [5]. A Lindström theorem for the *binary guarded fragment* had been obtained earlier by Balder ten Cate, Johan van Benthem and Joukko Väänänen [7].

An area that still appears to be unexplored is to what extent these results remain valid for *restricted classes of models*. In actual applications of modal logics, it is uncommon to work with the basic modal logic of *all* Kripke models of a given similarity type. Usually, one must impose certain restrictions on the models to make sure that they match the sort of structures one wishes to speak about. Common constraints include reflexivity, transitivity, linearity etc. Also, in many cases, constraints are imposed to make sure that various modalities interact properly: an important example is propositional dynamic logic, where the accessibility relations for various programs are constrained so that the program constructors get their intended interpretation. For example, the accessibility relation corresponding to π^* should equal the transitive closure of that corresponding to π , the accessibility relation for $\pi_1; \pi_2$ should be the composition for those of π_1, π_2 respectively, and so forth.

To address this issue, we impose some definitions. Consider an abstract modal logic \mathcal{L} and a class C of pointed models of the appropriate similarity type (for simplicity I consider only similarity types with unary modalities). We say that \mathcal{L} is bisimulation invariant over C if any two bisimilar models in C satisfy the same formulas of \mathcal{L} . If an abstract modal logic is bisimulation invariant then of course it is bisimulation invariant over a given class C , but the converse need not be true. Similarly, we say that \mathcal{L} is compact over C if for any set of \mathcal{L} -formulas Γ , if every finite subset of Γ has a model in C then Γ has a model in C . Finally, the relativization property in C says that, for any formula ϕ and propositional variable p , there is a formula $Rel(\phi, p)$ such that for any model $(M, u) \in C$, $(M, u) \models Rel(\phi, p)$ iff $(M \upharpoonright p, u) \models \phi$. (For this last property to make sense, we must make sure to consider classes of models that are closed under the submodel relation.) Finally, say that \mathcal{L} is *equivalent to basic modal logic \mathcal{ML} over C* , written $\mathcal{L} \equiv_C \mathcal{ML}$, if for any \mathcal{L} formula ϕ , there is a basic modal formula ψ such that $(M, u) \models \phi$ iff $(M, u) \models \psi$ whenever $(M, u) \in C$.

For some classes of models C , \mathcal{ML} will not be compact over C and therefore we cannot hope to characterize \mathcal{ML} with respect to C in terms of the properties used in van Benthem’s characterization result. While

this poses an interesting problem, I will rather focus on those cases where \mathcal{ML} is compact over C , so that it becomes meaningful to ask whether a van Benthem-style characterization result can be obtained with respect to C . As a first observation, I show that, by slight modifications of the unraveling technique, such results can indeed be obtained for certain classes of models; two simple examples are the class of reflexive models and the class of symmetric models of the similarity type with just one unary modality. However, for certain classes of models the situation appears to be much less straightforward. In particular, we soon run into trouble when we consider classes of *transitive* models. The root of the problem lies in the fact that van Benthem's result is obtained using de Rijke's characterization result as a crucial lemma, and the latter result makes use of the so-called *finite depth property*. If we consider just a single unary modality, then the finite depth property says that for every formula ϕ there exists a natural number n such that we have $(M, u) \models \phi$ iff $(M \upharpoonright_u^n, u) \models \phi$ for any model (M, u) . Here $M \upharpoonright_u^n$ is the submodel of M generated by the set of nodes v such that v is reachable from u in at most n steps, i.e. there are w_1, \dots, w_k such that

$$uRw_1R \dots Rw_kRv$$

and $k \leq n - 1$. For transitive models, the finite depth property becomes rather trivial, and not very useful: if (M, u) is a transitive model, then $M \upharpoonright_u^n = M \upharpoonright_u^1$ for any $n \geq 1$. Furthermore, while in general the model $(M \upharpoonright_u^n, u)$ may differ radically from (M, u) , if M is a transitive model it will always be the case that (M, u) and $(M \upharpoonright_u^n, u)$ are bisimilar. Since the crucial step in van Benthem's result is to show that compactness + bisimulation invariance + relativization property implies the finite depth property, and then exploit this property as in de Rijke's proof, it seems we must look for some other property of \mathcal{ML} to use for characterization results that are general enough to include the class of transitive models.

My suggestion is to follow the approach taken by Otto and Piro in their characterization of \mathcal{ML}_\forall , modal logic with the global modality. In their characterization, they use three different properties of \mathcal{ML}_\forall : bisimulation invariance (in a modified sense), compactness and, instead of relativization, the so-called *Tarski union property*. An abstract modal logic \mathcal{L} has the Tarski union property if, for every countable sequence of models $(M_i)_{i \in \omega}$ such that M_i is an \mathcal{L} -elementary submodel of M_{i+1} – that is, for every $u \in M_i$ and every \mathcal{L} -formula ϕ we have $(M_i, u) \models \phi$ iff $(M_{i+1}, u) \models \phi$ – it holds that $\bigcup_{i \in \omega} M_i$ is an \mathcal{L} -elementary extension of each M_i . Here, the union of the models M_i is defined in the obvious manner.

Using this format we can easily get a characterization of \mathcal{ML} over the class of transitive models, and it is quite a natural solution to the problem: there is a certain analogy between the basic modal logic of transitive models and the logic \mathcal{ML}_\forall equipped with the global modality. In \mathcal{ML}_\forall , we are able to quantify over the entire universe of a model. On the other hand, in a transitive model, the standard modality \Box allows quantification over the set of all nodes that are finitely reachable from the point of evaluation u . While this doesn't in general amount to quantification over the entire universe, it does allow us to quantify over the part of the model that *matters*, so to speak, since the restriction of the model M to the “domain of quantification” for \Box yields a model bisimilar with (M, u) .

In general, we get the following result:

Theorem 2. *Let C be a class of models based on a class of frames definable by set of universal Horn clauses, i.e. first order sentences of the form*

$$\forall \vec{x} (\alpha_1 \wedge \dots \wedge \alpha_n \rightarrow \beta)$$

where $\alpha_1 \dots \alpha_n$ and β are all atomic formulas. Then \mathcal{ML} is the strongest abstract modal logic over C with compactness, bisimulation invariance and the Tarski union property over C . In other words, if \mathcal{L} is compact over C , bisimulation invariant over C and has the Tarski union property over C , then $\mathcal{L} \equiv_C \mathcal{ML}$.

Here, the property of “having the Tarski union property over C ” means that, for any \mathcal{L} -elementary chain within C , the union of the chain is a model in C and is an \mathcal{L} -elementary extension of every member of the chain. Since the class of transitive frames is definable by the universal Horn clause

$$\forall x \forall y \forall z (xRy \wedge yRz \rightarrow xRz)$$

this result implies the aforementioned characterization of \mathcal{ML} over transitive models. It also yields a characterization for the class of symmetric models, which is definable by the Horn clause $\forall x \forall y (xRy \rightarrow yRx)$, a characterization for the class of reflexive models which is definable by the Horn clause $\forall x (xRx)$, a characterization for the class of euclidean models which is definable by $\forall x \forall y \forall z (xRy \wedge xRz \rightarrow yRz)$, and many others.

Also, in a certain sense, we get Otto and Piro’s characterization of \mathcal{ML}_\forall as a special case of this theorem. Instead of viewing \mathcal{ML}_\forall as an extension of modal logic with a single box operator, we can view it, alternatively, as a restriction of the basic modal logic with two operators A, \Box to a special class of models, namely those satisfying the constraint

$$\forall x \forall y (x R_A y)$$

where R_A is the accessibility relation corresponding to the operator A . Within this class of models, the operator A is indeed interpreted as a global modality. Furthermore, the condition defining the class is a universal Horn clause (in which the set of conjuncts in the antecedent is empty), so it is covered by the general characterization result. Since the property of bisimulation invariance relative to this restricted class of models amounts to exactly the notion of bisimulation invariance used in Otto and Piro’s theorem, what we get is essentially the same characterization result as theirs.

It is natural to ask whether the general Lindström theorem for Horn clause definable model classes can be extended to any first-order elementary class. The answer to this question, it turns out, is *no*. Another interesting question is whether the general characterization result holds if we replace the Tarski union property with relativization; this question is left open. Other open problems include extensions of the characterization result to classes of models given by frame conditions that are not definable by universal Horn clauses. One particularly interesting example here is the class of models with the *Church-Rosser property*:

$$\forall x \forall y \forall z (x R y \wedge x R z \rightarrow \exists s (y R s \wedge z R s))$$

The class of frames with the Church-Rosser property is not definable by a set of universal Horn clauses, so the characterization theorem of \mathcal{ML} over Horn clause definable frame classes does not cover this case. A natural next step is therefore to look for more general characterization results, with more relaxed conditions on the formulas used to define the relevant frame classes.

References

- [1] Barwise, J. and S. Feferman (Eds.), *Model-theoretic logics*. Springer Verlag, 1985.
- [2] Blackburn, P., M. de Rijke and Y. Venema, *Modal logic*. Cambridge University Press, 2001.
- [3] Kurz, A. and Y. Venema, “Coalgebraic Lindström theorems”. In *Advances in Modal Logic (Aiml 2010)*.
- [4] Lindström, P., *On extensions of elementary logic*. Theoria, 35: 1–11, 1969.
- [5] Otto, M. and R. Piro, *A Lindström characterization of the guarded fragment and of modal logic with a global modality*. In *Advances in Modal Logic (Aiml 2008)*.
- [6] van Benthem, J., *A new modal Lindström theorem*. Logica Universalis, 1: 125–138, 2007.
- [7] van Benthem, J., ten Cate, B. and J. Väänänen, *Lindström theorems for fragments of first-order logic*. Logical Methods in Computer Science, 5: 1–27, 2009.

Terminating sequent calculi for proving and refuting formulas in **S4**

Camillo Fiorentini

Dipartimento di Informatica
Università degli Studi di Milano
Via Comelico 39, 20135 Milano, Italy
fiorentini@di.unimi.it

Proof search for modal logic has been deeply investigated in the literature (see e.g. [2, 4, 9] for a review). Here we consider the modal logic **S4**, semantically characterized by finite Kripke models having a reflexive and transitive accessibility relation [1]. These conditions are tricky to handle from a proof search viewpoint. Standard sequent calculi for **S4** are harmful from a proof search viewpoint, since bottom-up application of the rules give rise to non-terminating derivations. To overcome the problem, one has to implement some mechanism to narrow the search space. A typical approach is loop-checking: whenever the “same” sequent occurs twice along a branch of the proof under construction, the search is cut (see, e.g., [5]). Other solutions exploit auxiliary tools to explicitly represent the accessibility relation inside the calculus, see for instance the labelled calculi [4]. Recent developments are [7, 12], where modalities are annotated by special indexes with the aim to block the application of a rule which might cause non-termination.

We present a G3-style [14] sequent calculus **GS4** for **S4** such that all the rules are *decreasing* and enjoy the *subformula property*. This means that, after a backward application of a rule to a sequent σ , the obtained sequents σ' are smaller than σ and the formulas of σ' are built-up using the formulas occurring in σ . Accordingly, the calculus is terminating and proof search is feasible in finite time. Following [6, 10], we work on sequents in Mints-like normal form [8]. The accessibility relation is implicitly represented by the rules of the calculus (see the classification [4]); to capture the semantics of some formulas (in particular, formulas $\Box(E \vee \Diamond F)$) we introduce two new modal connectives. Sequents are decomposed by the rules until we get *cluster sequents*. A cluster sequent $\Gamma, \Gamma_{cl} \xRightarrow{cu} \Delta_{cl}$ is a special sequent where Γ_{cl} and Δ_{cl} only contain classical formulas (namely, formulas not containing modal operators), while Γ contains formulas $\Box A$ and $\Box \Diamond B$, with A and B classical formulas. Reasoning on cluster sequents can be accomplished within classical logic; we treat cluster sequents as initial sequents and we exploit some external device (for instance, a SAT-solver) to deal with them.

Beside **GS4** we introduce a refutation sequent calculus **RS4** strictly related to **GS4** and having the same nice properties (decreasing rules, subformula property). A proof of σ in **RS4**, we call it a *refutation* of σ , can be viewed as a “constructive proof” of the non-provability of σ in **S4**. Refutations calculi for modal logics

are known since the nineties, see for instance [3, 13]; however, these calculi have non-decreasing rules, so they are not suited for proof search.

We present an algorithm that, given a sequent σ (in normal form), outputs either a proof of σ in **GS4** or a refutation of σ in **RS4**. From a refutation π of a sequent $\sigma = \Gamma \Rightarrow \Delta$, we can build a *model of σ* , namely an **S4**-model such that, at some world w , all the formulas in Γ are true and all the formulas in Δ are false. Let us call *cluster model* an **S4**-model \mathcal{M}_{clu} only containing one cluster (i.e., for every world w, w' of \mathcal{M}_{clu} , w is a successor of w'). If a cluster sequent σ_{clu} is not provable, one can find out a cluster model \mathcal{M}_{clu} of σ_{clu} . Now, suppose to have a refutation π of a sequent σ . By definition, the initial sequents of π are non-provable cluster sequents. Let us take, for every initial sequent σ_{clu} of π , a cluster model \mathcal{M}_{clu} of σ_{clu} ; then, according to the structure of π , we can “glue” the given cluster models so to obtain a model \mathcal{M} of σ .

Our approach is strictly related to [6, 10], where formulas in Mints normal form are used. Calculi in [6, 10] use slight different normal forms. In [6], completeness is proved by syntactical techniques and models construction is not dealt with. The calculus [10] has some non-decreasing rules, and the related decision algorithm, based on model generation, exploits loop-check. The idea of combining proofs and refutations goes back to [11], where Intuitionistic Logic is studied; refutations are used to build up Kripke models of non-valid formulas. We aim to extend the techniques used for **S4** to other modal logics defined by transitive models.

We have developed a prototypical Prolog implementation; cluster sequents are decided using `logic2cnf`, an extension of minisat SAT-solver. Additional material is available at <http://homes/dsi/unimi/it/~fiorentini/g4>.

References

1. A. Chagrov and M. Zakharyashev. *Modal Logic*. Oxford University Press, 1997.
2. M. Fitting. Modal proof theory. In Johan Van Benthem Patrick Blackburn and Frank Wolter, editors, *Handbook of Modal Logic*, volume 3 of *Studies in Logic and Practical Reasoning*, pages 85 – 138. Elsevier, 2007.
3. V. Goranko. Refutation systems in modal logic. *Studia Logica*, 53(2):299–324, 1994.
4. R. Goré. Tableau Methods for Modal and Temporal Logics. In M. D’Agostino et al., editor, *Handbook of tableau methods*, pages 297–396. Kluwer Academic Press, 1999.
5. A. Heuerding, A. Seyfried, and A. Zimmermann. Efficient loop-check for backward proof search in some non-classical propositional logics. In P. Miglioli et al., editor, *TABLEAUX*, volume 1071 of *LNCS*, pages 210–225. Springer, 1996.
6. J. Hudelmaier. Improved Decision Procedures for the Modal Logics K, T, and S4. In H. K. Büning, editor, *CSL*, volume 1092 of *LNCS*, pages 320–334. Springer, 1995.
7. D. Leszczyńska-Jasion. A loop-free decision procedure for modal propositional logics K4, S4 and S5. *J. Philos. Logic*, 38(2):151–177, 2009.
8. G. Mints. Gentzen-type systems and resolution rules. Part I. Propositional logic. In P. Martin-Löf and G. Mints, editors, *Conference on Computer Logic*, volume 417 of *LNCS*, pages 198–231. Springer, 1988.

9. S. Negri. Proof theory for modal logic. *Philosophy Compass*, 6(8):523–538, 2011.
10. L. A. Nguyen. A New Space Bound for the Modal Logics K4, KD4 and S4. In M. Kutylowski et al., editor, *MFCS*, volume 1672 of *LNCS*, pages 321–331. Springer, 1999.
11. L. Pinto and R. Dyckhoff. Loop-free construction of counter-models for intuitionistic propositional logic. In Behara et al., editor, *Symposia Gaussiana, Conference A*, pages 225–232. Walter de Gruyter, Berlin, 1995.
12. R. Pliuskevicius and A. Pliuskeviciene. A new method to obtain termination in backward proof search for modal logic S4. *J. Log. Comput.*, 20(1):353–379, 2010.
13. T. Skura. A Łukasiewicz-style refutation system for the modal logic S4. *J. Philos. Logic*, 24(6):573–582, 1995.
14. A.S. Troelstra and H. Schwichtenberg. *Basic Proof Theory*, volume 43 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1996.

Recovering theories from their models

Henrik Forssell
University of Oslo

Abstract

Given a theory \mathbb{T} and its category of models and homomorphisms $\text{Mod}_{\mathbb{T}}$, is it possible to recover \mathbb{T} from $\text{Mod}_{\mathbb{T}}$, up to some suitable notion of equivalence of theories, and perhaps by equipping $\text{Mod}_{\mathbb{T}}$ with some additional structure? A topological and topos theoretic approach to this question is presented with respect to certain fragments of first order logic and classical first-order logic, leading to a first-order syntax-semantics duality. A topological characterization of the definable subclasses of a class of models is given as an application.

The question of recovering a theory from its models was given a positive and elegant answer by Makkai in [6] for the case of regular theories. Given a (first-order, possibly many-sorted) signature Σ , we take a *regular formula* ϕ to be one constructed using only the connectives \top , \wedge , and \exists ; a *regular sequence* $\phi \vdash_{\mathbf{x}} \psi$ to be a sequence where both ϕ and ψ are regular; and a regular theory to be a deductively closed set of regular sequents. We can represent a regular theory \mathbb{T} by its *syntactical category* $\mathcal{C}_{\mathbb{T}}$, the objects of which are regular formulas $\phi(\mathbf{x})$. With \mathbb{T} regular, $\mathcal{C}_{\mathbb{T}}$ is a regular category with the property that the category of \mathbb{T} -models and homomorphisms in any regular category \mathcal{R} is equivalent to the category of regular functors from $\mathcal{C}_{\mathbb{T}}$ to \mathcal{R} (naturally in \mathcal{R}). In particular, $\text{Mod}_{\mathbb{T}} \simeq \mathbf{Reg}(\mathcal{C}_{\mathbb{T}}, \mathbf{Set})$, where \mathbf{Set} is the category of sets and functions. $\mathcal{C}_{\mathbb{T}}$ embeds in the category of set-valued functors on $\text{Mod}_{\mathbb{T}}$,

$$\mathcal{C}_{\mathbb{T}} \hookrightarrow [\text{Mod}_{\mathbb{T}}, \mathbf{Set}]$$

by sending a formula $\phi(\mathbf{x})$ to the ‘definable set’ functor which takes a model to the extension of the formula in that model, $\mathbf{M} \mapsto \phi(\mathbf{M})$. It is shown in [6] that the image of this embedding is, up to effective completion, the full subcategory of functors that preserve (small) products and filtered colimits. Thus a regular theory can be recovered (in the form of its syntactic category up to effective completion) from its models as the category of set-valued functors on models that preserve products and filtered colimits. In addition, the results and constructions of [6] serve to establish that the subcategory of filtered colimit preserving functors $\text{FC}(\text{Mod}_{\mathbb{T}}, \mathbf{Set})$ is the so-called *classifying topos* $\mathbf{Set}[\mathbb{T}]$ of \mathbb{T} . This topos can also be constructed ‘syntactically’ by taking sheaves on $\mathcal{C}_{\mathbb{T}}$ (equipped with a certain Grothendieck coverage), $\mathbf{Set}[\mathbb{T}] \simeq \text{Sh}(\mathcal{C}_{\mathbb{T}})$, from which $\mathcal{C}_{\mathbb{T}}$ can be recovered as the stably supercompact objects (again up to effective completion). Thus Makkai’s construction yields a semantic representation of this topos in addition to the previously known syntactical one (and an additional characterization of $\mathcal{C}_{\mathbb{T}}$ in the semantic representation).

As an alternative construction, it is possible to recover \mathbb{T} from $\text{Mod}_{\mathbb{T}}$ by equipping the latter category with topological structure. Using a ‘logical’ topology (similar to certain topologies used in descriptive set theory, see e.g. the expository [3]), it can be shown that being a filtered colimit preserving functor from $\text{Mod}_{\mathbb{T}}$ to \mathbf{Set} is the ‘same thing’ as being an equivariant sheaf of $\text{Mod}_{\mathbb{T}}$ considered as a topological category. That is, there is an equivalence

$$\text{FC}(\text{Mod}_{\mathbb{T}}, \mathbf{Set}) \simeq \text{Sh}_{H_{\mathbb{T}}}(M_{\mathbb{T}})$$

where $M_{\mathbb{T}}$ is a (sufficiently large) space of \mathbb{T} -models equipped with the logical topology, $H_{\mathbb{T}}$ is the space of homomorphisms between them, and $\text{Sh}_{H_{\mathbb{T}}}(M_{\mathbb{T}})$ is the topos of sheaves on $M_{\mathbb{T}}$ equipped with a continuous action of $H_{\mathbb{T}}$. Thus $\text{Sh}_{H_{\mathbb{T}}}(M_{\mathbb{T}})$ is (another representation of) the classifying topos $\mathbf{Set}[\mathbb{T}]$, from which we can, then, recover $\mathcal{C}_{\mathbb{T}}$.

Although this alternative construction is clearly somewhat redundant in the regular case, the topological approach becomes more relevant when passing to *coherent* logic, by allowing also the connectives \perp and \vee . Considering coherent theories also means considering classical first-order theories, in so far as any classical first-order theory can be ‘Morleyized’ to obtain a coherent theory (over a different signature) with the same models (and the same syntactic category, see [4]). Coherent theories cannot in general be recovered from their categories of models and homomorphisms by considering only structure intrinsic to those categories. Thus it becomes necessary to equip $\text{Mod}_{\mathbb{T}}$ with some additional structure. Makkai’s solution in this case is to use additional structure based on ultra-products (see [5], [7]). However, the topological approach above extends directly to this case, yielding an equivalence

$$\mathbf{Set}[\mathbb{T}] \simeq \text{Sh}_{H_{\mathbb{T}}}(M_{\mathbb{T}}) \tag{1}$$

also for a coherent \mathbb{T} , with the topological category of models defined as in the regular case. Since $\mathcal{C}_{\mathbb{T}}$ can be recovered from $\mathbf{Set}[\mathbb{T}]$ up to pretopos completion as the stably compact objects in the coherent case (see [4]), this yields a way to recover a coherent theory from its models and homomorphisms equipped with topological structure.

Strictly speaking, in order to use the logical topology referred to above to obtain (1), it is necessary to restrict to coherent theories in which inequality is definable (a class which still includes all Morleyized first-order theories). As shown in [1] (building on results in [2]), that restriction also makes it possible to consider only isomorphisms between theories and represent the classifying topos of such a theory as equivariant sheaves on the topological groupoid of \mathbb{T} -models and isomorphisms; i.e. there is an equivalence

$$\mathbf{Set}[\mathbb{T}] \simeq \text{Sh}_{I_{\mathbb{T}}}(M_{\mathbb{T}}) \tag{2}$$

where $I_{\mathbb{T}} \subseteq H_{\mathbb{T}}$ is the subspace of isomorphisms. This semantic representation of the classifying topos of a coherent theory can then be used to extend Stone duality to first-order logic by constructing a duality between coherent theories (with definable inequality) on one side and a category of ‘semantical’ topological groupoids on the other.

The topological approach to the question of recovering a theory from its models springs from the dual syntax-semantics (and algebra-geometry) aspects

of Grothendieck toposes. A further illustration of these aspects and application of the approach is given by considering the connection between subtoposes, quotient theories, and subgroupoids. Given a theory \mathbb{T} consider its classifying topos with its syntactic and semantic representations

$$\mathrm{Sh}(\mathcal{C}_{\mathbb{T}}) \simeq \mathbf{Set}[\mathbb{T}] \simeq \mathrm{Sh}_{I_{\mathbb{T}}}(M_{\mathbb{T}})$$

as sheaves on the syntactic category $\mathcal{C}_{\mathbb{T}}$ and as equivariant sheaves on the topological groupoid $(I_{\mathbb{T}}, M_{\mathbb{T}})$ of models and isomorphisms, respectively. On the groupoid side, subgroupoids of $(I_{\mathbb{T}}, M_{\mathbb{T}})$ can be identified with subsets of $M_{\mathbb{T}}$ closed under isomorphisms, and there is a Galois connection between such sets and subtoposes of $\mathrm{Sh}_{I_{\mathbb{T}}}(M_{\mathbb{T}})$. Moreover, the subsets arising from subtoposes can be intrinsically characterized in terms of the topological groupoid. Combining this with the well known correspondence between subtoposes of $\mathrm{Sh}(\mathcal{C}_{\mathbb{T}})$ and quotient theories of \mathbb{T} , this yields a topological characterization of those subsets of \mathbb{T} -models that are definable by quotient theories of \mathbb{T} .

References

- [1] Steve Awodey and Henrik Forssell. First-order logical duality. Preprint, <http://arxiv.org/abs/1008.3145> (submitted).
- [2] Carsten Butz and Ieke Moerdijk. Representing topoi by topological groupoids. *Journal of Pure and Applied Algebra*, 130:223–235, 1998.
- [3] Greg Hjorth. Group actions and countable models. <http://www.math.ucla.edu/~greg/as1/utrecht.dvi>.
- [4] Peter T. Johnstone. *Sketches of an Elephant*, volume 43 and 44 of *Oxford Logic Guides*. Clarendon Press, Oxford, 2002.
- [5] Michael Makkai. Stone duality for first order logic. *Advances in Mathematics*, 65(2):97–170, 1987.
- [6] Michael Makkai. A theorem on barr-exact categories, with an infinitary generalization. *Annals of Pure and Applied Logic*, 47:225–268, 1990.
- [7] Michael Makkai. *Duality and Definability in First Order Logic*. Number 503 in *Memoirs of the American Mathematical Society*. American Mathematical Society, 1993.

The Doxastic Interpretation of Team Semantics

Pietro Galliani
ILLC
University of Amsterdam
The Netherlands
(pgallian@gmail.com)

May 29, 2012

Dependence Logic [17] is an extension of First Order Logic which adds to its language *dependence atoms* of the form $=(\vec{t}_1, \vec{t}_2)$, where \vec{t}_1 and \vec{t}_2 are tuples of terms¹, with the intended interpretation of “the value of \vec{t}_2 is a function of the value of \vec{t}_1 .” It is a first-order *logic of imperfect information*, like IF Logic [11, 10, 15] or Branching Quantifier Logic [9]; but rather than adding new possible patterns of dependence or independence between *quantifiers*, as these logics do, Dependence Logic isolates the notion of dependence away from the one of quantification and permits the examination of patterns of dependence and independence between *variables* or, more in general, between tuples of terms.

This different outlook makes Dependence Logic a most suitable framework for the formal study, in a first-order setting, of functional dependence itself; and, furthermore, this logic is readily adaptable to the analysis of other, non-functional notions of dependence or independence [8, 5, 6].

Like other logics of imperfect information, Dependence Logic admits both a Game Theoretic Semantics, an imperfect information variant of the one for First Order Logic, and a *Team Semantics*, a compositional semantics which is a natural adaptation of Hodges’ Trump Semantics [12]. One striking peculiarity of the current state of the art of the research in Dependence Logic and its extensions is a willingness to take Team Semantics – and not Game Theoretic Semantics, as for the case of much IF Logic research – as the fundamental semantic framework; and this different approach is at the root of many recent technical developments in the field, such as, for example, the characterizations of team class definability of [14], [13] and [6], the hierarchy results of [3], and the study of notions of generalized quantification of [5] and [4].

I will give a detailed account of a *doxastic* interpretation for Team Semantics, according to which formulas are to be interpreted as assertions about beliefs and belief updates. This is not a novel idea: as a matter of fact, it is already implicit in the equivalence proof between Trump Semantics and Game Theoretic Semantics of [12]. However, the consequences of this insight are far from fully explored: until now, doxastic concerns have played very little role in the development of extensions of Dependence Logic, and the doxastic meanings of known connectives have been left largely unexamined.

In this talk, I will gradually develop a formal system – in essence, a notational variant of Jouko Väänänen’s Team Logic [18] – and show that many of the atoms, connectives and operators of Team Semantics arise naturally from concerns about beliefs and belief updates.

References

- [1] Samson Abramsky and Jouko Väänänen. From IF to BI. *Synthese*, 167:207–230, 2009. 10.1007/s11229-008-9415-6.

¹Sometimes only atoms of the form $=(\vec{t}, t')$ are taken as primitive, where t' is a single term. $=(\vec{t}_1, \vec{t}_2)$ is then equivalent to $\bigwedge_{t' \in \vec{t}_2} =(\vec{t}_1, t')$.

- [2] Arthur Dempster. Upper and lower probabilities induced by a multivalued mapping. In Roland Yager and Liping Liu, editors, *Classic Works of the Dempster-Shafer Theory of Belief Functions*, volume 219 of *Studies in Fuzziness and Soft Computing*, pages 57–72. Springer Berlin / Heidelberg, 2008.
- [3] Arnaud Durand and Juha Kontinen. Hierarchies in dependence logic. *CoRR*, abs/1105.3324, 2011.
- [4] F. Engström and J. Kontinen. Characterizing quantifier extensions of dependence logic. arXiv:1202.5247, 2012.
- [5] Fredrik Engström. Generalized quantifiers in dependence logic. *Journal of Logic, Language and Information (to appear)*, 2011.
- [6] Pietro Galliani. Inclusion and exclusion dependencies in team semantics: On some logics of imperfect information. *Annals of Pure and Applied Logic*, 163(1):68 – 84, 2012.
- [7] Peter Gardenfors, editor. *Belief Revision*. Cambridge University Press, New York, NY, USA, 1992.
- [8] Erich Grädel and Jouko Väänänen. Dependence and Independence. *Studia Logica (to appear)*, 2010.
- [9] Leon Henkin. Some Remarks on Infinitely Long Formulas. In *Infinitistic Methods. Proc. Symposium on Foundations of Mathematics*, pages 167–183. Pergamon Press, 1961.
- [10] Jaakko Hintikka. *The Principles of Mathematics Revisited*. Cambridge University Press, 1996.
- [11] Jaakko Hintikka and Gabriel Sandu. Informational independence as a semantic phenomenon. In J.E Fenstad, I.T Frolov, and R. Hilpinen, editors, *Logic, methodology and philosophy of science*, pages 571–589. Elsevier, 1989.
- [12] Wilfrid Hodges. Compositional Semantics for a Language of Imperfect Information. *Journal of the Interest Group in Pure and Applied Logics*, 5 (4):539–563, 1997.
- [13] Juha Kontinen and Ville Nurmi. Team logic and second-order logic. In Hiroakira Ono, Makoto Kanazawa, and Ruy de Queiroz, editors, *Logic, Language, Information and Computation*, volume 5514 of *Lecture Notes in Computer Science*, pages 230–241. Springer Berlin / Heidelberg, 2009.
- [14] Juha Kontinen and Jouko Väänänen. On definability in dependence logic. *Journal of Logic, Language and Information*, 3(18):317–332, 2009.
- [15] Allen L. Mann, Gabriel Sandu, and Merlijn Sevenster. *Independence-Friendly Logic: A Game-Theoretic Approach*. Cambridge University Press, 2011.
- [16] Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.
- [17] Jouko Väänänen. *Dependence Logic*. Cambridge University Press, 2007.
- [18] Jouko Väänänen. Team Logic. In J. van Benthem, D. Gabbay, and B. Löwe, editors, *Interactive Logic. Selected Papers from the 7th Augustus de Morgan Workshop*, pages 281–302. Amsterdam University Press, 2007.
- [19] Lotfi A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1(1):3 – 28, 1978.

First-order theories of classes of trees associated with classes of linear orders

VALENTIN GORANKO¹ AND RUAAN KELLERMAN²

¹Department of Informatics and Mathematical Modelling,
Technical University of Denmark

²Department of Mathematics and Applied Mathematics,
University of Pretoria, South Africa

Emails: vfgo@imm.dtu.dk, ruaan.kellerman@up.ac.za

1 Introduction

Every class of linear order types \mathcal{A} determines a class $\mathcal{T}(\mathcal{A})$ of those trees in which the order type of every path belongs to \mathcal{A} . Conversely, every class of trees \mathcal{B} gives rise to a class $\mathcal{L}(\mathcal{B})$ of those linear order types that occur as paths in the trees in \mathcal{B} .

Besides $\mathcal{T}(\mathcal{A})$, several naturally arising classes of trees associated with any given class of linear order types \mathcal{A} are identified in [3, 1]. They are defined in terms of how the paths in these trees and the first-order theories of these trees relate to the order types in \mathcal{A} and to their first-order theories.

The general problem of the present study is the *transfer of logical properties, including expressiveness, axiomatizations and decidability of logical theories, from classes of linear orders to the respective classes of trees*. Such study was initiated in [2] for the case of temporal logics, and in [1] for the case of first-order logic. The main question studied in the latter paper is how to characterize the first-order theory of a class of trees defined in terms of a class of linear order types \mathcal{A} , by means of the first-order theory of \mathcal{A} . Some partial results on that were reported in [3, 1], of which the work reported here is a continuation.

Particularly interesting is the case when $\mathcal{A} = \{\alpha\}$, where α is an ordinal with $\alpha < \omega^\omega$, because the first-order theories of these ordinals are well-known [4]. The class $\mathcal{T}(\{\alpha\})$ then consists of all trees of which the paths are isomorphic to α . Explicit axiomatizations of the first-order theories of $\mathcal{T}(\{\alpha\})$ and the other classes of trees associated with α is not generally known yet, although the problem is solved in [3, 1] for some cases, including finite α and $\alpha = \omega$. Here we present a partial solution to the problem for the case where α is a successor ordinal with $\omega < \alpha < \omega^\omega$.

2 Preliminaries: Trees and \mathcal{A} -classes of trees

A *tree* is a partially ordered set $\mathfrak{T} = (T; <)$ such that for every $x \in T$ the set $\{y \in T : y < x\}$ is totally ordered and for all $x, y \in T$, there exists $z \in T$ such that $z \leq x, y$. A maximal totally ordered subset of a tree is called a *path*. A maximal node in a tree is called a *leaf*. The set of leaves in a tree can be defined using the formula $\text{leaf}(x) := \forall y (x \leq y \rightarrow x = y)$. Given a tree $\mathfrak{T} = (T; <)$ and a node a in \mathfrak{T} , define $a^> = \{x \in T : x < a\}$, $C(a) = \{x \in T : a \not\leq x\}$ and $\mathfrak{T}^a = (C(a); <|_{C(a)})$. The relativization of a sentence σ to the formula $\theta(u, x) = u \leq x$ is denoted as $\sigma^{\leq x}$; hence with $T^{\leq a} = \{x \in T : x \leq a\}$, we have that $\mathfrak{T} \models \sigma^{\leq x}(a/x)$ if and only if $(T^{\leq a}; <|_{T^{\leq a}}) \models \sigma$. The *rank* of a formula is the number of distinct variables which occur in the formula. The *quantifier rank* of a formula is the number of distinct variables in the formula which are bound by a quantifier. Thus the rank of a formula is the sum of its quantifier rank and the number of distinct freely occurring

variables in the formula. Two structures \mathfrak{A} and \mathfrak{B} are called *n-equivalent*, denoted $\mathfrak{A} \equiv_n \mathfrak{B}$, when \mathfrak{A} and \mathfrak{B} satisfy the same sentences of quantifier rank up to n , and *elementarily equivalent*, denoted $\mathfrak{A} \equiv \mathfrak{B}$, when \mathfrak{A} and \mathfrak{B} are n -equivalent for every natural number n . The notation $\mathfrak{A} \preceq_n \mathfrak{B}$ indicates that $|\mathfrak{A}| \subseteq |\mathfrak{B}|$ (where $|\mathfrak{A}|$ denotes the domain of \mathfrak{A}) and, for every formula $\varphi(x_0, x_1, \dots, x_k)$ of rank n and for any elements a_1, \dots, a_k in $|\mathfrak{A}|$, if $\mathfrak{B} \models \exists x_0 \varphi(x_0, a_1/x_1, \dots, a_k/x_k)$ then there is an element a_0 in $|\mathfrak{A}|$ such that $\mathfrak{B} \models \varphi(a_0/x_0, a_1/x_1, \dots, a_k/x_k)$. $\mathfrak{A} \preceq \mathfrak{B}$ indicates that $\mathfrak{A} \preceq_n \mathfrak{B}$ for every natural number n .

Let \mathcal{A} be a class of linear order types and let $\mathfrak{T} = (T; <)$ be a tree. A path A in \mathfrak{T} is called an α -path when $(A; <|_A)$ is isomorphic to α . The tree \mathfrak{T} is called

- (i) an \mathcal{A} -tree when every path in \mathfrak{T} is an α -path for some $\alpha \in \mathcal{A}$;
- (ii) a *uniformly \mathcal{A} -like tree* (*U- \mathcal{A} -like tree*) when $\mathfrak{T} \equiv \mathfrak{G}$ for some \mathcal{A} -tree \mathfrak{G} ;
- (iii) an \mathcal{A} -like tree if, for every $n \in \mathbb{N}$, there is an \mathcal{A} -tree \mathfrak{G} such that $\mathfrak{T} \equiv_n \mathfrak{G}$;
- (iv) a *pathwise uniformly \mathcal{A} -like tree* (*PU- \mathcal{A} -like tree*) if, for every path X in \mathfrak{T} , there exists $\alpha \in \mathcal{A}$ such that $(X; <|_X) \equiv \alpha$;
- (v) a *pathwise \mathcal{A} -like tree* (*P- \mathcal{A} -like tree*) if, for every path X in \mathfrak{T} and for every $n \in \mathbb{N}$, there exists $\alpha \in \mathcal{A}$ such that $(X; <|_X) \equiv_n \alpha$;
- (vi) a *definably \mathcal{A} -tree* (*D- \mathcal{A} -tree*) if every parametrically definable path X in \mathfrak{T} is an α -path for some $\alpha \in \mathcal{A}$ dependent on X ;
- (vii) a *definably uniformly \mathcal{A} -like tree* (*DU- \mathcal{A} -like tree*) if, for every parametrically definable path X in \mathfrak{T} , there exists $\alpha \in \mathcal{A}$ such that $(X; <|_X) \equiv \alpha$;
- (viii) a *definably \mathcal{A} -like tree* (*D- \mathcal{A} -like tree*) if, for every parametrically definable path X in \mathfrak{T} and for every $n \in \mathbb{N}$, there exists $\alpha \in \mathcal{A}$ such that $(X; <|_X) \equiv_n \alpha$.

The eight classes of trees defined by these conditions are called \mathcal{A} -classes of trees. If $\mathcal{A} = \{\alpha\}$ then \mathfrak{T} is simply called an α -tree, a uniformly α -like tree, etc.

The set-theoretical relationships between the \mathcal{A} -classes of trees, and between their first-order theories, established in [3, 1], are summarized in Figures 1 and 2.

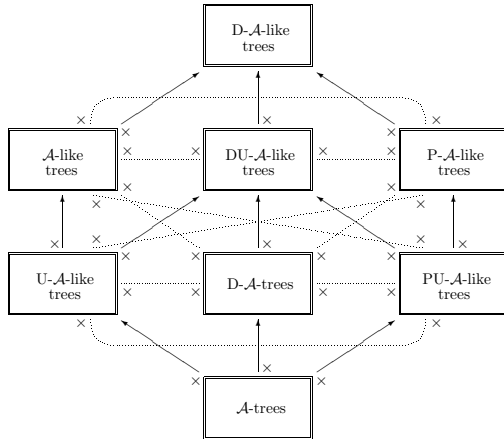


Figure 1: Set-theoretical relationships between \mathcal{A} -classes of trees. Inclusions $X \subseteq Y$ are denoted as $X \rightarrow Y$. Non-inclusions are indicated by \times .

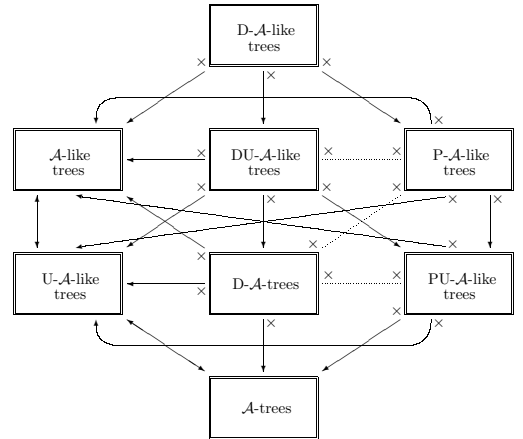


Figure 2: Set-theoretical relationships between the first-order theories of \mathcal{A} -classes of trees. Inclusions $X \subseteq Y$ are denoted as $X \rightarrow Y$. Non-inclusions are indicated by \times .

3 The first-order theory of the class of α -trees for an ordinal α

For any ordinal α with $\alpha < \omega^\omega$, let Φ_α denote a sentence which axiomatizes the first-order theory of α , see [4].

A tree $\mathfrak{T} = (T; <)$ satisfies the *cover property* if, for every natural number n and for any two paths A and B in \mathfrak{T} and any increasing sequences $(a_i)_{i \in \xi}$, cofinal in A , and $(b_i)_{i \in \xi}$, cofinal in B , such that $\mathfrak{T}^{a_i} \equiv_n \mathfrak{T}^{b_i}$ for every i with $i \in \xi$, it follows that $(\mathfrak{T}; A) \equiv_n (\mathfrak{T}; B)$.

We define the following axioms:

- Ir : $\forall x (x \not< x)$ (irreflexivity)
- Tr : $\forall x \forall y \forall z (x < y \wedge y < z \rightarrow x < z)$ (transitivity)
- ST : $\forall x \forall y \forall z (y < x \wedge z < x \rightarrow (y < z \vee y = z \vee z < y))$ (subtotalness)
- Co : $\forall x \forall y \exists z (z \leq x \wedge z \leq y)$ (connectedness)
- Do : $\forall x \exists y (\text{leaf}(y) \wedge x \leq y)$ (every node is dominated by a leaf)
- Le $_{\Phi_\alpha}$: $\forall x (\text{leaf}(x) \rightarrow \Phi_\alpha^{\leq x})$ (every path containing a leaf is elementarily equivalent to α)
- Ter : $\forall x \forall y (\text{leaf}(x) \wedge \forall z (z < x \leftrightarrow z < y) \rightarrow \text{leaf}(y))$ (every sibling of a leaf must also be a leaf)
- El : El is an axiom scheme consisting of the sentences

$$\forall \bar{z} \left(\forall x (\varphi(x, \bar{z}) \rightarrow \exists y (x < y \wedge \varphi(y, \bar{z}))) \rightarrow \right. \\ \left. \exists x \left(\text{leaf}(x) \wedge \forall y (y < x \rightarrow \exists u (y < u < x \wedge \varphi(u, \bar{z}))) \right) \right)$$

for every formula $\varphi(x, \bar{z})$ (including formulas $\varphi(x)$ for which the tuple \bar{z} is empty).

The scheme El states that for every infinite totally ordered chain of nodes satisfying the formula φ , there is a leaf b (an Elder) and a sequence of nodes, cofinal in $b^>$, which satisfy φ .

Let $T_\alpha = \{\text{Ir}, \text{Tr}, \text{ST}, \text{Co}, \text{Do}, \text{Le}_{\Phi_\alpha}, \text{Ter}\} \cup \text{El}$.

Theorem Let α be a successor ordinal with $\omega < \alpha < \omega^\omega$. Let \mathfrak{T} be a well-founded tree which satisfies the cover property and is a model of the theory T_α . For every positive integer n there exists an α -tree \mathfrak{S} such that $\mathfrak{T} \preceq_n \mathfrak{S}$.

We will discuss the consequences of this result for the first-order theories of $\mathcal{T}(\{\alpha\})$ and the other classes of α -trees.

References

- [1] V. Goranko and R. Kellerman, *Classes and Theories of Trees Associated with a Class of Linear Orders*, **Logic Journal of the IGPL**, vol. 19 (2011), pp. 217–232.
- [2] V. Goranko and A. Zanardo, *From Linear to Branching-Time Temporal Logics: Transfer of Semantics and Definability*, **Logic Journal of the IGPL**, vol. 15 (2007), pp. 53–76.
- [3] R. Kellerman, **Logical Theories of Trees**, Ph.D. thesis, University of the Witwatersrand, 2010.
- [4] J. Rosenstein, **Linear Orderings**, Academic Press, 1982.

Modeling pluralistic ignorance in epistemic logic

Jens Ulrik Hansen
ILLC, University of Amsterdam
E-mail: jensuh@ruc.dk

June 24, 2012

The term “pluralistic ignorance” originates in social psychology (O’Gorman, 1986) and refers to the phenomenon where “[...] *no one believes, but everyone believes that everyone else believes.*” (Krech and Crutchfield, 1948, p. 388–389). Put differently, pluralistic ignorance is “[...] *a social comparison error where an individual holds an opinion, but mistakenly believes that others hold the opposite opinion.*” (Halbesleben and Buckley, 2004, p. 126). Or yet differently, pluralistic ignorance is “[...] *a state characterized by the belief that one’s private thoughts, feelings, and behaviors are different from those of others, even though one’s public behavior is identical.*” (Miller and McFarland, 1991, p. 287).

Examples of pluralistic ignorance are plentiful in the literature. Classic examples includes drinking among college student, attitudes towards racial segregation, the absence of questions in a classroom full of students, and many more. Prentice and Miller (1993) found in a study of college students at Princeton that most students believed that the average student was much more comfortable with alcohol than they were themselves. Fields and Schuman (1976) conducted a study, which showed that on issues of racial and civil liberties most people perceived others to be more conservative than they actually were and O’Gorman and Garry (1976) found a tendency among whites to overestimate white support for racial segregation. After presenting some difficult material to a class of students, a teacher might experience that none of the students ask any questions even though none of the students understood the material and the teacher explicitly request them to ask questions if they did not understand the material. (An extensive study of this phenomenon was done by Miller and McFarland (1987).)

The three different characterizations of pluralistic ignorance mentioned above are just a few of those found in the literature. While they all apply to the paradigmatic cases of pluralistic ignorance, it seems plausible that there might be situations that would be classified as pluralistic ignorance by some of the characterizations while not by others. Focusing merely on the epistemic states of the involved agents supports this view as well. The different characterizations also differs in what epistemic states are ascribed to the involved individuals. Using an epistemic/doxastic logic to formalize different notions of pluralistic ignorance can help clarify exactly what the differences are.

A logic suitable for formalizing the different notions of pluralistic ignorance is the epistemic/doxastic logic based on plausibility models, such as the logic of Baltag and Smets (2008). In such a logic, “ $B_a\varphi$ ” formalizes that agent a believes φ . The notion of pluralistic ignorance expressed by Krech and Crutchfield (1948) can then be formalized as

$$\bigwedge_{a \in \mathbb{A}} (\neg B_a\varphi \wedge B_a(\bigwedge_{b \in \mathbb{A} \setminus \{a\}} B_b\varphi)). \quad (1)$$

The notion of pluralistic ignorance endorsed by Halbesleben and Buckley (2004) focuses only on the belief state of a single agent and can therefore be formalized in line with

$$\neg B_a \varphi \wedge B_a \left(\bigwedge_{b \in \mathbb{A} \setminus \{a\}} B_b \varphi \right). \quad (2)$$

Ignoring for a moment, the exact placement of the negation, what exactly φ is, and taking “opinion” to be the same as belief, the relation between these two formulas should be obvious – (1) is simply the conjunction of all the agents being in a belief state as described by (2). Now the notion of pluralistic ignorance advocated by Miller and McFarland (1991) is actually a little more involved, since it makes reference to behavior. As claimed by Bjerring et al. (2012) how agents involved in pluralistic ignorance act is important to the phenomenon. Adding an action expression $[a \text{ ActAs } \varphi]$ for every agent a expressing that “ a act as if φ ” allows us formalize Miller and McFarland (1991)’s notion of pluralistic ignorance as

$$\bigwedge_{a \in \mathbb{A}} \left(\neg B_a \varphi \wedge [a \text{ ActAs } \varphi] \wedge B_a \left(\bigwedge_{b \in \mathbb{A} \setminus \{a\}} B_b \varphi \right) \right). \quad (3)$$

However, how exactly to define the semantic of $[a \text{ ActAs } \varphi]$ is not obvious.

In addition, there might be more one can say about the epistemic states of agents involved in pluralistic ignorance, depending on how the phenomenon arose. For instance, Halbesleben et al. (2007) mention minority influence as one possible cause of pluralistic ignorance. It is the case when a minority of a group is believed by everybody to express the majority view. If a minority believes φ , but the entire group believes that this minority expresses the majority view of the group it might lead to the agents forming wrong belief about what the majority of the group believes – everyone might come to believe that everyone else believes φ . The result might be a case of pluralistic ignorance in line with (1). However, there is an additional belief among the group, namely that there is a dependency between the agents’ beliefs; whatever the minority believes is believed by the entire group. Taking the minority to be $B \subseteq \mathbb{A}$, one can formalize a situation of this kind as

$$\bigwedge_{a \in \mathbb{A}} \left(\neg B_a \varphi \wedge B_a \left(\bigwedge_{b \in B} B_b \varphi \leftrightarrow \bigwedge_{b \in \mathbb{A} \setminus \{a\}} B_b \varphi \right) \wedge \bigwedge_{b \in B} B_b \varphi \right). \quad (4)$$

Another cause for pluralistic ignorance is the fact that agents involved in the phenomenon has a tendency to believe that everyone else is different from themselves, and based on this belief, agents form the belief that everyone else believes something different than they do themselves. Again, this might be interpreted as every agent believes that everyone else’s beliefs are dependent, that is, every agent a believes that

$$B_{b_1} \varphi \leftrightarrow B_{b_2} \varphi \leftrightarrow \dots \leftrightarrow B_{b_n} \varphi, \quad \text{where } \mathbb{A} \setminus \{a\} = \{b_1, b_2, \dots, b_n\}. \quad (5)$$

Pluralistic ignorance can have several bad consequences and thus, it is of importance to investigate how the phenomenon can be avoided or dissolved. Depending on what kinds of epistemic states the agents involved in pluralistic ignorance are in, different measures need to be taken to dissolve the phenomenon. Therefore, the additional information about the agents’ belief states based on what have caused the phenomenon, can be quite important. For instance, if pluralistic ignorance has arisen due to minority influence and one can pick out the minority, then changing the apparent belief of the minority might dissolve pluralistic ignorance. If pluralistic ignorance has arisen because everyone thinks they are different from everyone else, then informing the agents

that at least one other agent is like them might dissolve the phenomenon. On the other hand, if nothing more is known about the agents' belief states, other than they satisfy (1), then the pluralistic ignorance can be extremely robust and not easy to dissolve (Hansen, 2012).

Modeling pluralistic ignorance using epistemic logic can give us many insights into the differences between the different notions of pluralistic ignorance by focusing on the involved agents' epistemic states. Furthermore, these differences can be quite important for what it takes to dissolve the phenomenon, which again can be clarified by a modeling in epistemic logic.

References

- Alexandru Baltag and Sonja Smets. A qualitative theory of dynamic interactive belief revision. In W. van der Hoek and M. Wooldridge, editors, *Logic and the Foundations of Game and Decision Theory, Texts in Logic and Games, Vol 3*, pages 9–58. Amsterdam University Press, 2008.
- Jens Christian Bjerring, Jens Ulrik Hansen, and Nikolaj Jang Lee Linding Pedersen. On the rationality of pluralistic ignorance. under review, April 2012.
- James M. Fields and Howard Schuman. Public beliefs about the beliefs of the public. *The Public Opinion Quarterly*, 40(4):427–448, 1976.
- Jonathon R. B. Halbesleben and M. Ronald Buckley. Pluralistic ignorance: historical development and organizational applications. *Management Decision*, 42(1):126–138, 2004.
- Jonathon R. B. Halbesleben, Anthony R. Wheeler, and M. Ronald Buckley. Understanding pluralistic ignorance: application and theory. *Journal of Managerial Psychology*, 22(1):65–83, 2007.
- Jens Ulrik Hansen. A logic-based approach to pluralistic ignorance. In Jonas De Vuyst and Lorenz Demey, editors, *Future Directions for Logic – Proceedings of PhDs in Logic III*, volume to appear. College Publications, 2012.
- David Krech and Richard S. Crutchfield. *Theories and Problems of Social Psychology*. New York: McGraw-Hill, 1948.
- Dale T. Miller and Cathy McFarland. Pluralistic ignorance; when similarity is interpreted as dissimilarity. *Journal of Personality and Social Psychology*, 53:298–305, 1987.
- Dale T. Miller and Cathy McFarland. When social comparison goes awry: The case of pluralistic ignorance. In J. Suls and T. Wills, editors, *Social comparison: Contemporary theory and research*, pages 287–313. Erlbaum, Hillsdale, NJ, 1991.
- Hubert J. O’Gorman. The discovery of pluralistic ignorance: An ironic lesson. *Journal of the History of the Behavioral Sciences*, 22(October):333–347, 1986.
- Hubert J. O’Gorman and Stephen L. Garry. Pluralistic ignorance – a replication and extension. *The Public Opinion Quarterly*, 40(4):449–458, 1976.
- Deborah A. Prentice and Dale T. Miller. Pluralistic ignorance and alcohol use on campus: Some consequences of misperceiving the social norm. *Journal of Personality and Social Psychology*, 64(2):243–256, 1993.

Brouwer's Free Choice Sequences and Bivalence

Casper Storm Hansen
Northern Institute of Philosophy
University of Aberdeen

The purpose of this talk is to give an internal critique of Brouwer's Intuitionism: it is argued that accepting his anti-realism and use of free choice sequences does not lead to a rejection of bivalence. Further it is claimed that he does not succeed in paralleling the classical analysis of the continuum as a point-set.

First the constitution of a lawless sequence is analyzed. Brouwer relies implicitly on such sequences being at any given stage of its development an object of the form

$$\langle a_1, \dots, a_n, _ , _ , _ , \dots \rangle ,$$

i.e., the sequence's momentary incarnation is as an infinite sequence with n specific terms followed by an infinity of "blank slots" which represent some sort of middle ground between non-existence of the term and "full" existence. On the one hand, the term must exist enough to be able to fill the rôle of carrier of properties. On the other hand, the term can not exist in so strong a sense that it has individual properties like "being equal to 12" which sets it apart from the other terms in a way that results in an actual infinity of distinct facts. It is argued that this is not consistent with Brouwer's metaphysical parsimony. Instead, at any given time a sequence is simply constituted as a finite object:

$$\langle a_1, \dots, a_n, \text{intention to expand} \rangle$$

Using this analysis it is argued that free choice sequences do not force a rejection of bivalence using a main argument which is backed up by considerations of definite descriptions, future contingents and fictional objects.

Main argument When there are no "shadowy" future terms, there are also no "fleeing" properties to threaten bivalence. The definition is as follows: "A property f having a sense for natural numbers is called a *fleeing property* if it satisfies the following three requirements:

- (i) For each natural number n , it can be decided whether or not n possesses the property f ;
- (ii) no way is known to calculate a natural number possessing f ;
- (iii) the assumption that at least one natural number possesses f , is not known to be contradictory."

A property which would be an example according to Brouwer is "the n th term of α equals 99", when α is a lawless sequence that does not yet have 99 for any of its

terms. That this property satisfies requirement (ii) is uncontroversial. Requirement (i) is considered satisfied because for a given n , simply waiting until term number n is chosen, is taken to be a decision procedure. The interesting requirement is the last one. Brouwer would claim that the mentioned property satisfies (iii) because it is possible that some future term will be 99.

This is what I want to dispute. To do that, it must first be noted that the notion of possibility which is used with the word “contradictory” is not the metaphysical one. If, for example, the two first terms of α have been chosen to be 6 and 7, he *would* consider the claim that those terms are equal to be “(known to be) contradictory”, even though it is metaphysically possible that different terms could have been chosen, specifically chosen to be both, say, 6. Instead, it is epistemic possibility; “contradictory” means “not possible relative to what is known”.

But it is *not possible relative to what is known* that “at least one natural number n possesses the property that the n th term of α equals 99”. For it is known that there are only a certain finite number of terms and that none of them equals 99. Ergo, “the n th term of α equals 99” is not a fleeing property.

And if a real number in the form of a free choice sequence of intervals is defined to be positive if it has a term with a positive left endpoint, then it is false that

$$\langle [-1, 1], [-1/2, 1/2], [-1/4, 1/4], \text{intention to expand} \rangle$$

is positive. For it is known that the sequence only has these three terms and that both -1 , $-1/2$ and $-1/4$ are negative. When Brouwer claims that this sequence does not satisfy the criterion for being positive nor the criterion for not being positive, that is not a genuine violation of bivalence. It is just a weird use of negation which is not forced upon us by the ontology of the subject matter.

What Brouwer claims are fleeing properties, are just unstable properties. They are properties that a sequence may go from having to not having or vice versa. The mentioned sequence is not positive but will be if, say, $[0, 1/4]$ is added as the 4th term and $[1/8, 1/4]$ as the 5th. And in that case it would also change from having the property of being equal to 0 to not having that property.

Definite descriptions If we look at the sentence “the 10th term of α equals 99” (assuming that less than 10 terms of α have been chosen) through Russell’s eyes, the conclusion is easily reached: The sentence is equivalent to “there exists a unique 10th term of α and anything that is a 10th term of α equals 99”, so due to the existence part, the sentence is false. And I will argue that because of the metaphysical views she is committed to, siding with e.g. Frege or Meinong instead in the famous “the present king of France is bald” controversy is not a move that is open to the Brouwerian.

Future contingents It is argued that a Brouwerian about mathematics must be a Peircian (as interpreted by Prior) about the semantics of sentences about the future: such sentences are really about what the present necessitates. So the sentence “99 will be added to α ” is to be read as “necessarily, 99 will be added to α ” which is false. And its apparent negation, “99 will not be added to α ” means “necessarily, 99 will not be added to α ” which is also false, and is so without violating any principle of classical logic.

This interpretation also gives Brouwer the desired truth value of “the 10th term of α is a natural number”. If α is intended by the creating subject to be a sequence of natural numbers then the sentence, read with an implicit necessity operator, is true by virtue of the presently existing restriction on future terms of α .

Fictional objects A likeness between fictional objects and lawless choice sequences is another thing that may induce one to claim that bivalence fails for the latter (van Atten, for one, makes this argument). For it is natural to think that it does for the former. The idea is that since Shakespeare does nowhere in *Hamlet* specify Hamlet’s height, Hamlet is an incomplete object not having a specific height, so a sentence such as “Hamlet is 5 foot 7” is neither true nor false. And it seems that the crucial feature of fictional objects which is responsible for this incompleteness is that they are creations of our minds. Since choice sequences share this feature, the analogy argument goes, they too are incomplete and defy the principle of bivalence. I argue that this analogy is misguided.

Consequences for the continuum The older Brouwer attempts, in one crucial respect, to do exactly the same as the classical mathematician, namely finding a non-denumerable totality of points with which to identify the continuum. There are still major differences in what that totality of points looks like, but the idea that the continuum can be “build up” from more basic entities is adopted.

My claim is that Brouwer’s approach fails for the following reason. The analysis of the constitution of choice sequences shows that there are not uncountably many of them. For we can not individuate them by how the entire process of choices goes. They can only be individuated by who the creating subject is, starting time and, at any given time, the created initial segment and adopted restrictions. Hence there are always just finitely many of them, i.e. the class of choice sequences is potentially, countably infinite.

I explain why the difference between the intuitionist and the classical reals consisting in the former being potential creations while the latter exist in a non-temporal way, does not help Brouwer.

References

- Brouwer, L. E. J. (1975). *Collected Works 1*. North-Holland.
- Frege, G. (1892). Über Sinn und Bedeutung. *Zeitschrift für Philosophie und philosophische Kritik* 100, 25–50. Translated as “On sense and reference” in P. Geach and M. Black (eds.) (1962), *Translations from the philosophical writings of Gottlob Frege*. Basil Blackwell.
- Meinong, A. (1904). Über gegenstandstheorie. In *Untersuchungen zur Gegenstandstheorie und Psychologie*. J. A. Barth.
- Prior, A. (1967). *Past, Present and Future*. Oxford University Press.
- Russell, B. (1905). On denoting. *Mind, New Series* 14, 479–493.
- Thomasson, A. L. (1999). *Fiction and Metaphysics*. Cambridge University Press.
- van Atten, M. (2007). *Brouwer Meets Husserl*. Springer.
- van Stigt, W. P. (1990). *Brouwer’s Intuitionism*. North-Holland.

Weak Models of Distributed Computing and Modal Logic

Lauri Hella

School of Information Sciences,
University of Tampere

This presentation is based on the joint paper [3] with *Matti Järvisalo, Antti Kuusisto, Juhana Laurinharju, Tuomo Lempinen, Kerkko Luosto, Jukka Suomela, and Jonni Virtama*, that is published in Proceedings of PODC 2012.

Distributed Algorithms

We present a classification of weak models of distributed computing that are obtained by natural restrictions on the widely-studied port-numbering model. A distributed algorithm is best understood as a state machine \mathcal{A} . In a distributed system, each node of an input graph $G = (V, E)$ is a copy of the same machine \mathcal{A} . Computation proceeds in synchronous steps. In each step, each machine

- (1) sends messages to its neighbours,
- (2) receives messages from its neighbours, and
- (3) updates its state based on the messages that it received.

If the new state on node $v \in V$ is a stopping state s_v , the machine halts. If the machine halts on all nodes of G , the output of the algorithm is the function $S: V \rightarrow Y$ defined by $S(v) = s_v$, where Y is the set of stopping states of \mathcal{A} .

In the port-numbering model, the input ports and output ports of each node of degree d are numbered with $1, 2, \dots, d$. These numbers are given by a *port numbering* p of the input graph G . A port numbering is *consistent*, if the input port and the output port with same number is always connected to the same neighbour. The output of an algorithm may depend on the port numbering.

A *graph problem* is a function Π that associates with each undirected graph $G = (V, E)$ a set $\Pi(G)$ of *solutions*. Each solution $S \in \Pi(G)$ is a mapping $S: V \rightarrow Y$, where Y is a finite set. Let T be a function $\mathbb{N} \rightarrow \mathbb{N}$. We say that algorithm \mathcal{A} *solves* Π *in time* T if the following hold for any graph G , and any port numbering p of G :

- (a) Algorithm \mathcal{A} stops in time $T(|V|)$ in (G, p) .
- (b) The output of \mathcal{A} is in $\Pi(G)$.

Let $\mathcal{F}(\Delta)$ be the class of all graphs $G = (V, E)$ such that $\deg(v) \leq \Delta$ for all $v \in V$. We say that \mathcal{A} *solves* Π *in time* T *on* $\mathcal{F}(\Delta)$, if the conditions above hold for all $G \in \mathcal{F}(\Delta)$.

Let \mathbf{VV}_c be the class of all graph problems that can be solved in the port-numbering model assuming consistency. We define the following subclasses of \mathbf{VV}_c :

- VV**: Input port i and output port i are not necessarily connected to the same neighbour.
- MV**: Input ports are not numbered; algorithms receive a multiset of messages.
- SV**: Input ports are not numbered; algorithms receive a set of messages.
- VB**: Output ports are not numbered; algorithms send the same message to all output ports.
- MB**: Combination of MV and VB.
- SB**: Combination of SV and VB.

Furthermore, we let $\mathbf{VV}_c(1)$, $\mathbf{VV}(1)$, $\mathbf{MV}(1)$, $\mathbf{SV}(1)$, $\mathbf{VB}(1)$, $\mathbf{MB}(1)$ and $\mathbf{SB}(1)$ be the constant-time versions of these classes.

Capturing Constant Time Classes by Modal Logics

Each of the constant time classes defined above can be characterized by a corresponding version of modal logic, in the spirit of descriptive complexity theory. The modal logics used in the characterization are *basic modal logic* ML, *graded modal logic* GML, *multimodal logic* MML, and *graded multimodal logic* GMML (see [1], [2]).

There is a natural correspondence between the distributed algorithms in the port numbering model and the logics ML, GML, MML, and GMML. For any input graph G and port numbering p of G , the pair (G, p) can be transformed into a Kripke model $K(G, p) = (W, (R_\alpha)_{\alpha \in I}, \tau)$ in a canonical way. Given a local algorithm \mathcal{A} , its execution can then be simulated by a modal formula φ . The crucial idea is that the truth condition for a diamond formula $\langle \alpha \rangle \psi$ is interpreted as communication between the nodes:

$$K, v \models \langle \alpha \rangle \psi \iff v \text{ receives the message ``}\psi \text{ is true'' from some } u \text{ s.t. } (v, u) \in R_\alpha.$$

Conversely, given a modal formula φ , the evaluation of its truth in the Kripke model $K(G, p)$ can be done by a local algorithm \mathcal{A} .

There are in fact four different versions of $K(G, p)$, reflecting the fact that algorithms in the lower classes do not use all the information encoded in the port numbering. Let $G = (V, E) \in \mathcal{F}(\Delta)$, and let p be a port numbering of G . The accessibility relations used in $K_1(G, p)$ are the following:

$$R_{(i,j)} = \{(u, v) \in V \times V : p \text{ maps output port } j \text{ of } v \text{ to port } i \text{ of } u\} \text{ for } 1 \leq i, j \leq \Delta.$$

The accessibility relations in $K_2(G, p)$ and $K_3(G, p)$ are:

$$R_{(i,*)} = \bigcup_{j \in [\Delta]} R_{(i,j)} \quad \text{and} \quad R_{(*,i)} = \bigcup_{j \in [\Delta]} R_{(j,i)} \quad \text{for each } 1 \leq i \leq \Delta.$$

Finally, $K_4(G, p)$ has only one accessibility relation $R_{(*,*)} = \bigcup_{1 \leq i, j \leq \Delta} R_{(i,j)}$.

For each $i \in \{1, 2, 3, 4\}$ and $G \in \mathcal{F}(\Delta)$, we denote the class of all Kripke models of the form $K_i(G, p)$ by $\mathcal{K}_i(\Delta)$. Furthermore, we denote by $\mathcal{K}_0(\Delta)$ the subclass of $\mathcal{K}_1(\Delta)$ consisting of the models $K_1(G, p)$, where p is a consistent port numbering of G .

Let $i \in \{1, 2, 3, 4\}$, and let ψ be a modal formula. Then ψ *defines a solution* for a graph problem Π on the class $\mathcal{K}_i(\Delta)$ if

- for all $G \in \mathcal{F}(\Delta)$, and all port numberings p of G , the subset $\|\psi\|^{K_i(G,p)}$ defined by the formula ψ in the model $K_i(G, p)$ is in set $\Pi(G)$.

Furthermore, ψ defines a solution for Π on the class $\mathcal{K}_0(\Delta)$, if the condition above with $i = 1$ holds for all *consistent* port numberings p . Note that any modal formula ψ gives rise to a canonical graph problem Π_ψ that it solves: for each $G \in \mathcal{F}(\Delta)$, the solution set $\Pi_\psi(G)$ consists of the sets $\|\psi\|^{K_i(G,p)}$ where p ranges over the (consistent) port numberings of G .

Let \mathcal{L} be a modal logic, let $i \leq 4$, and let C be a class of graph problems. We say that \mathcal{L} *captures* C on $\mathcal{K}_i(\Delta)$ if the following two conditions hold:

- If ψ is an \mathcal{L} -formula in the vocabulary of $\mathcal{K}_i(\Delta)$, then $\Pi_\psi \in C$.
- For every graph problem $\Pi \in C$ there is an \mathcal{L} -formula ψ in the vocabulary of $\mathcal{K}_i(\Delta)$, which defines a solution for Π on $\mathcal{K}_i(\Delta)$.

Furthermore, we say that \mathcal{L} captures C on \mathcal{K}_i , if it captures C on $\mathcal{K}_i(\Delta)$ for all $\Delta \in \mathbb{N}$.

Theorem.

- | | |
|--|--|
| (a) MML captures $\mathbf{VV}_c(1)$ on \mathcal{K}_0 . | (b) MML captures $\mathbf{VV}(1)$ on \mathcal{K}_1 . |
| (c) GMML captures $\mathbf{MV}(1)$ on \mathcal{K}_2 . | (d) MML captures $\mathbf{SV}(1)$ on \mathcal{K}_2 . |
| (e) MML captures $\mathbf{VB}(1)$ on \mathcal{K}_3 . | (f) GML captures $\mathbf{MB}(1)$ on \mathcal{K}_4 . |
| (g) ML captures $\mathbf{SB}(1)$ on \mathcal{K}_4 . | |

Containments between the Classes

There are many trivial containment relations, such as $SB \subseteq MB \subseteq VB \subseteq VV \subseteq VV_c$, but it is not obvious if, e.g., either of $VB \subseteq SV$ or $SV \subseteq VB$ should hold. Nevertheless, it turns out that we can identify a *linear order* on these classes. Indeed, we prove that $SB \subsetneq MB = VB \subsetneq SV = MV = VV \subsetneq VV_c$. The same holds for the constant-time versions of these classes.

All the three separations between the classes can be proved by using the correspondence to modal logics. More precisely, we use bisimulations to show that a suitable graph problem is not definable in the modal logic corresponding to the weaker class, while it is easily seen to be in the stronger class.

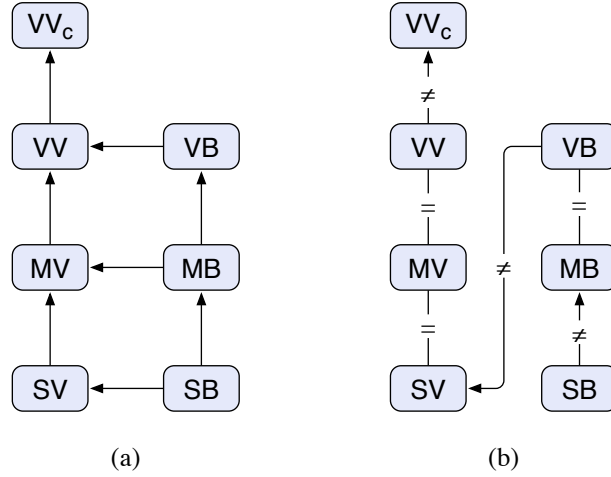


Figure 1: Classes of graph problems. (a) Trivial subset relations between the classes. (b) The linear order identified in this work.

References

- [1] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, UK, 2001.
- [2] Patrick Blackburn, Johan van Benthem, and Frank Wolter, editors. *Handbook of Modal Logic*, volume 3 of *Studies in Logic and Practical Reasoning*. Elsevier, Amsterdam, The Netherlands, 2007.
- [3] L. Hella, M. Järvisalo, A. Kuusisto, J. Laurinharju, T. Lempinen, K. Luosto, J. Suomela, and J. Virtama. Weak models of distributed computing, with connections to modal logic. To appear in *Proc. 31st Annual ACM Symposium on Principles of Distributed Computing (PODC, Madeira, Portugal, July 2012)*. Full version available in arXiv:1205.2051 [cs.DC].

Dynamic Condition Response Graphs - A Dynamic Temporal Logic for Event-based Processes

Thomas Hildebrandt¹ *

IT University of Copenhagen, Rued Langgaardsvej 7, 2300 Copenhagen, Denmark
hilde@itu.dk

The Dynamic Condition Response Graphs (DCR Graphs) [4, 6] process logic has been developed in the Trustworthy Pervasive Healthcare Services (TrustCare) [1] research project as a declarative formal foundation for event-based, adaptable and flexible pervasive workflow processes and services. The DCR Graphs model is inspired by our industrial partner's declarative workflow model [5, 8] and generalizes the classical labelled event structure model [12] to a so-called systems model, allowing for *finite* descriptions of infinite behavior and to distinguish between may (i.e. possible) and must (i.e. required) behavior.

A DCR Graph as defined in [4] is a directed graph described by an 8-tuple $(E, M, \rightarrow\bullet, \bullet\rightarrow, \rightarrow+, \rightarrow\%, L, l)$. The nodes of the graph are given by the set E of *events*, labelled by the *labeling function* $l : E \rightarrow L$. The linear time semantics of a DCR Graph is a subset of finite and infinite sequences of events satisfying the constraints defined by the edges of the graph. The edges are given by four relations between events: The *condition* ($\rightarrow\bullet$), *response* ($\bullet\rightarrow$), *include* ($\rightarrow+$), and *exclude* ($\rightarrow\%$) relation respectively. A key new ingredient of DCR Graphs is that the semantics is defined relative to a *marking* M , defined as a triple of three sets of events (Ex, Re, In) . The set $Ex \subseteq E$ is the events that have happened in the past. The set $Re \subseteq E$ are events that are required to happen or be excluded (as explained next) in the future in order for a sequence to be accepting. Finally, the set $In \subseteq E$ is the set of (currently) *included* events.

The formal definition of DCR Graphs and their semantics are given in Def. 1 below. We employ the following notation. For a set E we write $\mathcal{P}(E)$ for the power set of E (i.e. set of all subsets of E), E^* for the set of all finite sequences of elements of E , $^\omega$ for the set of all infinite sequences of elements of E and $E^\infty = E^* \cup E^\omega$. We write ϵ the empty sequence. For a binary relation $\rightarrow \subseteq E \times E$ and a subset $\xi \subseteq E$ of E we write $\rightarrow \xi$ and $\xi \rightarrow$ for the set $\{e \in E \mid (\exists e' \in \xi \mid e \rightarrow e')\}$ and the set $\{e \in E \mid (\exists e' \in \xi \mid e' \rightarrow e)\}$ respectively, and abuse notation writing $\rightarrow e$ and $e \rightarrow$ for $\rightarrow \{e\}$ and $\{e\} \rightarrow$ respectively when $e \in E$.

Definition 1. A Dynamic Condition Response Graph (DCR Graph) G is a tuple $(E, M, \rightarrow\bullet, \bullet\rightarrow, \rightarrow+, \rightarrow\%, L, l)$, where

- (i) E is a set of events (or activities),
- (ii) $M = (Ex_G, Re_G, In_G) \in \mathcal{P}(E) \times \mathcal{P}(E) \times \mathcal{P}(E)$ is the marking
- (iii) $\rightarrow\bullet, \bullet\rightarrow, \rightarrow+, \rightarrow\% \subseteq E \times E$ is the condition, response, include and exclude relation respectively.

* This research is supported by the Danish Research Agency through the Trustworthy Pervasive Healthcare Services project (grant #2106-07-0019, www.trustcare.eu)

(iv) L is the set of labels and $l : E \rightarrow L$ is a labeling function mapping events to a label.

We define that an event $e \in E$ is enabled, written $G \vdash e$, if $e \in \text{In}_G \wedge (\text{In}_G \cap \rightarrow \bullet e) \subseteq \text{Ex}_G$. We define the result on the (marking) of the graph G if an event e happen as $G \oplus e =_{\text{def}} (E, M', \rightarrow \bullet, \bullet \rightarrow, \rightarrow +, \rightarrow \%, L, l)$, where $M' = (\text{Ex} \cup \{e\}, (\text{Re} \setminus \{e\}) \cup e \bullet \rightarrow, (\text{In}_G \setminus e \rightarrow \% \cup e \rightarrow +))$. This is extended inductively to finite sequences $\sigma \in E^*$ of events by $G \oplus \sigma = (G \oplus e) \oplus \sigma'$, if $\sigma = e\sigma'$ $G \oplus \epsilon = G$.

An event $e \in E$ can happen (is enabled) if it is included ($e \in \text{In}_G$) and all its included conditions have happened in the past ($\text{In}_G \cap \rightarrow \bullet e \subseteq \text{Ex}_G$). The set of included events changes dynamically according to the include and exclude relations of the DCR Graph, which generalizes the (binary, symmetric and irreflexive) conflict relation of event structures. If an event happens it results in a new marking of the graph, where the included set is given by $\text{In}_G \setminus e \rightarrow \% \cup e \rightarrow +$, that is, the events excluded by e are first removed from In and then the events included by e are added. The set Ex_G of events that have happened in the past is simply extended with e . The set Re_G of events required to happen in the future (or be excluded) is updated by first removing the event e that just happened, and then adding all response events for e , i.e. events required to happen in the future whenever e happens.

The definition of when an event is enabled and the result of executing it is used to define the set of sequences accepted by a DCR Graph G below. The first condition states that any intermediate event must be enabled in the graph resulting from executing the sequence of events leading to that event. The second condition captures the response constraint and states that any intermediate, included response event must eventually happen or be excluded.

Definition 2. We define a (finite or infinite) sequence $\sigma \in E^\infty$ of events to be accepted by G , written $\sigma \models G$, if

1. $\sigma = \sigma' e \sigma''$ implies $G \oplus \sigma' \vdash e$ and
2. $\sigma = \sigma' \sigma''$ for $\sigma' \in E^*$ and $G' = G \oplus \sigma'$ implies $\forall e \in \text{Re}_{G'} \cap \text{In}_{G'}. \exists \sigma''' \in E^*. \sigma'' = \sigma''' e' \sigma'''' \wedge (e' = e \vee e \notin \text{In}_{G' \oplus \sigma''' e'})$.

We refer to the set of (finite and infinite) sequences of labels that label sequences of events accepted by G to as the language of G .

It is shown in [7] that the language of a DCR Graph can be characterized by a Büchi automaton over $L \cup \{\tau\}$. Conversely, it is shown in [6] that any Büchi-automaton can be encoded as a DCR Graph accepting the same language. Thus, the linear time semantics of DCR Graphs characterizes exactly the ω -regular languages and thus in particular all processes that can be specified in Linear-time Temporal Logic (LTL).

Note that the definition of enabledness and execution also shows how a DCR Graph can be used as an execution model, which is exploited by our industrial partners using variants of DCR Graphs as both specification and execution models for workflow and case management systems. Basically, every enabled event is offered as a possible next activity of the workflow process, while every included response event is indicated as an event that must be carried at some point in the future. This provides a much more

direct correspondence between the DCR Graph as specification and its execution than approaches based on LTL as specification language and an automata representation of LTL as operational semantics [9–11].

In the talk we will present the DCR Graph model, its relation to event structures, LTL and Büchi-automata, and its applications as described in [6], in particular techniques for distribution by projections [2, 3] and extensions with timed constraints and partial order semantics. The work is carried out jointly with Raghava Rao Mukkamala and Tijs Slaats.

References

1. Thomas Hildebrandt. Trustworthy pervasive healthcare processes (TrustCare) research project. Webpage, 2008. <http://www.trustcare.dk/>.
2. Thomas Hildebrandt, Raghava Rao Mukkamala, and Tijs Slaats. Declarative modelling and safe distribution of healthcare workflows. In *International Symposium on Foundations of Health Information Engineering and Systems*, Johannesburg, South Africa, August 2011.
3. Thomas Hildebrandt, Raghava Rao Mukkamala, and Tijs Slaats. Safe distribution of declarative processes. In *9th International Conference on Software Engineering and Formal Methods (SEFM) 2011*, 2011.
4. Thomas T. Hildebrandt and Raghava Rao Mukkamala. Declarative event-based workflow as distributed dynamic condition response graphs. In *PLACES*, volume 69 of *EPTCS*, pages 59–73, 2010.
5. Karen Marie Lyng, Thomas Hildebrandt, and Raghava Rao Mukkamala. From paper based clinical practice guidelines to declarative workflow management. In *Proceedings of 2nd International Workshop on Process-oriented information systems in healthcare (ProHealth 08)*, pages 36–43, Milan, Italy, 2008. BPM 2008 Workshops.
6. Raghava Rao Mukkamala. *A Formal Model For Declarative Workflows - Dynamic Condition Response Graphs*. PhD thesis, IT University of Copenhagen, March 2012. Forthcomming.
7. Raghava Rao Mukkamala and Thomas Hildebrandt. From dynamic condition response structures to Büchi automata. In *Proceedings of 4th IEEE International Symposium on Theoretical Aspects of Software Engineering (TASE 2010)*, August 2010.
8. Raghava Rao Mukkamala, Thomas Hildebrandt, and Janus Boris Tøth. The resultmaker online consultant: From declarative workflow management in practice to LTL. In *Proceeding of DDBP*, 2008.
9. Wil van der Aalst, Maja Pesic, Helen Schonenberg, Michael Westergaard, and Fabrizio M. Maggi. Declare. Webpage, 2010. <http://www.win.tue.nl/declare/>.
10. Wil M.P van der Aalst and Maja Pesic. A declarative approach for flexible business processes management. In *Proceedings DPM 2006*, LNCS. Springer Verlag, 2006.
11. Wil M.P van der Aalst and Maja Pesic. DecSerFlow: Towards a truly declarative service flow language. In M. Bravetti, M. Nunez, and Gianluigi Zavattaro, editors, *Proceedings of Web Services and Formal Methods (WS-FM 2006)*, volume 4184 of *LNCS*, pages 1–23. Springer Verlag, 2006.
12. Glynn Winskel. Event structures. In Wilfried Brauer, Wolfgang Reisig, and Grzegorz (Eds.) Rozenberg, editors, *Advances in Petri Nets*, volume Vol. 255 of *Lecture Notes in Computer Science*, pages 325–392. Springer, 1986.

GENERALIZED ISOMORPHISMS IN METRIC MODEL THEORY

ÅSA HIRVONEN

Department of Mathematics and Statistics, University of Helsinki

Metric model theory studies structures consisting basically of a metric space as domain and continuous functions on it. When studying these one is often interested in mappings between the structures that are not genuine isomorphisms but are allowed to make small errors on part of the vocabulary. Examples include linear isomorphisms of Banach spaces or approximations of operators on Hilbert spaces. These can be considered a form of generalized isomorphisms, as we would like to give them the role of isomorphisms in classical model theory.

An essential gain from generalizing the concept of isomorphism is that this improves the degree of stability of the class under consideration. Classes that are non-superstable may become ω -stable when small changes are allowed. Ben Yaacov first created a formalism for these changes when he introduced his notion of *perturbation systems* in [BY08]. To better be able to use this improvement in stability, Hyttinen and I [HH12] introduced an abstract framework allowing for the built-in treatment of perturbations. The framework is syntax-free, based on Shelah's *abstract elementary classes*, and generalizes earlier approaches to metric model theory. The generalized isomorphisms appear as classes of ε -isomorphisms specifying how the structures may be perturbed. In this setting we have built splitting theory and constructible models with respect to the generalized isomorphisms, giving rise to a nicely-behaved independence notion and a dominance theorem.

In the talk I will illustrate by examples what generalized isomorphisms are and what properties are needed of them in our constructions. I will also illustrate the main techniques used so far and shed some light on further directions and challenges.

The talk is based on joint work with Tapani Hyttinen.

REFERENCES

- [BY08] Itai Ben Yaacov. On perturbations of continuous structures. *J. Math. Log.*, 8(2):225–249, 2008.
- [HH12] Åsa Hirvonen and Tapani Hyttinen. Metric abstract elementary classes with perturbations. *Fund. Math.*, 217:123–170, 2012.

Analysis and Synthesis of Algorithms in Theorema

EXTENDED ABSTRACT

Tudor Jebelean

Research Institute for Symbolic Computation
Johannes Kepler University
Altenbergerstraße 69, A-4040, Linz, Austria
`Tudor.Jebelean@jku.at`

Introduction. We present a short overview of the *Theorema* system and of our research concerning the verification of programs and of proof-based algorithm synthesis.

Theorema aims at supporting the logic and computational activities involved in algorithm design, including modelling, proving, and implementing. In this framework we study the theory of programming: On one hand we design verification condition generators for algorithms expressed as programs, and on the other hand we develop principles and proving techniques for algorithm synthesis.

The syntax of programs is based on the logical syntax of the underlying theory of the objects manipulated by the program. Their semantics is defined by translation into predicate logic. The verification conditions are generated through path sensitive symbolic execution, and the termination condition is expressed as an induction principle. We show that for programs obeying safety verification conditions, the functional verification conditions are necessary and sufficient for partial correctness. Furthermore we show that together with the termination condition they ensure the logico-mathematical existence of the function implemented by the program.

For algorithm synthesis, we first prove the existence of the value of the desired function, and from it we extract the terms which are necessary for the construction of the algorithm. We synthesize several sorting algorithms by using various induction principles and some novel proof techniques developed for the domain of lists.

Theorema. The *Theorema* system (www.theorema.org) aims at supporting all phases of the algorithmic problem solving cycle: developing mathematical models, proving conjectures, implementing algorithms and experimenting with computations. The system is based on the computer algebra system Mathematica, from which it makes extensive use of the rewrite-based programming style, as well as of some of the library functions for symbolic and numeric computing.

The main characteristics of the system are: input and output of mathematical formulae in natural (two dimensional) notation, natural style proving (similar to human style) and the use of the same language framework (predicate logic) both for constructing mathematical theories as well as for constructing algorithms.

The *Theorema* system started as a collection of provers for propositional and for predicate logic [3] and then it was enhanced with various domain specific

provers [2] and with a computation environment which completes its theory exploration capabilities [1]. Over the years various specific domains have been investigated by constructing their theories and by developing special proving methods for them [8]. A special flavour of the system is the combination of logical methods with techniques from computer algebra [12] and from algebraic combinatorics [9].

Program Analysis. We approach the analysis of algorithms expressed as programs using a specific logical foundation. As a starting principle, we assume that the terms and the formulae occurring in programs are composed using the signature of an object theory – the theory of the objects manipulated by the programs.

Functional programs are assimilated with conjunctions of (conditional) equalities, thus they constitute just logical formulae. Therefore a definition of program semantics is not necessary, because it is just the semantics of predicate logic. Every program consists in a function definition and it must be associated to a specification (input and output conditions). We require that each program is "*coherent*": the arguments of each function call must satisfy the input condition of that function. (This is also called "safety" by other researchers.) For coherent programs we show that the functional correctness conditions are necessary and sufficient for partial correctness [10,11].

Moreover we generate a necessary and sufficient termination condition in form of an induction principle. Using this approach we can prove that the total correctness ensures the existence and the uniqueness of the function implemented by the program. Thus mathematical logical correctness of function definitions is equivalent to computational program correctness [6]. This proof is now completely formalized in *Theorema*, and it constitutes a very interesting exercise in the investigation of the logical foundations of program verification. Namely, by formalizing this proof in an automated reasoning system, one can detect what is the minimal set of axioms and of inference rules which are necessary in order to construct a sufficiently expressive theory of programs.

Imperative programs are expressed as meta-terms using only few constructs for statement sequences and for elementary statements (assignments, conditionals, return, and optionally loops). Also here we keep the previously mentioned approach of using formulae and terms from the object theory in the program text. The semantics is defined by translation into functional programs using path sensitive symbolic execution [7], thus the analysis of imperative programs reduces to the analysis of functional programs.

Algorithm Synthesis. Even more challenging is the problem of synthesizing programs starting from their specification. We approach this problem in a proof-based manner: using various induction principles, one proves the existence of the output value of the specified function, and from this proof one extracts the algorithm. The most difficult part is the construction of the proof. We illustrate various proof techniques for the theory of tuples (lists) which can be used for the synthesis of sorting algorithms [4]. These techniques include: Prolog style back-chaining, normal forms with respect to the relation of equivalence on tuples (a

tuple is the permutation of another), and decomposition of atoms expressing inequalities over elements and tuples. Finally we construct a systematic method for finding a sorted list equivalent to a given list expression. A key ingredient of our synthesis method is the cascading of inductive proofs: when an inductive case of a proof does not succeed in first order logic because the absence of a required function, then this function is synthesized again by another inductive proof. The new function has the same output specification as the old one, but the input specification is more complex, and thus the proof will be easier. For instance, in the case of the synthesis of the merge-sort algorithm, few steps of cascading will reduce the problem to the elementary functions on lists (*head*, *tail* and the list-like *cons*) [5].

References

1. B. Buchberger, A. Craciun, T. Jebelean, L. Kovacs, T. Kutsia, K. Nakagawa, F. Piroi, N. Popov, J. Robu, M. Rosenkranz, and W. Windsteiger, *Theorema: Towards Computer-Aided Mathematical Theory Exploration*, Journal of Applied Logic **4** (2006), no. 4, 470–504.
2. B. Buchberger, C. Dupre, T. Jebelean, F. Kriftner, K. Nakagawa, D. Vasaru, and W. Windsteiger, *The Theorema Project: A Progress Report*, Calculemus 2000: Integration of Symbolic Computation and Mechanized Reasoning, 2000.
3. B. Buchberger, T. Jebelean, F. Kriftner, M. Marin, E. Tomuta, and D. Vasaru, *A Survey of the Theorema project*, Proceedings of ISSAC’97 (W. Kuechlin, ed.), ACM Press, 1997, pp. 384–391.
4. I. Dramnescu and T. Jebelean, *Proof Techniques for Synthesis of Sorting Algorithms*, Proceedings of SYNASC 2011 (D. Wang et al., ed.), IEEE Computer Society, 2011, pp. 101–109.
5. ———, *Theory Exploration in Theorema: Case Study on Lists*, Proceedings of the IEEE 7th International Symposium on Applied Computational Intelligence and Informatics, IEEE Xplore, 2012.
6. M. Erascu and T. Jebelean, *A Purely Logical Approach to Program Termination*, Proceedings of the 11th International Workshop on Termination, FLOC 2010, 2010.
7. ———, *A Purely Logical Approach to the Termination of Imperative Loops*, Proceedings of SYNASC 2010 (T. Ida et al., ed.), IEEE Computer Society, 2010.
8. T. Jebelean et. al, *Automated Reasoning*, Hagenberg Research (B. Buchberger et al., ed.), Springer, 2009, pp. 63–101.
9. L. Kovacs and T. Jebelean, *Finding Polynomial Invariants for Imperative Loops in the Theorema System*, Proc. of Verify’06 Workshop, IJCAR’06, FLOC 2006 (S. Autexier and H. Mantel, eds.), 2006, pp. 52–67.
10. L. Kovacs, N. Popov, and T. Jebelean, *Verification Environment in Theorema*, Annals of Mathematics, Computing and Teleinformatics **1** (2005), no. 2, 27–34.
11. N. Popov and T. Jebelean, *Sound and Complete Verification Condition Generator for Functional Recursive Programs.*, Numerical and Symbolic Scientific Computing: Progress and Prospects (U. Langer and P. Paule, eds.), Springer, 2011, pp. 219–256.
12. R. Vajda, T. Jebelean, and B. Buchberger, *Combining Logical and Algebraic Techniques for Natural Style Proving in Elementary Analysis*, Mathematics and Computers in Simulation **79** (2009), no. 8, 2310–2316.

Paraconsistent Computational Logic

Andreas Schmidt Jensen and Jørgen Villadsen

Algorithms and Logic Section, DTU Informatics, Denmark

Abstract. In classical logic everything follows from inconsistency and this makes classical logic problematic in areas of computer science where contradictions seem unavoidable. We describe a many-valued paraconsistent logic, discuss the truth tables and include a small case study.

1 Introduction — Motivation and Definitions

Often consistency cannot be assumed and a paraconsistent logic seems needed, in particular for applications of logic in computer science and artificial intelligence [1, 2]. We consider the propositional fragment of a higher-order paraconsistent logic [4, 5]. We have the two classical determinate truth values $\Delta = \{\bullet, \circ\}$ for truth and falsity and a countably infinite set of indeterminate truth values $\nabla = \{I, II, III, \dots\}$. The only designated truth value \bullet yields the logical truths. The indeterminate truth values are not at all ordered with respect to truth content. The logic is a generalization of Łukasiewicz’s three-valued logic (originally proposed 1920–30), with the intermediate value duplicated many times and ordered such that none of the copies of this value imply other ones, but it differs from Łukasiewicz’s many-valued logics as well as from logics based on bilattices [3]. The motivation for the logical operators is based on key equalities shown to the right of the semantic clauses (we also have $\varphi \Leftrightarrow \neg\neg\varphi$):

$$\begin{aligned} \llbracket \neg\varphi \rrbracket &= \begin{cases} \bullet & \text{if } \llbracket \varphi \rrbracket = \circ \\ \circ & \text{if } \llbracket \varphi \rrbracket = \bullet \\ \llbracket \varphi \rrbracket & \text{otherwise} \end{cases} & \begin{array}{l} \top \Leftrightarrow \neg\perp \\ \perp \Leftrightarrow \neg\top \end{array} \\ \llbracket \varphi \wedge \psi \rrbracket &= \begin{cases} \llbracket \varphi \rrbracket & \text{if } \llbracket \varphi \rrbracket = \llbracket \psi \rrbracket \\ \llbracket \psi \rrbracket & \text{if } \llbracket \varphi \rrbracket = \bullet \\ \llbracket \varphi \rrbracket & \text{if } \llbracket \psi \rrbracket = \bullet \\ \circ & \text{otherwise} \end{cases} & \begin{array}{l} \varphi \Leftrightarrow \varphi \wedge \varphi \\ \psi \Leftrightarrow \top \wedge \psi \\ \varphi \Leftrightarrow \varphi \wedge \top \end{array} \end{aligned}$$

The basic semantic clause and the clause $\llbracket \top \rrbracket = \bullet$ are omitted. In the semantic clauses several cases may apply if and only if they agree on the result. Note that the semantic clauses work for classical logic too.

Abbreviations:

$$\perp \equiv \neg\top \quad \varphi \vee \psi \equiv \neg(\neg\varphi \wedge \neg\psi)$$

We continue with biimplication (and we then simply obtain implication and modality as abbreviations). The semantic clauses extend the clauses for equality $=$ which always give a determinate truth value:

$$\begin{aligned} \llbracket \varphi \Leftrightarrow \psi \rrbracket &= \begin{cases} \bullet & \text{if } \llbracket \varphi \rrbracket = \llbracket \psi \rrbracket \\ \circ & \text{otherwise} \end{cases} \\ \llbracket \varphi \leftrightarrow \psi \rrbracket &= \begin{cases} \bullet & \text{if } \llbracket \varphi \rrbracket = \llbracket \psi \rrbracket \\ \llbracket \psi \rrbracket & \text{if } \llbracket \varphi \rrbracket = \bullet \\ \llbracket \varphi \rrbracket & \text{if } \llbracket \psi \rrbracket = \bullet \\ \llbracket \neg\psi \rrbracket & \text{if } \llbracket \varphi \rrbracket = \circ \\ \llbracket \neg\varphi \rrbracket & \text{if } \llbracket \psi \rrbracket = \circ \\ \circ & \text{otherwise} \end{cases} & \begin{array}{l} \top \Leftrightarrow \varphi \leftrightarrow \varphi \\ \psi \Leftrightarrow \top \leftrightarrow \psi \\ \varphi \Leftrightarrow \varphi \leftrightarrow \top \\ \neg\psi \Leftrightarrow \perp \leftrightarrow \psi \\ \neg\varphi \Leftrightarrow \varphi \leftrightarrow \perp \end{array} \end{aligned}$$

Abbreviations:

$$\varphi \Rightarrow \psi \equiv \varphi \Leftrightarrow \varphi \wedge \psi \quad \varphi \rightarrow \psi \equiv \varphi \leftrightarrow \varphi \wedge \psi \quad \Box\varphi \equiv \varphi = \top \quad \sim\varphi \equiv \neg\Box\varphi$$

2 Discussion of Truth Tables

Although we have a countably infinite set of truth value we can investigate the logic by truth tables since the indeterminate truth values are not ordered with respect to truth content.

\wedge	●	○			\vee	●	○			\neg	
●	●	○			●	●	●	●	●	●	○
○	○	○	○	○	○	●	○			○	●
		○		○		●			●		
		○	○			●		●			
\Leftrightarrow	●	○			\Rightarrow	●	○			\Box	
●	●	○	○	○	●	●	○	○	○	●	●
○	○	●	○	○	○	●	●	●	●	○	○
	○	○	●	○		●	○	●	○		○
	○	○	○	●		●	○	○	●		○
\leftrightarrow	●	○			\rightarrow	●	○			\sim	
●	●	○			●	●	○			●	○
○	○	●			○	●	●	●	●	○	●
			●	○		●		●			●
			○	●		●			●		●

The required number of indeterminacies corresponds to the number of propositions in a given formula. This also means that the logic is weakened when additional indeterminate truth values are added.

Given an atomic formula, it is clear that $\nabla = \{|\}$ suffices. To see this we use the fact that there exist no ordering between indeterminate truth values. If $\llbracket P \rrbracket = ||$ and we replace the truth value with $|$ then the truth value is still indeterminate. Now consider the tautology $P \rightarrow Q \leftrightarrow \neg Q \rightarrow \neg P$ (contraposition). Using a single indeterminacy yields no difference; the formula still holds. When using two indeterminacies we can give a counter-example:

$$P \rightarrow Q \leftrightarrow \neg Q \rightarrow \neg P$$

$$| \quad | \quad || \quad \circ \quad || \quad || \quad || \quad | \quad |$$

We could require $\nabla = \{|\,||\,|||\}$, but the third indeterminacy is not needed since we already have one unique indeterminacy for each proposition.

As an example for three propositions, we can consider the formula $\neg(Q \rightarrow P) \rightarrow (\neg(\neg R \vee (R \rightarrow Q)) \rightarrow (P \rightarrow Q))$. It is a tautology when $\nabla = \emptyset$ (classical propositional logic), $\nabla = \{|\}$ and $\nabla = \{|\,||\}$. When $\nabla = \{|\,||\,|||\}$ it is no longer a tautology:

$$\neg (Q \rightarrow P) \rightarrow (\neg (\neg R \vee (R \rightarrow Q)) \rightarrow (P \rightarrow Q))$$

$$| \quad | \quad | \quad || \quad | \quad ||| \quad ||| \quad ||| \quad ||| \quad ||| \quad ||| \quad | \quad ||| \quad || \quad || \quad |$$

3 Case Study

Paraconsistent logics are useful in areas where inconsistency is an acceptable feature of the systems involved. One such area is multi-agent systems since the belief base of an agent very well could contain contradictory beliefs and thus be inconsistent. Consider an agent with a set of beliefs (item 0) and rules:

0. $P \wedge Q \wedge \neg R$
1. $P \wedge Q \rightarrow R$
2. $R \rightarrow S$
3. $\neg S \rightarrow \neg R$

We can deduce R using classical logic. This leaves the agent with contradictory beliefs, namely R and $\neg R$. This entails everything, so the agent might start behaving in an undesirable way. It could now believe that $\neg P$, or $\neg Q$ – or even φ for any formula φ .

In our paraconsistent logic this is not the case. The following is not a tautology (the truth value of the main connective is 1):

$$(P \wedge Q \wedge \neg R) \wedge (P \wedge Q \rightarrow R) \rightarrow R$$

● | | | ● ○ | ● | | | ○ | ○

Note that while rule 2 and 3 are classically equivalent (contraposition), it is not the case in our paraconsistent logic. This means that even if R follows, this does not necessarily mean that S follows as well.

We do however need some kind of modus ponens in order for the agent to be able to reason. We therefore require that rules are not allowed to be inconsistent and use the necessity operator to ensure either truth or falsity. This makes sense; after all, the agent requires absolute knowledge about whether its rules are applicable or not.

$$(P \wedge Q \wedge \neg R) \wedge \Box (P \wedge Q \rightarrow R) \Rightarrow R$$

● | | | ● ○ ○ ○ ● | | | ○ ● ○

Note that applying the necessity operator on rules does not make the agent classical. $\neg P$ and $\neg Q$ still does not follow. We let $\triangleright_{XYZ} P$ mean that P follows from the agents rules X , Y and Z .

$$(P \wedge Q \wedge \neg R) \wedge \Box (P \wedge Q \rightarrow R) \Rightarrow R \equiv \triangleright_{01} R$$

The agent can then conclude the following from its beliefs and rules:

$$\begin{array}{ccc} \not\triangleright_{012} \neg P & \not\triangleright_{012} \neg Q & \not\triangleright_{012} \neg S \\ \triangleright_{012} R & \triangleright_{012} \neg R & \triangleright_{012} S \end{array}$$

We observe that the same follows when using rules 1 and 3, since the implication becomes classical when necessity is applied; $\Box(R \rightarrow S) \equiv \Box(\neg S \rightarrow \neg R)$. Note that using \Rightarrow instead of \Box and \rightarrow would yield the same result.

This result differs from classical logic, where all the above propositions would follow from the rules and beliefs. The paraconsistent logic allows the agent to reason using inconsistent beliefs without entailing everything.

4 Conclusions

We have presented a paraconsistent logic defined using semantic clauses and motivated by key equalities. Although infinite-valued only a finite number of truth values must be considered for a given formula.

A small case study has been described and we have recently also investigated applications in logical semantics of natural language using a higher order logic extension with only propositional types [6].

References

1. D. Batens, C. Mortensen, G. Priest and J. Van-Benedem (editors). *Frontiers in Paraconsistent Logic*. Research Studies Press, 2000.
2. H. Decker, J. Villadsen and T. Waragai (editors). *International Workshop on Paraconsistent Computational Logic*. Volume 95 of Roskilde University, Computer Science, Technical Reports, 2002.
3. S. Gottwald. *A Treatise on Many-Valued Logics*. Research Studies Press, 2001.
4. J. Villadsen. *A Paraconsistent Higher Order Logic*. Pages 38–51 in B. Buchberger and J. A. Campbell (editors), Springer Lecture Notes in Computer Science 3249, 2004.
5. J. Villadsen. *Supra-Logic: Using Transfinite Type Theory with Type Variables for Paraconsistency*. Journal of Applied Non-Classical Logics, 15(1):45–58, 2005.
6. J. Villadsen. *Infinite-Valued Propositional Type Theory for Semantics*. Pages 277–297 in J.-Y. Béziau and A. Costa-Leite (editors), Dimensions of Logical Concepts, Unicamp Coleç. CLE 54, 2009.

Juliette Kennedy
Department of Mathematics and Statistics
University of Helsinki, Finland

April 11, 2012

In his 1946 Princeton Bicentennial Lecture Gödel suggested the problem of finding a notion of definability for set theory which is “formalism free” in a sense similar to the notion of computable function—a notion which is very robust with respect to the various formalisms associated with it. One way to interpret this suggestion is to consider standard notions of definability in set theory, which are usually built over first order logic, and change the underlying logic. We show that constructibility (in the sense of Gödel’s L) is not very sensitive to the underlying logic, and the same goes for hereditary ordinal definability (or HOD). We suggest that under an extensional notion of meaning for set theoretic discourse, Quine’s Dictum “change the logic change the meaning” is only partially true. This is joint work with Menachem Magidor and Jouko Väänänen.

LOGICAL LIMIT LAWS OF GRAPHS SUBJECT TO CONSTRAINTS

VERA KOPONEN

This talk is mainly concerned with questions regarding the probability that a first-order sentence is true in a large finite graph that satisfies some given constraint. In particular we are interested in, for a fixed but arbitrary sentence φ , whether the probability that φ is satisfied by a random graph with n vertices converges as n approaches infinity. Time permitting, I will also explain how one can show, using results from random graph theory, that for certain properties P , such as the property of being rigid, connected or hamiltonian (for instance), there is *no* logic \mathcal{L} extending first order logic such that \mathcal{L} can express P and \mathcal{L} has the compactness property.

For positive integers n we consider *undirected graphs*, from now on called *graphs*, with vertex set (or universe) $[n] = \{1, \dots, n\}$, viewed as first-order structures for a language whose vocabulary has, besides the identity symbol, a binary relation symbol that is interpreted as the edge relation in a graph. By the well known results of Glebski et. al. [Gle] and Fagin [Fag], if \mathbf{K}_n is the set of *all* graphs with vertex set $[n]$, then for every first-order sentence φ , the proportion of graphs in \mathbf{K}_n which satisfy φ approaches either 0 or 1 as n tends to infinity. In such a case we say that $\mathbf{K} = \bigcup_{n=1}^{\infty} \mathbf{K}_n$ satisfies a *zero-one law*. More generally, if, for positive integers n , \mathbf{K}_n is a set of graphs with vertex set $[n]$ and for every first-order sentence φ the proportion of graphs in \mathbf{K}_n in which φ is true converges as $n \rightarrow \infty$, then we say that $\mathbf{K} = \bigcup_{n=1}^{\infty} \mathbf{K}_n$ satisfies a (*logical*) *limit law*. Observe that the proportion of graphs in \mathbf{K}_n which have a given property can also be viewed as the probability, with the uniform probability measure on \mathbf{K}_n , that a random member of \mathbf{K}_n has that property.

Just as in infinite model theory, one may not be interested in all structures for a given language, but only those which satisfy certain constraints (expressed by a formal theory, for example). So for some property P , we may want to consider only those graphs with vertex set $[n]$ that have the property P . Denote the set of such graphs \mathbf{K}_n^P and let $\mathbf{K}^P = \bigcup_{n=1}^{\infty} \mathbf{K}_n^P$. I will now explain the known results within this context when $|\mathbf{K}_n^P|$ grows relatively fast as $n \rightarrow \infty$, which means that the methods developed by K. Compton and others (see [Bur]) are not applicable.

Given graphs \mathcal{H} and \mathcal{G} , we say that the graph \mathcal{G} is \mathcal{H} -free if it contains no subgraph which is isomorphic to \mathcal{H} .¹ Kolaitis, Prömel and Rothschild [KPR] have proved that if $l \geq 2$ and \mathcal{H} is a complete graph on $l+1$ vertices², also called an $(l+1)$ -clique, and P is the property of being \mathcal{H} -free, then \mathbf{K}^P satisfies a zero-one law. Lynch [Lyn] has proved that if d is a natural number and P is the property of being d -regular, which means that every vertex has degree d , then \mathbf{K}^P satisfies a limit law (but not zero-one law). In fact he has proved a more general result concerning graphs with specified degree sequences.

We will now consider properties of two kinds, that turn out to be related. The first kind of property P is that which, for some fixed natural number d , says “no vertex has degree higher than d ”. The other kind of property is that of being \mathcal{H} -free, where we consider \mathcal{H} of the form $\mathcal{K}_{1,s_1,\dots,s_l}$, where $l \geq 2$ and $1 \leq s_1 \leq \dots \leq s_l$ are integers and $\mathcal{K}_{1,s_1,\dots,s_l}$ denotes a graph whose vertex set can be partitioned into $l+1$ parts of cardinalities

¹We use the word *subgraph* in its graph theoretic sense. That is, \mathcal{H}_1 is a subgraph of \mathcal{H}_2 if the vertex set of \mathcal{H}_1 is a subset of the vertex set of \mathcal{H}_2 and every edge of \mathcal{H}_1 is an edge of \mathcal{H}_2 .

²That is, a graph with exactly $l+1$ vertices and with edges between all pairs of distinct vertices.

$1, s_1, \dots, s_l$ such that two vertices are adjacent (belong to an edge) if and only if they belong to different parts of this partition. Note that if we take $s_1 = \dots = s_l = 1$, then $\mathcal{K}_{1,s_1,\dots,s_l}$ is a complete graph on $l + 1$ vertices. Hundack, Prömel and Steger [HPS] have proved a result implying that if $l \geq 2$ and $1 \leq s_1 \leq \dots \leq s_l$ are fixed integers, then the proportion of $\mathcal{K}_{1,s_1,\dots,s_l}$ -free graphs with vertex set $[n]$ that have the following property approaches 1 as $n \rightarrow \infty$:

- (*) The vertex set can be partitioned into l parts such that every vertex has at most $s_1 - 1$ neighbours in its own part. In other words, for every part W of the partition, the induced subgraph with vertex set W has the property that no vertex has degree higher than $s_1 - 1$.

Thus to understand (eventual) limit laws for $\mathcal{K}_{1,s_1,\dots,s_l}$ -free graphs we should first understand, for any fixed $d \geq 0$, limit laws for graphs with maximum degree d , and graphs that satisfy (*). The first task is dealt with in [Kop1] where the following is proved.

Theorem 1. *Let $d \geq 0$ be an integer and let $\mathbf{G}_{n,d}$ denote the set of graphs \mathcal{G} with vertex set $[n] = \{1, \dots, n\}$ such that no vertex of \mathcal{G} has degree higher than d . Then, for every first-order sentence φ , the proportion of $\mathcal{G} \in \mathbf{G}_{n,d}$ that satisfy φ converges as $n \rightarrow \infty$. If $d \geq 2$ then this proportion need not converge to 0 or 1.*

In order to prove this result one needs to understand the structure of a “typical” member of $\mathbf{G}_{n,d}$ for large n . It turns out that if $d \geq 2$ (the non-trivial case), then key roles are played by the number of vertices of degree $d - 2$, short cycles and certain types of short paths as well as the the Poisson distribution. Once this is done, an Ehrenfeucht-Fraïssé game argument takes care of the rest.

For integers $l \geq 2$ and $d \geq 0$, let $\mathbf{P}_n(l, d)$ denote the set of graphs \mathcal{G} with vertex set $[n]$ such that $[n]$ can be partitioned into l parts such that every vertex of \mathcal{G} has at most d neighbours in its own part. In [Kop2] the following is proved, with the help of results from [Kop1] and the above mentioned result of Hundack, Prömel and Steger [HPS].

Theorem 2. *Let $l \geq 2$ and $d \geq 0$ be integers. Then, for every first-order sentence φ , the proportion of $\mathcal{G} \in \mathbf{P}_n(l, d)$ that satisfy φ converges as $n \rightarrow \infty$. If $d \leq 1$, then this proportion always converges to 0 or 1. Otherwise this need not be so.*

For any graph \mathcal{H} , let $\mathbf{F}_n(\mathcal{H})$ denote the set of \mathcal{H} -free graphs with vertex set $[n]$. With the help of Theorem 2 one can now prove, which is done in [Kop2], the following:

Theorem 3. *Let $l \geq 2$ and $1 \leq s_1 \leq \dots \leq s_l$ be integers. Then, for every first-order sentence φ , the proportion of $\mathcal{G} \in \mathbf{F}_n(\mathcal{K}_{1,s_1,\dots,s_l})$ that satisfy φ converges as $n \rightarrow \infty$. If $s_1 \leq 2$ then this proportion always converges to 0 or 1. Otherwise this need not be so.*

Time permitting, I will also explain how results from random graph theory, such as results saying that, for $d \geq 5$,

“almost all” finite d -regular graphs are rigid, connected and hamiltonian, can be used to show that if P is any of these properties (being rigid, connected or hamiltonian), then there does *not* exist a logic \mathcal{L} that extends first-order logic, can express P and has the compactness property. It is known that these properties cannot be expressed within first-order logic. The basic idea is that one can choose sentences φ_k , for $k \in \mathbb{N}$, such that every finite subset of $\{\varphi_k : k \in \mathbb{N}\}$ has a model, a finite graph, which is rigid, connected and hamiltonian, but every model of $\{\varphi_k : k \in \mathbb{N}\}$ is infinite, non-rigid, disconnected and therefore not hamiltonian. I believe that random graph theory can be used to get more results of this kind for other properties P than those mentioned.

[Bur] S. N. Burris, *Number Theoretic Density and Logical Limit Laws*, Mathematical Surveys and Monographs, Vol. 86, American Mathematical Society (2001).

- [Fag] R. Fagin, Probabilities on finite models, *The Journal of Symbolic Logic*, Vol. 41 (1976) 50-58.
- [Gle] Y. V. Glebskii, D. I. Kogan, M. I. Liogonkii, V. A. Talanov, Volume and fraction of satisfiability of formulas of the lower predicate calculus, *Kibernetika* Vol. 2 (1969) 17-27.
- [HPS] C. Hundack, H. J. Prömel, A. Steger, Extremal graph problems for graphs with a color-critical vertex, *Combinatorics, Probability and Computing*, Vol. 2 (1993) 465-477.
- [KPR] Ph. G. Kolaitis, H. J. Prömel, B. L. Rothschild, K_{l+1} -free graphs: asymptotic structure and a 0-1 law, *Transactions of The American Mathematical Society*, Vol. 303 (1987) 637-671.
- [Kop1] V. Koponen, Random graphs with bounded maximum degree: asymptotic structure and a logical limit law, submitted. Preprint available on:
<http://arxiv.org/pdf/1204.2446.pdf>
- [Kop2] V. Koponen, A limit law of almost l -partite graphs, submitted. Preprint available on:
<http://arxiv.org/pdf/1204.2454.pdf>
- [Lyn] J. F. Lynch, Convergence law for random graphs with specified degree sequence, *ACM Transactions on Computational Logic*, Vol. 6 (2005) 727-748.

VERA KOPONEN, DEPARTMENT OF MATHEMATICS, UPPSALA UNIVERSITY, BOX 480, 75106 UPPSALA, SWEDEN.

E-mail address: vera@math.uu.se

Undecidable First-Order Theories of Affine Geometries

Antti Kuusisto*, Jeremy Meyers†, Jonni Virtema*

May 30, 2012

1 Introduction

Tarski initiated a logic-based approach to formal geometry that studies first-order structures with a ternary *betweenness* relation (β) and a quaternary *equidistance* relation (\equiv). Tarski established, inter alia, that the first-order (FO) theory of $(\mathbb{R}^2, \beta, \equiv)$ is decidable. For further information on the development of Tarski's geometry, see [11]. Aiello and van Benthem conjectured in [1] that the FO-theory of the class of expansions of (\mathbb{R}^2, β) by unary predicates is decidable. We refute this conjecture by showing that for all $n \geq 2$, the FO-theory of the class of monadic expansions of (\mathbb{R}^n, β) is Π_1^1 -hard and therefore not even arithmetical. We also define a natural and comprehensive class \mathcal{C} of geometric structures (T, β) , where $T \subseteq \mathbb{R}^n$, and show that for each structure $(T, \beta) \in \mathcal{C}$, the FO-theory of the class of monadic expansions of (T, β) is undecidable. We then consider classes of expansions of structures (T, β) with restricted unary predicates, for example finite predicates, and establish a variety of related undecidability results. In addition to decidability questions, we briefly study the expressivity of universal MSO and weak universal MSO over expansions of (\mathbb{R}^n, β) . While the logics are incomparable in general, over expansions of (\mathbb{R}^n, β) , formulae of weak universal MSO translate into equivalent formulae of universal MSO.

Our results could turn out interesting in investigations concerning logical aspects of spatial databases. It turns out that there is a canonical correspondence between (\mathbb{R}^2, β) and $(\mathbb{R}, 0, 1, \cdot, +, <)$, see [7]. See the survey [9] for further details on logical aspects of spatial databases.

The betweenness predicate is also studied in spatial logic [3]. The recent years have witnessed a significant increase in the research on spatially motivated logics. Several interesting systems with varying motivations have been investigated, see the surveys [2] and [4]. Our results contribute to the understanding of spatially motivated first-order languages, and hence they can be useful in the search for decidable (modal) spatial logics.

2 Preliminaries

Tiling methods constitute a flexible framework for establishing different degrees of undecidability of different kinds of problems. An input to a tiling problem is a finite set of *tile types*, i.e., a finite set of rectangles with coloured edges. The problem is to decide whether it is possible to *tile* a predetermined region of space with tiles of the given type, under the constraint that adjacent edges of tiles have the same colour. We make use of the three following variants of the tiling problem. The standard *tiling problem* asks whether a set T of tile types can tile the $\mathbb{N} \times \mathbb{N}$ grid, the *recurrent tiling problem* asks whether T and some assigned tile type $t \in T$ can tile the $\mathbb{N} \times \mathbb{N}$ grid such that t occurs infinitely many times on the leftmost column of the grid, and the *torus tiling problem* asks if there exists some finite torus (i.e., a finite grid whose borders wrap around to form a torus) such that the input set T tiles it.

Theorem 2.1. *The tiling problem is Π_1^0 -complete [5], the recurrent tiling problem Σ_1^1 -complete [8], and the periodic tiling problem Σ_1^0 -complete [6].*

Let (\mathbb{R}^n, d) be the n -dimensional Euclidean space with the canonical metric d . We define the ternary Euclidean *betweenness* relation β such that $\beta(s, t, u)$ iff $d(s, u) = d(s, t) + d(t, u)$. We study geometric betweenness structures of the type (T, β) , where $T \subseteq \mathbb{R}^n$ and where β is the restriction of the betweenness predicate of \mathbb{R}^n to the set T .

A subset $S \subseteq \mathbb{R}^n$ is an *m -dimensional flat* of \mathbb{R}^n , where $0 \leq m \leq n$, if there exists a set of m linearly independent vectors $v_1, \dots, v_m \in \mathbb{R}^n$ and a vector $h \in \mathbb{R}^n$ such that S is the h -translated span of the vectors v_1, \dots, v_m , in other words $S = \{u \in \mathbb{R}^n \mid u = h + r_1 v_1 + \dots + r_m v_m, r_1, \dots, r_m \in \mathbb{R}\}$.

Note that $\{(0, \dots, 0)\}$ is not considered to be a linearly independent set.

A set $U \subseteq \mathbb{R}^n$ is a *linearly regular m -dimensional flat*, where $0 \leq m \leq n$, if the following conditions hold.

*University of Tampere, {antti.j.kuusisto, jonni.virtema}@uta.fi

†Stanford University, jjmeyers@stanford.edu

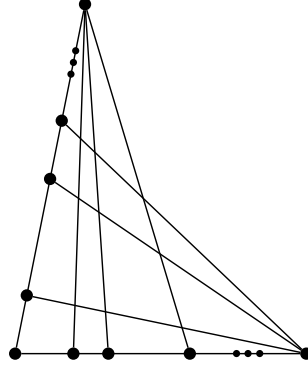


Figure 1: Interpreting the grid in (T, β, P, Q) .

1. There exists an m -dimensional flat S such that $U \subseteq S$.
2. There does not exist any $(m - 1)$ -dimensional flat S such that $U \subseteq S$.
3. U is *linearly complete*, i.e., if $L \subseteq U$ is a line in U and $L' \supseteq L$ the corresponding line in \mathbb{R}^n , and if $r \in L'$ is a point and $\epsilon \in \mathbb{R}_+$ a positive real number, then there exists a point $s \in L$ such that $d(s, r) < \epsilon$. Here d is the canonical metric of \mathbb{R}^n .
4. U is *linearly closed*, i.e., if points $x_1, x_2 \in U$ and $x_3, x_4 \in U$ determine two lines that intersect in \mathbb{R}^n , then the corresponding lines in U intersect in U .

A set $T \subseteq \mathbb{R}^n$ *extends linearly in mD* , where $m \leq n$, if there exists a linearly regular m -dimensional flat S , a positive real number $\epsilon \in \mathbb{R}_+$ and a point $x \in S \cap T$ such that $\{ u \in S \mid d(x, u) < \epsilon \} \subseteq T$. It is easy to show that for example the rational plane \mathbb{Q}^2 and the closed rectangle $[0, 1] \times [0, 1] \subseteq \mathbb{R}^2$ extend linearly in $2D$.

3 Results

While $\forall\text{WMSO} \not\leq \text{MSO}$ and $\forall\text{MSO} \not\leq \text{WMSO}$ in general, over models embedded in (\mathbb{R}^n, β) , $\forall\text{WMSO}$ translates into $\forall\text{MSO}$ and WMSO into MSO .

Theorem 3.1 (Heine-Borel). *A set $S \subseteq \mathbb{R}^n$ is closed and bounded iff every open cover of S has a finite subcover.*

Theorem 3.2. *Let \mathcal{C} be the class of expansions (\mathbb{R}^n, β, P) of (\mathbb{R}^n, β) with a unary predicate P , and let $\mathcal{F} \subseteq \mathcal{C}$ be the subclass of \mathcal{C} where P is finite. The class \mathcal{F} is first-order definable with respect to \mathcal{C} .*

Proof. It follows directly from the Heine-Borel theorem that a set $T \subseteq \mathbb{R}^n$ is finite iff it is closed, bounded and consists of isolated points of T . The proof of the current theorem relies on this fact. The argument is based on encoding topological information about open balls by first-order formulae. The idea is to replace open balls by *open n -dimensional triangles*.

We first define a formula $\text{parallel}(x, y, u, v)$ stating in \mathbb{R}^n that the lines defined by x, y and u, v are parallel. With this formula we construct formulae $\text{basis}_k(x_0, \dots, x_k)$ and $\text{flat}_k(x_0, \dots, x_k, z)$ by simultaneous recursion. The formulae state roughly that vectors (x_0, x_i) form a *basis* of an x_0 -centered k -dimensional flat, and that z is in the flat. With these formulae we recursively define formulae $\text{opentriangle}(x_0, \dots, x_k, z)$ stating that z is in the k -dimensional open triangle defined by the points x_0, \dots, x_k . □

The first-order theory of the class of expansions $(T, \beta, P_{i \in \mathbb{N}})$ of any structure (T, β) that extends linearly in $2D$ is undecidable. Here P_i are monadic predicates. This is shown by interpreting the $\mathbb{N} \times \mathbb{N}$ grid, or some superstructure of the grid, in the class of monadic expansions of (T, β) . This is done by defining two linear sequences of points that are stored in two predicates P and Q . The sequences correspond to linear orders that begin by a prefix of the order type ω . A superstructure of the grid is then interpreted by connecting the points of the linear sequence P to some upper bound of the sequence Q , and vice versa. The grid points are the intersection points of the lines created in this fashion, see the Figure 1 for an illustration.

Theorem 3.3. *Let $T \subseteq \mathbb{R}^n$ be a set that extends linearly in $2D$. The monadic Π_1^1 -theory of (T, β) is Σ_1^0 -hard.*

Extending linearly in $1D$ is not a sufficient condition for undecidability of the $\forall\text{MSO}$ -theory of (T, β) . This can be seen from the fact that the $\forall\text{MSO}$ -theory of $(\mathbb{Q}, <)$ is decidable [10].

In \mathbb{R}^n , it is possible to define predicates P and Q such that they correspond to linear sequences of the order type ω exactly. This enables the encoding of an isomorphic copy of the $\mathbb{N} \times \mathbb{N}$ grid in expansions of structures (\mathbb{R}^n, β) , where $n \geq 2$. With an isomorphic copy of the grid, we can interpret the *recurring* tiling problem in the structure created.

Theorem 3.4. *Let $n \geq 2$. The monadic Π_1^1 -theory of the structure (\mathbb{R}^n, β) is Π_1^1 -hard, and therefore not even arithmetical.*

When limiting attention to expansions of structures (T, β) with *finite* monadic predicates, we can use the periodic tiling problem in order to establish undecidability of monadic expansion classes of structures (T, β) .

Theorem 3.5. *Let $T \subseteq \mathbb{R}^n$ be a set that extends linearly in $2D$. The weak monadic Π_1^1 -theory of (T, β) is Π_1^0 -hard.*

In addition to expansions with finite predicates, the periodic tiling problem can be easily modified to yield undecidability of a wide variety of natural restricted monadic expansions classes of (\mathbb{R}^2, β) . These include expansions with predicates corresponding to finite unions of closed rectangles, polygons, and other simple canonical classes of sets.

In the future we shall identify fragments of first-order logic that lead to a decidable theory of the monadic expansion class of (\mathbb{R}^2, β) .

References

- [1] M. Aiello and J. van Benthem. A Modal Walk through Space. *Journal of Applied Non-Classical Logics* 12(3-4):319-363, Hermes, 2002.
- [2] M. Aiello, I. Pratt-Hartmann, and J. van Benthem. What is Spatial Logic. In [3].
- [3] M. Aiello, I. Pratt-Hartmann and J. van Benthem. Handbook of Spatial Logics. Springer, 2007.
- [4] P. Balbiani, V. Goranko, R. Kellerman and D. Vakarelov. Logical Theories for Fragments of Elementary Geometry. In [3].
- [5] R. Berger. The undecidability of the domino problem. *Mem. Amer. Math. Soc.*, 66, 1966.
- [6] Y. Gurevich and I. O. Koryakov. Remarks on Berger's paper on the domino problem. *Siberian Mathematical Journal* 13, 319-321, 1972.
- [7] M. Gyssens, J. Van den Bussche and D. Van Gucht. Complete Qeometric Query Languages. *J. Comput. System Sci.* 58, 483-511, 1999.
- [8] D. Harel. Recurring Dominoes: Making the Highly Undecidable Highly Understandable. *Annals of Discrete Mathematics* 24, 51-72, 1985.
- [9] B. Kijpers and J. Van den Bussche. Logical aspects of spatial database theory. In *Finite and Algorithmic Model Theory*, Cambridge University Press, 2011.
- [10] M. O. Rabin. Decidability of second-order theories and automata on infinite trees. *Trans. of the Amer. Math. Soc.* 141, 1-35, 1969.
- [11] A. Tarski and S. Givant. Tarski's System of Geometry. *Bull. Symbolic Logic* 5(2), 1999.

Stone Duality for Markov Processes

Kim Larsen

Department of Computer Science
University of Aalborg
Aalborg, Denmark

Radu Mardare

Department of Computer Science
University of Aalborg
Aalborg, Denmark

Prakash Panangaden

School of Computer Science
McGill University
Montreal, Canada

I. INTRODUCTION

The Stone representation [7] theorem is one of the recognized landmarks of mathematics. The Stone representation theorem [7] states that every (abstract) boolean algebra is isomorphic to a boolean algebra of sets; in modern terminology one has an equivalence of categories between the category of boolean algebras and the (opposite of) the category of compact Hausdorff zero-dimensional spaces, or Stone spaces.

In this paper we develop exactly such a duality for continuous-time continuous-space transitions systems where transitions are governed by an exponentially-distributed waiting time, essentially a continuous-time Markov chain (CTMC) with a continuous space. The logical characterization of bisimulation for such systems was proved a few years ago [3] using much the same techniques as were used for labelled Markov processes [5]. Recent work by the first two authors and Cardelli [1], [2] have established completeness theorems and finite model theorems for similar logics. Thus it seemed ripe to capture these logics algebraically and explore duality theory.

One of the critiques of logics and equivalences being used for the treatment of probabilistic systems is that boolean logic is not robust with respect to small perturbations of the real-valued system parameters. Accordingly, a theory of metrics [4] was developed and metric reasoning principles were advocated. In conjunction with our exploration of duality theory therefore we investigated the role of metrics and discovered a striking metric analogue of the duality theory. This paper describes both these theories. One can view the latter as the analogue of a completeness theorem for metric reasoning principles.

One of the points of departure of the present work from earlier work is the use of hemimetrics: analogues of pseudometrics that are not symmetric. This fits in well with the order structure of the boolean algebra. Nearly 25 years ago, Mike Smyth [6] advocated the use of such structure to combine metric and domain theory ideas. The interplay between the hemimetric and the boolean algebra is somewhat delicate and had to be carefully examined for the duality to emerge. It is a pleasant feature that exactly these axioms relating the hemimetric and the boolean algebra are satisfied in our examples without any artificial fiddling.

We summarize the key results of the present work:

- a description of a new class of algebras that captures, in algebraic form, the probabilistic modal logics used for continuous Markov processes,
- a duality between these algebras and continuous Markov processes
- a (hemi)metrized version of the algebras and of the Markov processes and
- a metric analogue of the duality.

II. DEFINITIONS

Let M be a set and $d : M \times M \rightarrow \mathbb{R}$.

Definition 1. We say that d is a hemimetric on M if for arbitrary $x, y \in M$,

- (1): $d(x, x) = 0$
- (2): $d(x, y) \leq d(x, z) + d(z, y)$

We say that (M, d) is a hemimetric space.

Note that a hemimetric is not necessarily symmetric nor does $d(x, y) = 0$ imply that $x = y$. A symmetric hemimetric is called a pseudometric.

Definition 2. For a hemimetric d on M we define the Hausdorff hemimetric d^H on the class of subsets of X by

$$d^H(X, Y) = \sup_{x \in X} \inf_{y \in Y} d(x, y).$$

We also define the dual of the Hausdorff hemimetric d_H on the class of subsets of X by

$$d_H(X, Y) = \sup_{y \in Y} \inf_{x \in X} d(x, y).$$

Definition 3 (Continuous Markov processes). Given a measurable space (M, Σ) , a continuous Markov process (CMK) is a tuple $\mathcal{M} = (M, \Sigma, \theta)$, where $\theta \in \llbracket M \rightarrow \Delta(M, \Sigma) \rrbracket$. M is the support set of \mathcal{M} denoted by $\text{supp}(\mathcal{M})$. If $m \in M$, (\mathcal{M}, m) is a continuous Markov process (CMP).

Definition 4 (Aumann algebra). An Aumann algebra (AA) over the set $B \neq \emptyset$ is a structure $\mathcal{A} = (B, \top, \perp, \sim, \sqcup, \sqcap, \{F_r, G_r\}_{r \in \mathbb{Q}^+}, \sqsubseteq)$ where $\mathcal{B} = (B, \top, \perp, \sim, \sqcup, \sqcap, \sqsubseteq)$ is a meet-continuous boolean Algebra, for each $r \in \mathbb{Q}^+$, $F_r, G_r : B \rightarrow B$ are monadic operations and the elements of B satisfy the axioms in Table I, for arbitrary $a, b \in B$ and $r, s \in \mathbb{Q}^+$.

Definition 5 (Metrized Aumann algebra). A metrized Aumann algebra is a tuple (\mathcal{A}, δ) , where $\mathcal{A} = (B, \top, \perp, \sim, \sqcup, \sqcap, \{F_r, G_r\}_{r \in \mathbb{Q}^+}, \sqsubseteq)$ is an Aumann algebra and $\delta : B \times B \rightarrow [0, 1]$ is a hemimetric on B satisfying, for arbitrary $a, b \in B$,

- (AA1): $\top \sqsubseteq F_0 a$
(AA2): $F_{r+s} a \sqsubseteq \sim G_r a$, for $s > 0$
(AA3): $\sim F_r a \sqsubseteq G_r a$
(AA4): $(\sim F_r(a \sqcap b)) \sqcap (\sim F_s(a \sqcap \sim b)) \sqsubseteq \sim F_{r+s} a$
(AA5): $(\sim G_r(a \sqcap b)) \sqcap (\sim G_s(a \sqcap \sim b)) \sqsubseteq \sim G_{r+s} a$
(AA6): If $a \sqsubseteq b$ then $F_r a \sqsubseteq F_r b$
(AA7): $\bigwedge \{F_r b \mid r < s\} = F_s b$
(AA8): $\bigwedge \{G_r b \mid r > s\} = G_s b$
(AA9): $\bigwedge \{F_r b \mid r > s\} = \perp$

TABLE I
AUMANN ALGEBRA

and arbitrary filtered set $A \subseteq B$ for which there exists $\bigwedge A'$ in B , the axioms in Table II.

- (H0): if $\delta(a, b) = 0$, then $a \sqsubseteq b$
(H1): $\delta(a, b) = \delta(a \sqcap (\sim b), b)$
(H2): $\delta(b, \bigwedge A) = \inf_{a \in A} \delta(b, a)$
(H3): $\delta(\bigwedge A, b) = \sup_{a \in A} \delta(a, b)$

TABLE II
HEMIMETRIC AXIOMS FOR METRIZED AA

III. RESULTS

We have a duality theorem between CMPs and Aumann Algebras.

Theorem 6 (Representation Theorem). (i) Any CMP $\mathcal{M} = (M, \Sigma, \theta)$ is bisimilar to $\mathcal{M}(\mathcal{L}(\mathcal{M}))$ and the bisimulation relation is given by the mapping α defined, for arbitrary $m \in M$, by

$$m \mapsto \alpha(m) = \{\phi \in \mathcal{L}(\mathcal{M}) \mid \mathcal{M}, m \models \phi\}.$$

(ii) Any Aumann algebra $\mathcal{A} = (B, \top, \perp, \sim, \sqcup, \sqcap, \{F_r, G_r\}_{r \in \mathbb{Q}^+}, \sqsubseteq)$ is isomorphic to $\mathcal{L}(\mathcal{M}(\mathcal{A}))$ and the isomorphism is given by the mapping β defined, for arbitrary $a \in B$, by

$$a \mapsto \beta(a) =$$

$$\bigwedge (\{\phi \in \mathcal{L}(\mathcal{M}(\mathcal{A})) \mid \forall u \in \mathcal{U}(B) \text{ s. t. } \uparrow(a) \sqsubseteq u, \mathcal{M}(\mathcal{A}), u \models \phi\}).$$

This extends to a duality between the hemi-metric spaces in the following sense.

Theorem 7 (The metric duality theorem). (i) Given a metrized CMP (\mathcal{M}, d) with $\mathcal{M} = (M, \Sigma, \theta)$, \mathcal{M} is bisimilar to $\mathcal{M}(\mathcal{A}(\mathcal{L}(\mathcal{M})))$ by the map α defined in the Representation Theorem and, in addition, for arbitrary $m, n \in M$,

$$d(m, n) = (d^H)_H(\alpha(m), \alpha(n)).$$

(ii) Given a metrized AA (\mathcal{A}, δ) with $\mathcal{A} = (B, \top, \perp, \sim, \sqcup, \sqcap, \{F_r, G_r\}_{r \in \mathbb{Q}^+}, \sqsubseteq)$, \mathcal{A} is isomorphic to $\mathcal{A}(\mathcal{L}(\mathcal{M}(\mathcal{A})))$ by the map β defined in the Representation Theorem and, in addition, for arbitrary $a, b \in B$

$$\delta(a, b) = (\delta^H)_H(\beta(a), \beta(b)).$$

REFERENCES

- [1] Luca Cardelli, Kim G. Larsen, and Radu Mardare. Continuous markovian logic - from complete axiomatization to the metric space of formulas. In *CSL*, pages 144–158, 2011.
- [2] Luca Cardelli, Kim G. Larsen, and Radu Mardare. Modular markovian logic. In *ICALP (2)*, pages 380–391, 2011.
- [3] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Approximating labeled Markov processes. *Information and Computation*, 184(1):160–200, July 2003.
- [4] Josée Desharnais, Vineet Gupta, Radhakrishnan Jagadeesan, and Prakash Panangaden. A metric for labelled Markov processes. *Theoretical Computer Science*, 318(3):323–354, June 2004.
- [5] Prakash Panangaden. *Labelled Markov Processes*. Imperial College Press, 2009.
- [6] Michael B. Smyth. Quasi uniformities: Reconciling domains with metric spaces. In *MFPS*, pages 236–253, 1987.
- [7] Marshall H. Stone. The theory of representations for boolean algebras. *Trans. Amer. Math. Soc.*, 40:37–111, 1936.

Intuitionistic logic and partial probability functions

François Lepage
Université de Montréal

Abstract

The aim of the present paper is to provide a semantical analysis of propositional intuitionistic logic with strong negation in term of partial probability functions. The system is proved to be sound and complete for these interpretations. Our analysis is based on what is called ‘Adam’s assumption’, i.e. that the probability of an indicative conditional $A \rightarrow B$ is the probability of B conditional on A . In the first section, we make a short review of the state of the art, including Lewis’s famous triviality result: starting from the assumption that $\Pr(A \rightarrow B) = \Pr(B / A)$ for any A and B such that $\Pr(A) \neq 0$, he showed that there are at most two incompatible sentences in the language. Surprisingly, Morgan and Leblanc [6] showed that it is possible to define probabilistic semantic for intuitionistic logic but in their canonical model every non-theorem has a zero probability. This is quite problematic since for any classical tautology A and any B such that A is not an intuitionistic consequence of B , $\Pr(A, B) = 0$ (they used Popper probability functions instead of absolute ones). Worst, Morgan and Mares [7] showed that the implicational fragment of intuitionistic logic is the weakest logic for which the probability of the conditional is the conditional probability. They also showed that if negation is added to the fragment (even a very weak one), then the semantics is trivial. In the second section, we present the following system for intuitionistic logic with Nelson’s [8] strong negation.

- I1 $A \rightarrow (B \rightarrow A)$
- I2 $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$
- I3 $A \wedge B \rightarrow A$
- I4 $A \wedge B \rightarrow B$
- I5 $A \rightarrow A \vee B$
- I6 $B \rightarrow A \vee B$
- I7 $(A \rightarrow C) \rightarrow ((B \rightarrow C) \rightarrow (A \vee B \rightarrow C))$
- I8 $F \rightarrow A$
- PI1 $\sim\sim A \rightarrow A$
- PI2 $A \rightarrow \sim\sim A$
- PI3 $\sim(A \wedge B) \rightarrow \sim A \vee \sim B$
- PI4 $\sim(A \vee B) \rightarrow \sim A \wedge \sim B$
- PI5 $A \wedge \sim A \rightarrow F$
- PI6 $\sim(A \rightarrow B) \rightarrow (A \wedge \sim B)$
- PI7 $(A \wedge \sim B) \rightarrow \sim(A \rightarrow B)$
- PI8 $\sim \neg A \rightarrow A$
- PI9 $A \rightarrow \sim \neg A$
- PI20 $\sim A \rightarrow (A \rightarrow B)$
- PI21 $A \rightarrow (B \rightarrow (A \wedge B))$
- PI22 $\sim A \vee \sim B \rightarrow \sim(A \wedge B)$
- PI23 $\sim A \wedge \sim B \rightarrow \sim(A \vee B)$

where $\neg A$ (the intuitionistic negation) stand for $A \rightarrow F$ and MP is the only rule.

Following Lepage and Morgan [3], we introduce the notion of *partial* conditional probability function. A partial conditional probability function is a partial function Pr

$$\text{Pr} : L \times 2^L \rightarrow [0,1]$$

such that the following constraints are always satisfied:

- P.1 If $A \in \Gamma$, then $\text{Pr}(A, \Gamma)$ is defined;
- P.2 If $\text{Pr}(A, \Gamma)$ is defined, then $\text{Pr}(\sim A, \Gamma)$ is defined;
- P.3 If $\text{Pr}(\sim A, \Gamma)$ is defined, then $\text{Pr}(A, \Gamma)$ is defined;
- P.4 If $\text{Pr}(A \wedge B, \Gamma)$ is defined, then $\text{Pr}(B \wedge A, \Gamma)$ is defined;
- P.5 If $\text{Pr}(A, \Gamma)$ and $\text{Pr}(B, \Gamma)$ are undefined $\text{Pr}(A \wedge B, \Gamma)$ is undefined;
- P.6 If $\text{Pr}(A, \Gamma) = 0$, then $\text{Pr}(A \wedge B, \Gamma) = 0$;
- P.7 If $\text{Pr}(A, \Gamma) = 1$, then $\text{Pr}(A \vee B, \Gamma) = 1$;
- P.8 If $\text{Pr}(A, \Gamma)$ and $\text{Pr}(B, \Gamma)$ are undefined then $\text{Pr}(A \wedge B, \Gamma)$ is undefined;
- P.9 If $\text{Pr}(A \wedge B, \Gamma)$ is defined and $\text{Pr}(A, \Gamma)$ is undefined, then $\text{Pr}(B, \Gamma) = 0$;
- P.10 If $\text{Pr}(A \vee B, \Gamma)$ is defined and $\text{Pr}(A, \Gamma)$ is undefined, then $\text{Pr}(B, \Gamma) = 1$.

And, following Morgan [4], [5] when all the appropriate values of Pr are defined, the following constraints are satisfied :

- NP.1 $0 \leq \text{Pr}(A, \Gamma) \leq 1$;
- NP.2 If $A \in \Gamma$, then $\text{Pr}(A, \Gamma) = 1$;
- NP.3 $\text{Pr}(A \vee B, \Gamma) = \text{Pr}(A, \Gamma) + \text{Pr}(B, \Gamma) - \text{Pr}(B \wedge A, \Gamma)$;
- NP.4 $\text{Pr}(A \wedge B, \Gamma) = \text{Pr}(A, \Gamma) \times \text{Pr}(B, \Gamma \cup \{A\})$;
- NP.5 $\text{Pr}(\sim A, \Gamma) = 1 - \text{Pr}(A, \Gamma)$ provided Γ is Pr -normal (i.e., there is at least an A such that $\text{Pr}(A, \Gamma) = 0$);
- NP.6 $\text{Pr}(A \wedge B, \Gamma) = \text{Pr}(B \wedge A, \Gamma)$;
- NP.7 $\text{Pr}(C, \Gamma \cup \{A \wedge C\}) = \text{Pr}(C, \Gamma \cup \{A, C\})$;
- NP.8 $\text{Pr}(A \rightarrow B, \Gamma) = \text{Pr}(B, \Gamma \cup \{A\})$.

The notion of semantic consequence is introduced:

A is a semantical consequence of Γ (written $\Gamma \models A$) if and only if for all partial probability distributions Pr , $\text{Pr}(A, \Gamma \cup \Delta) = 1$ for all Δ .

We prove soundness of I1-I8 and PI1-PI-23.

For completeness, we use the notion of saturated sets Aczel [1].

A *Saturated Deductively Closed Consistent Set* (SDCCS) Γ is a set which is:

- (1) Saturated, i.e. $(A \vee B) \in \Gamma$ iff $A \in \Gamma$ or $B \in \Gamma$; and
- (2) Deductively closed, i.e. $A \in \Gamma$ iff $\Gamma \vdash A$; and
- (3) Consistent, i.e. there is an A such that $\Gamma \not\vdash A$.

It is a well known result that the set W of all saturated sets define a canonical Kripke frame [2] $\langle W, \subseteq \rangle$ where \subseteq is simply inclusion between sets.

Let Γ be any set. We define $U(\Gamma) = \{\Delta : \Delta \text{ is a SDCCS and } \Gamma \subseteq \Delta\}$.

We define a function $\text{Pr}_{\langle W, \subseteq \rangle}$ such that for any A

$$\text{Pr}_{\langle W, \subseteq \rangle}(A, \Gamma) = \begin{cases} 1 & \text{iff } A \in \Delta \text{ for all } \Delta \in U(\Gamma) \text{ such that } \Gamma \subseteq \Delta \\ 0 & \text{iff } \neg A \in \Delta \text{ for all } \Delta \in U(\Gamma) \text{ such that } \Gamma \subseteq \Delta \\ \text{undefined} & \text{otherwise} \end{cases}$$

We show that

$\text{Pr}_{\langle W, \subseteq \rangle}$ satisfy P.1-8 and NP.1-8

Furthermore, completeness is proved, because if $\Gamma \not\vdash A$, then there is a saturated set Δ such that $A \notin \Delta$ and $\Gamma \subseteq \Delta$.

Finally, in the last section, we discuss questions of triviality and we prove that when partial functions are forced to be total, then triviality is back.

References

- [1] Aczel P.H.G. “Saturated Intuitionistic Theories”, *Studies in Logic and the Foundations of Mathematics*, Volume 50, 1968, pp. 1-11, Contributions to Mathematical Logic - Proceedings of the Logic Colloquium, Hannover 1966.
- [2] Kripke, S., “Semantical analysis of intuitionistic logic I”, in *Formal systems and recursive functions*, North Holland, Amsterdam (1965), 92-130.
- [3] Lepage F. and Morgan C., “Probabilistic Canonical Models for Partial Logic”, *Notre Dame Journal of Formal Logic*, 44 (3), 2003, pp 125-138.
- [4] Morgan C., “There is a Probabilistic Semantics for Every Extension of Classical Sentence Logic”, *Journal of Philosophical Logic*, 11 (1982), pp. 431-442.
- [5] Morgan C., “Canonical Models and Probabilistic Semantics” pp. 17-29 in *Logic, Probability, and Science*, edited by N. Shanks and Gardner R.B., Rodopi, Atlanta/Amsterdam, 2000.
- [6] Morgan C. and Leblanc H., “Conditional, Probability, and Non-triviality”, *Journal of Philosophical Logic*, Vol. 24, no2, 1983, pp. 161-180.
- [7] Morgan C. and Mares E., “Probabilistic Semantics for Intuitionistic Logic”, *Notre Dame Journal of Formal Logic*, Vol. 24, 1995, pp. 455-467.
- [8] Nelson D., “Constructible Falsity”, *The Journal of Symbolic Logic*, Vol 14, no1, 1949, pp. 16-26.

Algorithmic Intensionality in Type Theory of Acyclic Recursion and Underspecification

Roussanka Loukanova

1 Background of L_{ar}^λ

Moschovakis [2] initiated development of a new approach to the mathematical notion of algorithm. A potential prospect for applications of the new approach is to computational semantics of artificial and natural languages¹ (NLs). In particular, the theory of acyclic recursion L_{ar}^λ , see Moschovakis [3], models the concepts of meaning and synonymy in typed models. Moschovakis formal system L_{ar}^λ is a higher-order type theory, which is a proper extension of Gallin's TY_2 , see Gallin [1], and thus, of Montague's Intensional Logic (IL), see Montague [4]. The type theory L_{ar}^λ and its calculi extend Gallin's TY_2 , at the level of the formal language and its semantics, by using several means: (1) two kinds of variables (*recursion variables*, called alternatively *locations*, and *pure variables*); (2) by formation of an additional set of *recursion terms*; (3) systems of rules that form various calculi, i.e., the reduction calculus and the calculus of referential synonymy. In the first part of the talk, we give the formal definitions of the syntax and denotational semantics of the language of L_{ar}^λ . Then, we introduce the intensional semantics of L_{ar}^λ . The second part of the talk is devoted to the necessity of extending a λ -calculus, like Montague's IL, to a system like L_{ar}^λ , for computational semantics of NL expressions.

2 Brief introduction to the type theory L_{ar}^λ

Types of L_{ar}^λ : The set *Types* is the smallest set defined recursively (using a wide-spread notation in computer science): $\tau ::= e \mid t \mid s \mid (\tau_1 \rightarrow \tau_2)$.

2.1 Syntax of L_{ar}^λ

The vocabulary of L_{ar}^λ consists of pairwise disjoint sets, for each type τ : $K_\tau = \{c_0, c_1, \dots, c_{k_\tau}\}$, a finite set of *constants* of type τ ; $PureVars_\tau = \{v_0, v_1, \dots\}$, a set of *pure variables* of type τ ; $RecVars_\tau = \{p_0, p_1, \dots\}$, a set of *recursion variables*, called also *locations*, of type τ .

¹*Natural Language (NL)* is a traditional way of address to human languages. We maintain the view that natural languages form a broader class of languages in nature.

The Terms of L_{ar}^λ : In addition to application and λ -abstraction terms, L_{ar}^λ has recursion terms that are formed by using a designated recursion operator, which is denoted by the constant `where` and can be used in infix notation. The recursive rules² for the set of L_{ar}^λ terms can be expressed by using a notational variant of “typed” BNF:

$$A ::= c^\tau : \tau \mid x^\tau : \tau \mid B^{(\sigma \rightarrow \tau)}(C^\sigma) : \tau \mid \lambda v^\sigma(B^\tau) : (\sigma \rightarrow \tau) \\ \mid A_0^\sigma \text{ where } \{p_1^{\sigma_1} := A_1^{\sigma_1}, \dots, p_n^{\sigma_n} := A_n^{\sigma_n}\} : \sigma$$

where $\{p_1^{\sigma_1} := A_1^{\sigma_1}, \dots, p_n^{\sigma_n} := A_n^{\sigma_n}\}$ is a set of assignments that satisfies the *acyclicity condition* defined as follows: For any terms $A_1 : \sigma_1, \dots, A_n : \sigma_n$, and locations $p_1 : \sigma_1, \dots, p_n : \sigma_n$ (where $n \geq 0$, and $p_i \neq p_j$ for all i, j such that $i \neq j$ and $1 \leq i, j \leq n$), the set $\{p_1 := A_1, \dots, p_n := A_n\}$ is an *acyclic system of assignments* iff there is a function $\text{rank} : \{p_1, \dots, p_n\} \rightarrow \mathbb{N}$ such that, for all $p_i, p_j \in \{p_1, \dots, p_n\}$, if p_j occurs free in A_i then $\text{rank}(p_j) < \text{rank}(p_i)$.

Terms of the form $A_0^\sigma \text{ where } \{p_1^{\sigma_1} := A_1^{\sigma_1}, \dots, p_n^{\sigma_n} := A_n^{\sigma_n}\}$ are called *recursion terms*. Intuitively, a system $\{p_1 := A_1, \dots, p_n := A_n\}$ defines recursive computations of the values to be assigned to the locations p_1, \dots, p_n . Requiring a ranking function rank , such that $\text{rank}(p_j) < \text{rank}(p_i)$, means that the value of A_i , which is assigned to p_i , may depend on the values of the location p_j , as well as on the values of the locations p_k with lower rank than p_j . An acyclic system guarantees that computations close-off after a finite number of steps. Omitting the acyclicity condition gives an extended type system L_r^λ , which admits full recursion.

2.2 Two kinds of semantics of L_{ar}^λ

Denotational Semantics of L_{ar}^λ : The language L_{ar}^λ has denotational semantics that is given by a definition of a denotational function for any semantic structure with typed domain frames. The denotational semantics of L_{ar}^λ follows the structure of the L_{ar}^λ terms, in a compositional way.

Intensional Semantics of L_{ar}^λ : The notion of intension in the languages of recursion covers the most essential, computational aspect of the concept of meaning. The *referential intension*, $\text{Int}(A)$, of a meaningful term A is the tuple of functions (a recursor) that is defined by the denotations $\text{den}(A_i)$ ($i \in \{0, \dots, n\}$) of the parts (i.e., the head sub-term A_0 and of the terms A_1, \dots, A_n in the system of assignments) of its canonical form $\text{cf}(A) \equiv A_0 \text{ where } \{p_1 := A_1, \dots, p_n := A_n\}$. Intuitively, for each meaningful term A , the intension of A , $\text{Int}(A)$, is the *algorithm* for computing its denotation $\text{den}(A)$. Two meaningful expressions are synonymous iff their referential intensions are naturally isomorphic, i.e., they are the same algorithms. Thus, the algorithmic meaning of a meaningful term (i.e., its sense) is the information about how to “compute” its denotation step-by-step: a meaningful term has sense by carrying instructions within its structure, which are revealed by its canonical form, for acquiring what they denote in a model.

²In an explicit definition of the L_{ar}^λ terms, the acyclicity condition is a proper part of the case of recursion terms, as the above notational variant of BNF is taken.

The canonical form $\text{cf}(A)$ of a meaningful term A encodes its intension, i.e., the algorithm for computing its denotation, via: (1) the basic instructions (facts), which consist of $\{p_1 := A_1, \dots, p_n := A_n\}$ and the head term A_0 , that are needed for computing the denotation $\text{den}(A)$, and (2) a terminating rank order of the recursive steps that compute each $\text{den}(A_i)$, for $i \in \{0, \dots, n\}$, for incremental computation of the denotation $\text{den}(A) = \text{den}(A_0)$. Thus, the languages of recursion offer a formalisation of central computational aspects: denotation, with (at least) two semantic “levels”: *referential intensions (algorithms)* and *denotations*. The terms in canonical form represent the algorithmic steps for computing semantic denotations by using all necessary basic components:

$$\text{NL Syntax} \implies L_{ar}^\lambda / L_r^\lambda \implies \underbrace{\text{Referential Intensions (Algorithms)} \implies \text{Denotations}}_{\text{Computational Semantics}}$$

3 Semantic Underspecification

In this part, by using linguistic motivations, we present arguments for the new kind of recursion variables and the distinctions between λ -calculus terms, recursion terms, and canonical forms, subject to weak β -reduction. One of the distinctive characteristics of the algorithmic theory of L_{ar}^λ is the possibility to formalize the concept of semantic underspecification, at the object level of the language L_{ar}^λ . We give renderings of NL expressions into L_{ar}^λ terms that represent computational patterns with potentials for further specifications depending on context. E.g., we present a class of terms rendering underspecified quantifier scopes. Furthermore, the language and theory L_{ar}^λ of typed acyclic recursion provide means for a generalized, predicative operator that is underspecified L_{ar}^λ term. We are presenting the potential use of such an operator for computational semantics of predicative expressions, e.g., headed by the copula verb *be*. The values of the predicative operator are specified by the phrasal environments of the copula.

References

- [1] D. Gallin. *Intensional and Higher-Order Modal Logic*. North-Holland, 1975.
- [2] Y. N. Moschovakis. Sense and denotation as algorithm and value. In J. Oikkonen and J. Vaananen, editors, *Lecture Notes in Logic*, number 2 in Lecture Notes in Logic, pages 210–249. Springer, 1994.
- [3] Y. N. Moschovakis. A logical calculus of meaning and synonymy. *Linguistics and Philosophy*, 29:27–89, 2006.
- [4] R. H. Thomason, editor. *Formal Philosophy: Selected Papers of Richard Montague*, ed. Richmond Thomason. Yale University Press, New Haven, Connecticut, 1974. Edited, with an introduction, by Richmond H. Thomason.

- KERKKO LUOSTO, *Unary quantifiers on finite and infinite structures*.
 Department of Mathematics and Statistics, University of Helsinki, P.O. Box 68 (Gustaf
 Hållströmin katu 2b), FI-00014 University of Helsinki, Finland.
E-mail: Kerkko.Luosto@Helsinki.FI.

Unary generalized quantifiers are logical constructs that enable us to express cardinal properties of unary predicates such as “there are as many x ’s satisfying ... as y ’s satisfying ...”. (This particular example is *the Hårtig or equicardinality quantifier I.*) I shall review some of the latest developments in the definability theory of such quantifiers, both from the perspective of finite and infinite structures.

Two special classes of interest are \mathcal{O} , the class of finite ordered structures, and \mathcal{O}_ω , the class of expansions of $\langle \mathbb{N}, \leq \rangle$. Let us say that a generalized quantifier Q has *finite character*, if it is universe-independent, its vocabulary τ is finite, and for every $\mathfrak{A} \in K_Q$ we have that $\bigcup_{R \in \tau} R^{\mathfrak{A}}$ is finite. Then we have the following simple transfer result: If the generalized quantifier Q has finite character and \mathcal{L} is regular, then definability in finite implies definability in infinite, or more formally, $\text{FO}(Q) \leq \mathcal{L}(\mathcal{O})$ implies $\text{FO}(Q) \leq \mathcal{L}(\mathcal{O}_\omega)$.

At first sight, it appears as if one could prove new undefinability theorems using the definability transfer result above. However, there is an obstacle that seems to make this improbable, at least as long as applications in descriptive complexity theory are concerned: Let $S \subseteq \mathbb{N}$ be fixed, and consider the logic $\text{FO}(\mathbf{C}_S)$ where the cardinality quantifier \mathbf{C}_S expresses “there are S -many”. Then we get the following dichotomy: either

S is eventually periodic,
 $\text{FO}(\mathbf{C}_S) < \text{FO}(\mathbf{I})(\mathcal{O}_\omega)$,
 $\text{FO}(\mathbf{C}_S)$ -theory of $\langle \mathbb{N}, \leq \rangle$ is decidable,
 and addition and multiplication are not definable in $\langle \mathbb{N}, \leq \rangle$,

or

S is aperiodic,
 $\text{FO}(\mathbf{C}_S) \geq \text{FO}(\mathbf{D})(\mathcal{O}_\omega)$,
 the corresponding theory $\text{Th}_{\text{FO}(\mathbf{C}_S)}(\langle \mathbb{N}, \leq \rangle)$ is undecidable,
 and addition and multiplication are definable.

Here, \mathbf{D} is the general divisibility quantifier.

More generally, if we consider logics $\text{FO}(Q)$ with a unary quantifier Q instead of a cardinality quantifier \mathbf{C}_S , we get a trichotomy with a new case falling in the middle of the two old cases and corresponding roughly to Presburger arithmetic. The proofs rely on quite basic combinatorics of words and the result of Krynicky and Lachlan [KL79] that $\text{FO}(\mathbf{I})$ -theory of $\langle \mathbb{N}, \leq \rangle$ is decidable.

In comparison, the analogous results in finite model theory are rather involved. In [Luo04], cardinality quantifiers \mathbf{C}_S such that $\text{FO}(\mathbf{I}) \leq \text{FO}(\mathbf{C}_S)(\mathcal{O})$ were characterized, and there are simple examples of quantifiers \mathbf{C}_S such that \mathbf{C}_S and \mathbf{I} are incomparable. Hella, Luosto and Kontinen [HKL10] give a sufficient condition for $\text{FO}(\mathbf{D}) \leq \text{FO}(\mathbf{C}_S)(\mathcal{O})$ to hold.

[HKL10]Lauri Hella, Juha Kontinen, and Kerkko Luosto. Regular representations of uniform TC^0 . Preprint, Mittag-Leffler Institute, Djursholm, autumn 2009/2010.

[KL79]Michał Krynicky and Alistair H. Lachlan. On the semantics of the Henkin quantifier. *J. Symbolic Logic*, 44(2):184–200, 1979.

[Luo04]Kerkko Luosto. Equicardinality on linear orders. In *Proceedings of the Nineteenth Annual IEEE Symposium on Logic in Computer Science (LICS 2004)*, pages 458–465. IEEE Computer Society Press, July 2004.

Multiplayer Game Semantics and Non-Polemic Conversation

Mathias Winther Madsen
University of Amsterdam
`mathias.winther@gmail.com`

May 30, 2012

Natural languages have a systematic ambiguity with respect to existential items such as disjunction, indefinite articles, possibility modals, and existential quantification. For instance, the sentence

(1) You may leave

is ambiguous between a possibility-reading and a permission-reading [1]. Even though ambiguities of this kind are frequently described in terms of “free choice,” the nature of these “choices” have not been described in the context of a social model that explicates how the parties to a conversation manage such choices.

In this abstract, I introduce a model that fills this gaps, drawing on ideas from Hintikka’s Game Semantics [4]. The resulting system accounts for a number of ambiguities of existential items such as *may*, *can*, *or*, *a*, *the*, and *some*.

A key property of the system I present here is the fact that it allows hearers to be more or less charitable in their interpretation of an utterance, within limits set out by word meanings. Classical logic then falls out as a special case where speaker and hearer have opposing interests, and the evidence standards are minimal.

1 Polemic Multiplayer Game Semantics

In its original form, Hintikka’s Game Semantics can be seen as a road map for how to debate an assertion: It specifies how a hearer may legitimately raise objections against it, and how a speaker may legitimately defend it. They do so by traversing downwards through the syntactic tree of the sentence in search of an unanalyzable atom which supports their own case.

The system I present here is similar, but it allows the speaker and hearer to rely on other agents as sources of evidence. This evocation of the third-person perspective is intended to simulate certain rhetorical moves, not actual acts of asking other people for information. Accordingly, the game includes two “ghost”

versions of all agents: one sympathetic to the speaker, and one sympathetic to the hearer.

The game is played over sentence from a standard first-order syntax with, for simplicity, a single modality. I assume that duals are taken as primitive, so that the syntax contains, e.g., both conjunction and disjunction. The game is played within the context of a single model (with an actual world w^*).

Definition 1. (Game elements) The game consists of the following elements:

1. A **sentence** φ .
2. A **set of players**, $N = \{S_1, S_2, \dots, S_n\} \cup \{H_1, H_2, \dots, H_n\}$. By convention, S_1 is the speaker, and H_2 is the hearer.
3. A set of **histories**, with a subset being **terminal histories**; both are defined by the player function introduced below.
4. For each player, a **preference relation** over the terminal histories. The preferences of S_i are assumed to be the same for all i , and likewise for H_i .
5. For each player, an **information partition** on the set of models. S_i and H_i are assumed to have the same information partition for all i .

For simplicity, we can let the preference relations be given by two **utility functions** u_1 and u_2 . In the polemic game, $u_1 + u_2 = 0$, and player i **wins** the game in a terminal history h if and only if $u_i(h) > 0$.

Definition 2. (Game state) A game state consists of a sentence (the **claim**), a world (the **current world**), and two agents (the **Verifier** and **Falsifier**). I use the notation $G(w \vdash \psi)$ for the state in which the claim is ψ and the current world is w (thus omitting, for brevity, the identity of the Verifier and Falsifier).

Definition 3. (Player function) The game starts at $G(w^* \vdash \varphi)$ with the speaker as Verifier and the hearer as Falsifier. From an arbitrary state $G(w \vdash \psi)$, it transitions into the next state according to the following rules:

Connectives. If $\psi = \theta \vee \eta$ ($\psi = \theta \wedge \eta$), then the Verifier (Falsifier) picks another player; this player then picks one of the states $G(w \vdash \theta)$ and $G(w \vdash \eta)$, and the game transitions into that state.

Quantifiers. If $\psi = \exists x \theta$ ($\psi = \forall x \theta$), then the Verifier (Falsifier) picks another player; this player then picks an individual a , and the game transitions into the state $G(w \vdash \theta.x/a)$, where $\theta.x/a$ is the sentence one obtains by replacing all occurrences of x in θ with a .

Modalities. If $\psi = \Diamond \theta$ (or $\psi = \Box \theta$), then Verifier (Falsifier) picks another player; this player then picks a world w' accessible from w , and the game transitions into the state $G(w' \vdash \theta)$.

Negation. If $\psi = \neg \theta$, then the Verifier and Falsifier swap roles, and the game transitions into the state $G(w \vdash \theta)$.

Atoms. If ψ is an atomic sentence, the game terminates.

2 Applications

While I can only scratch the surface here, I want to point to some of the semantic phenomena that this system allows us to conceptualize. Consider for instance

- (2) Either we win tomorrow, or we don't.

This sentence is defensible in a polemic context, even when the speaker who utters it is not reliably able to choose correctly between the two logical branches *we win tomorrow* and *we don't win tomorrow*. The reason is that there might be a perfectly informed third person (“God”) which will be able to make the choice. Handing over the initiative to this omniscient observer makes for a weaker but still winning defense strategy.

More interesting cases occur if we assume that the two utility functions do not sum to 0. We then get new ambiguities like

- (3) The bald guy wanted either steak or lobster.

This can either mean that the speaker doesn't know what the bald guy wanted (handing *him* the right to choose branch) or that the bald guy has no preference (handing *the hearer* the right of choice).

Similar considerations explain notoriously “illogical” cases like

- (4) Every student answered some or all of the questions. [2]

- (5) You may take an apple or a pear. [3]

More examples exist, many of them related to ambiguities that have traditionally been treated by a number of incompatible approaches (pragmatic inference, grammatical exhaustivity operators, lexical ambiguity). A full treatment of these issues, however, is obviously not possible in the space of this abstract.

References

- [1] Maria Aloni and Robert Van Rooij. Free choice items and alternatives. In *Proceedings of the KNAW academy colloquium: Cognitive foundations of interpretation*, pages 5–26, 2007.
- [2] Gennaro Chierchia, Danny Fox, and Benjamin Spector. Hurford's constraint and the theory of scalar implicatures. In Paul Egré and Giorgio Magri, editors, *Presuppositions and Implicatures: Proceedings of the MIT-Paris Workshop*, number 60 in MIT Working Papers in Linguistics, 2009.
- [3] Michael Franke. Free choice from iterated best response. *Logic, Language and Meaning*, pages 295–304, 2010.
- [4] Esa Saarinen, editor. *Game-Theoretical Semantics: Essays on Semantics by Hintikka, Carlson, Peacocke, Rantala, and Saarinen*. Springer, 1979.

Henkin on Completeness

María Manzano*

Universidad de Salamanca

Enrique Alonso

Universidad Autónoma de Madrid

May 31, 2012

Abstract

The Completeness of Formal Systems is the title of the thesis that Henkin presented at Princeton in 1947, and his director was Alonzo Church. His renowned results on completeness for both *type theory* and *first order logic* are part of his thesis. It is interesting to note that he obtained the proof of completeness of first order logic readapting the argument found for the theory of types.

It is surprising that the first-order completeness proof that Henkin explained in class was not his own but was developed by using Herbrand's theorem and the completeness of propositional logic.

We conclude this paper by pointing two of the many influences of his completeness proofs, one is the completeness of *basic hybrid type theory* and the other is in correspondence theory, as developed in [6].

1 The completeness of FOL in Henkin's course

The story behind this is that of María Manzano, who during the academic year of 1977-1978 attended his class of *metamathematics* for doctorate students at Berkeley. Before each class Henkin would give us a text of some 4-5 pages that summarised what was to be addressed in the class. The texts were printed in purple ink, done with the old multcopiers that we called "*Vietnamese copiers*" and that were so often used to (illegally) print pamphlets in our past revolutionary days in Spain.

It is surprising that the first-order completeness proof that Henkin developed in class was not his own, but it was developed by using Herbrand's theorem and the completeness of propositional logic.

Theorem 1 (Herbrand's Theorem) *For each first-order sentence A there exist an (infinite) set of sentences of propositional logic Ψ such that: $\vdash A$ in*

*This research has been possible thanks to the research project sustained by **Ministerio de Ciencia e Innovación** of Spain with reference FFI _2009 _09345MICINN.

FOL iff there is some $H \in \Psi$ such that $\vdash H$ in LP (\vdash_{PL} means that we just use sentential axioms and detachment).

The above result can be regarded as a special case of the following.

Theorem 2 *Let L be a first order language: We can extend L to L' by adjoining a set \mathcal{C} of individual constants, and we can effectively give a set Δ of sentences of L' with the following property: For any set of sentences $\Gamma \cup \{A\} \subseteq \text{Sent}(L)$*

$$\Gamma \vdash A \text{ iff } \Gamma \cup \Delta \vdash_{PL} A$$

Predicate logic: Reduction to sentential logic: Using the previous theorem we effectively reduce the completeness problem for first order logic to that of sentential logic. To this effect the following proposition was proved.

Proposition 3 *Theorem 2 and completeness of PL implies completeness of FOL .*

2 His renowned proofs of completeness

The theorem of completeness establishes the correspondence between deductive calculus and semantics. Gödel had solved it positively for first-order logic and negatively for any logical system able to contain arithmetic. The lambda calculus for the theory of types [2], with the usual semantics over a standard hierarchy of types, was able to express arithmetic and hence could only be incomplete. Henkin showed that if the formulae were interpreted in a less rigid way, accepting other hierarchies of types that did not necessarily have to contain all the functions but at least the definable ones, it is easily seen that all consequences of a set of hypotheses are provable in the calculus. The valid formulae with this new semantics, called general semantics, are reduced to coincide with those generated by the rules of calculus.

As it is well known, Henkin's completeness theorem rests on the proof that every consistent set of formulas has a model.

Curiously, he obtained the proof of completeness for first-order logic in the second place, readapting the argument found for the theory of types. Another interesting aspect that Henkin himself pointed out is the non-constructive nature of the proof, despite coming from a tradition as tightly bound to proofs with a constructive nature as those developed by Church.

3 Two results based on Henkin's ideas

Let us highlight how Henkin's *General Models* are related to *Correspondence theory*. We attribute most of the ideas handled in the reduction to many-sorted logic [6] to two articles by Henkin: "*Completeness in the theory of types*" from 1950, and the one from 1953, "*Banishing the rule of substitution for functional*

variables". Nevertheless, with all the foregoing we do not wish to deceive possible readers. In the article from 1950, there are no translations of formulae, and the language and many-sorted calculus do not even appear explicitly. Regarding higher-order logic, as far as is known many-sorted calculus appears for the first time in the 1953 article. In it, Henkin proposes the axiom of comprehension as an alternative to the substitution rule used in the calculuses previously proposed for higher-order logic. If the axiom of comprehension is removed from this calculus, one obtains the MSL calculus. There is also another idea (this time from the 1953 article) that is also interesting and is as follows: If we weaken the axiom of comprehension (for example, we restrict it to first-order formulae or to translations of dynamic or modal formulae or to any other recursive set), we obtain calculuses in between MSL and SOL. And it is easy to find their corresponding semantics. Naturally, the class of structures corresponding to them will be situated in between \mathcal{F} (structures for MSL) and \mathcal{GS} (general structures). The new logic, let us call it XL, will also be complete. The reason is because this class of models is axiomatizable.

In [1] a *Basic Hybrid Type Theory* is introduced. The goal of this paper is to investigate whether basic hybridization also leads to simple Henkin-style completeness proofs in the setting of (classical) higher-order modal logic (that is, modal logics built over Church's simple theory of types [2]), and as we show in [1], the answer is "yes". The crucial idea is to use $@_i$ as a rigidifier for arbitrary types. We shall interpret $@_i\alpha_a$, where α_a is an expression of any type a , to be an expression of type a that rigidly returns the value that α_a receives at the i -world. As we show, this enables us to construct a description of the required model inside a single maximal consistent set and hence to prove (generalized) completeness for higher-order hybrid logic.

References

- [1] Areces, C., Blackburn, P., Huertas, A. and Manzano, M. "*Completeness in Hybrid Type Theory*". To appear.
- [2] Church, A. [1940]. "*A formulation of the simple theory of types*". **The Journal of Symbolic Logic**. vol. 5, pp. 56-68.
- [3] Henkin, L. [1949]. "*The completeness of the first order functional calculus*". **The Journal of Symbolic Logic**. vol. 14, pp. 159-166.
- [4] Henkin, L. [1950]. "*Completeness in the theory of types*". **The Journal of Symbolic Logic**. vol. 15. pp. 81-91.
- [5] Henkin, L. [1953]. "*Banishing the Rule of Substitution for Functional Variables*". **The Journal of Symbolic Logic**. 18(3): 201-208.
- [6] Manzano, M. [1996]. *Extensions of First Order Logic*. Cambridge Tracts in Theoretical Computer Science. Cambridge: Cambridge University Press.

All identicals are equal, but some equals are more equal than others

María Manzano*

`mara@usal.es`

Manuel Crescencio Moreno

`manuelcrescencio@usal.es`

Philosophy Department, University of Salamanca

Abstract

By the relation of identity we mean that binary relation which holds between any object and itself, and which fails to hold between any two distinct objects. In first order logic (FOL) it is impossible to define identity and so we are forced to introduce it as a logical primitive concept. In second order logic (SOL) with standard semantics identity is introduced by Leibniz's principle, but with non-standard models this principle does not apply. In there, we can introduce the symbol of equality defining an equivalence relation but there is no guarantee that this relation is prototypical identity.

In modal logic, we should decide whether we want the ontological relation of identity or the symbolic relation of equality between terms, or both. We are going to focus in the powerful language of type theory, where there are basically two sound alternatives: introducing both extensional and intensional terms or dealing with rigids and non rigids terms.

1 Equality and Identity in Classical Logic

Why do we take identity as a logical primitive concept in first order logic¹? Is there a formula φ defining it? The answer is negative, even in the best scenario where we only have a finite set of non-logical predicate constants. In this situation we can express that x and y cannot be distinguished in our formal language by defining a binary relation that obeys the usual rules for equality. But the definition is not semantically adequate as there are models where the relation defined by this formula is not identity. The formula is the nearest we can come up with in first order logic to formalize *Leibniz's principle of indiscernibles* saying that two objects are identical when there is no property able to distinguish them.

In SOL we can formulate Leibniz's principle by

$$x = y \leftrightarrow \forall X(Xx \rightarrow Xy)$$

*This research has been possible thanks to the research project sustained by **Ministerio de Ciencia e Innovación** of Spain with reference FFI.2009 _09345MICINN.

¹See [4] for a detailed treatment of this issue, in particular in *second order logic* and *type theory*.

thus introducing equality for individuals. We do not need reflexivity of equality and equals substitution as primitive inference rules, since they are already derived rules. This formula can be used to define identity for individuals as the relation defined by it is ‘genuine’ identity in any standard second order structure.

SOL with standard semantics has an extraordinary expressive power but poor logical properties and when you want to retain logical properties you need to introduce non-standard semantics. Then we are back to the situation encountered in first order logic. Within non-standard structures, the Leibniz’s principle define an equivalence relation but it could be different from identity. Therefore, if you want the prototypical identity, you should either have it as primitive, or define as well the concept of *non-standard normal structure*. All we have said should serve to warn you that the possibility of defining identity is lost as long as there is no guarantee of having all possible sets as denotation, specifically, all the singletons.

In general, identity for relations is neither introduced as a primitive logical symbol nor defined using the rest of the symbols in SOL. The main reason being that this identity established between relations is not a first order but a proper second order relation. Can we define identity in this context? Can we use Leibniz’s principle to introduce the identity for relations? The answer to both questions is negative, since to follow Leibniz’s pattern we would need third order variables.

The extensionality principle could be used to introduce equality

$$X^n = Y^n \leftrightarrow \forall x_1 \dots x_n (X^n x_1 \dots x_n \leftrightarrow Y^n x_1 \dots x_n)$$

Being the formal definition of the equality sign, the formula stops working as an extensionality principle.

In type theory, the Leibniz’s principle could be added to the set of equality axioms, as the set of instances of formulas of this form

$$\alpha = \beta \leftrightarrow \forall \gamma (\gamma(\alpha) \rightarrow \gamma(\beta))$$

where $=$ is of any type $\langle a, a \rangle$, for some a , α and β are of type a and γ is of type $\langle a \rangle$. As in SOL the formula defines identity only with standard semantics.

2 Identity and Equality in Modal Type Theory

In modal logic an “ontological” point of view can be taken, according to which the relation of equality we are interested in is pure identity. In this case, we must ensure that the calculus includes both the rule of *necessity of identity* (NI)

$$x = y \rightarrow \Box(x = y)$$

and the rule of *substitutivity of identicals* (SI)

$$x = y \rightarrow (\varphi(x) \rightarrow \varphi(y))$$

and both are sound. But these rules turn out to be problematic when terms other than variables are used. In this connection, the proposal of the authors Hughes and Cresswell [2] of narrowing the equality relation to extensional objects solves

the problem, while enabling constant and function symbols to be eliminated as they can be introduced by definition. As a matter of fact, the equality relation remains the identity one, but intensions are not taken into account. In case we wanted to deal with intensions, the authors propose to renounce to the (NI) principle and weaken (SI) to formulas free from modal operators. These systems in which (NI) does not hold and (SI) is weakened are called *contingent identity systems*.

In first-order modal logic, Melvin Fitting introduces a new relation between intensional terms $\tau_1 \approx \tau_2$ —an abbreviation for $\langle \lambda x, y. y = x \rangle(\tau_2, \tau_1)$ — for which (NI) does not hold. In fact, the formula $\tau_1 \approx \tau_2 \rightarrow \Box(\tau_1 \approx \tau_2)$, is not valid. Whereas $x = y$ expresses that the objects are the same and the relation defined by it can be taken as the identity relation; $\tau_1 \approx \tau_2$ says that the terms τ_1 and τ_2 designate the same object. In this sense, if we want to extend the rule (NI) for \approx , we would be expressing a notion wider than that of simple equality, namely it would have the characteristics of *synonymy*.

In modal type theory, Fitting adds equality extending Leibniz's principle to type theory. In this case the equality relation is not the prototypical identity as intensional expressions are allowed. Fitting [3] develops a novel approach to higher-order modal logic and uses it to investigate Gödel's ontological argument for the existence of God. Fitting's work has proved influential. But it is his semantic innovation which is likely to be enduring: the use of intensional models, a mechanism which makes it possible to avoid restrictions to rigid terms.

In [1] a basic hybrid type theory is introduced, adding nominals, a \Box , and the $@_i$ -operators to Henkin's original higher-order logic. Nominals are formulas of type t , which is the type of propositions, and are regarded as forming a distinct syntactic class. Nominals are true at a unique world in any model, thus a nominal i names the world it is true at. Moreover, note that for any expression α_a of any type a the result of prefixing it with $@_i$ (where i can be any nominal) yields an expression $@_i\alpha_a$ which is also of type a . We shall interpret $@_i\alpha_a$, where α_a is an expression of any type a , to be an expression of type a that rigidly returns the value that α_a receives at the i -world. Nominals and expressions of the form $@_i\alpha_a$ play a central role in the completeness result: nominals are the building blocks of our Henkin models, $@_i\alpha_a$ expressions supply the architectural blueprint.

In this setting *substitution rules* and *replacement* apply just for rigids, and the same happens for the rules of *Universal elimination* and *Beta conversion*.

References

- [1] Areces, C., Blackburn, P., Huertas, A. and Manzano, M. “*Completeness in Hybrid Type Theory*”. To appear.
- [2] Hughes, G.E. and Cresswell, M.J. [1996]. *A New Introduction to Modal Logic*. Routledge. London.
- [3] Fitting, M. [2002]. *Types, Tableaus and Gödel's God*. Kluwer Academic Publishers. Dordrecht.
- [4] Manzano, M. [1996]. *Extensions of First Order Logic*. Cambridge University Press. Cambridge.

The enriched effect calculus - a calculus for linear usage of effects

Rasmus Ejlers Møgelberg
IT University of Copenhagen, Denmark
(joint work with Jeff Egger, Alex Simpson and Sam Staton)

Abstract

The *enriched effect calculus* (EEC) is a variant of intuitionistic linear type theory suitable for reasoning about linear usage of computational effects. EEC can be seen as an extension of various calculi for computational effects (Moggi's monadic metalanguage or Levy's call-by-push-value) with constructions from linear logic such as a linear function space, but can also be viewed as a fragment of intuitionistic linear logic suitable for non-commutative effects.

We give two main examples of usage of EEC as target language of translations. The first is for the linear-use continuation passing translation, and the second is for the linear state passing translation. In both these cases we can show that the translations are fully complete in the sense that they induce a bijection between terms, but moreover we can show that these arise as very natural constructions on models. In fact, we shall see that these translations are dual in a formal sense.

This talk describes joint work with Jeff Egger, Alex Simpson and Sam Staton previously published in [1, 2, 5].

1 Introduction

Computational effects are the non-pure aspects of computations such as side-effects (reading and writing cells in memory), input and output, error raising, non-termination, non-determinism, and control effects. Moggi made the observation that all these can be described in a uniform way, by distinguishing between a type of values X and the type of computations $T(X)$ possibly returning an element in X , and noting that the resulting type constructor T satisfies the monad laws of category theory. For example, computation with side effects can be described using the monad $TX = S \rightarrow S \times X$ (a computation takes a current state and returns an updated state along with a value), computations with errors by the monad $TX = X + E$ (either a computation returns or it returns one of the possible errors in E) and control effects by the continuation monad $TX = (X \rightarrow R) \rightarrow R$, where R is the type of results.

This leads to the monadic metalanguage λ_{ML} , from which we recall the important typing rule

$$\frac{\Gamma, x: A \vdash t: TB \quad \Gamma \vdash u: TA}{\Gamma \vdash \text{let } x \text{ be } u \text{ in } t: TB} \quad (1)$$

which should be given the following computational interpretation: first run u then bind the result to x and then run t .

In this talk we are concerned with extending the monadic metalanguage with constructions from intuitionistic linear logic. In particular we would like to add a linear function space \multimap with the intuitive interpretation that $TX \multimap TY$ types functions $TX \rightarrow TY$ that run their input exactly once.

An obvious first candidate for such an extension would be dual intuitionistic/linear lambda calculus (DILL). Recall that DILL is given by the type grammar

$$A ::= A \otimes A \mid A \multimap A \mid !A \mid I$$

and recall the Girard encoding of intuitionistic function space $A \rightarrow B =_{\text{def}} !A \multimap B$. Since $!$ acts as a monad with respect to intuitionistic function space there is an obvious translation of λ_{ML} into DILL. Unfortunately, this translation introduces many undesirable equations, such as the commutativity rule

$$\text{let } !x \text{ be } u \text{ in } (\text{let } !y \text{ be } s \text{ in } t) = \text{let } !y \text{ be } s \text{ in } (\text{let } !x \text{ be } u \text{ in } t)$$

which states that the monad $!$ is commutative. Most monads considered in computer science are not commutative, because the order of execution of computations matters.

2 The enriched effect calculus (EEC)

The enriched effect calculus [1] solves this problem by restricting DILL to a fragment corresponding to non-commutative effects. The motivation for the fragment comes from the following theorem.

Theorem 2.1 *Let T be a commutative monad on \mathbf{Set} , and let \mathbf{Set}^T denote the Eilenberg-Moore category of T and let $F \dashv U : \mathbf{Set}^T \rightarrow \mathbf{Set}$ be the Eilenberg-Moore resolution of T as an adjunction. Then (\mathbf{Set}^T, FU) is a model of DILL.*

We review a few aspects of the model. A type A of DILL is interpreted as an object of \mathbf{Set}^T , i.e., as an algebra $\xi_A : T[[A]] \rightarrow [[A]]$ for the monad T , $[[!A]] = FU[[A]]$ and $[[A \multimap B]]$ is the set of algebra homomorphisms from $[[A]]$ to $[[B]]$, which carries an algebra structure because T is assumed commutative.

If T is not commutative, the collection of homomorphisms between two algebras does not carry an algebra structure and so if we are to generalise the above model construction, we are forced to consider two collections of types: computation types (to be modelled as algebras), and value types (to be modelled as sets). The enriched effect calculus has the following grammar of types:

$$\begin{aligned} A &::= \alpha \mid \underline{\alpha} \mid 1 \mid A \times B \mid A \rightarrow B \mid !A \mid \underline{A} \multimap \underline{B} \mid !A \otimes \underline{B} \mid \underline{0} \mid \underline{A} \oplus \underline{B} \\ \underline{A} &::= \underline{\alpha} \mid 1 \mid \underline{A} \times \underline{B} \mid \underline{A} \rightarrow \underline{B} \mid !A \mid !A \otimes \underline{B} \mid \underline{0} \mid \underline{A} \oplus \underline{B} . \end{aligned}$$

where we use the convention of underlining metavariables for computation types. We refer to [1] for full details on term judgements and the equational theory of EEC.

There is a translation of λ_{ML} into EEC translating all types as value types and translating T as $!$. The translation is *fully complete*, i.e., it adds no new equations and no terms to types in the image of the translation.

3 Linear continuation passing and linear state

EEC is intended as a language for reasoning about linear usage of resources in computation. The two main applications studied so far are linear usage of state [5] and linear usage of continuations [2].

The usual monad used for stateful computation $TX = S \rightarrow S \times X$ has the defect that it allows for operations such as the ‘snapback’ operation $TA \rightarrow TA$ which takes a computation, runs it, and then reinstates the state as it was before the computation ran. Real computations treat their state linearly. We have proved that the translation from λ_{ML} to EEC corresponding to the linear state monad $\underline{S} \multimap !A \otimes \underline{S}$ is fully complete [5].

This result derives from the following sequence of type isomorphisms known from linear type theory and also valid in EEC

$$!A \cong 1 \rightarrow !A \cong !1 \multimap !A \cong !1 \multimap !(A \times 1) \cong !1 \multimap !A \otimes !1$$

Taking \underline{S} to be $!1$ we see that $!A \cong \underline{S} \multimap !A \otimes \underline{S}$. That is, in EEC any monad is a linear state monad.

The CPS translation arises from the continuation monad $TA = (A \rightarrow R) \rightarrow R$. It is well-known that this translation is not fully complete. In fact, the translation is often used to introduce control effects via e.g. the call/cc operator (which, under the Curry-Howard isomorphism corresponds to Peirce’s law). Interestingly, the image of the translation can be described as those terms that use their continuation linearly, i.e., those that can be typed using the linear-use state monad $TA = (A \rightarrow \underline{R}) \multimap \underline{R}$. This result was proved for the resulting translation into DILL by Hasegawa [3], but when considered as a translation into EEC the result arises in a surprising way: we have proved that linear CPS translation from λ_{ML} to EEC extends along the inclusion of λ_{ML} to EEC to a translation from EEC to itself. Moreover, this extension is involutive, i.e., is its own inverse!

Semantically, the linear-use cps translation from EEC to EEC arises as a dual model construction related to Lawvere’s dual monad construction [4]. The involutivity of the translation can be derived from involutivity of the dual monad construction.

References

- [1] J. Egger, R. E. Møgelberg, and A. Simpson. Enriching an effect calculus with linear types. In *Proc. Computer Science Logic (CSL)*, volume 5771 of *LNCS*. Springer, 2009.
- [2] J. Egger, R. E. Møgelberg, and A. Simpson. Linearly-used continuations in the enriched effect calculus. In *Proc. Foundations of Software Science and Computation Structures (FoSSaCS)*, volume 6014 of *LNCS*. Springer, 2010.
- [3] M. Hasegawa. Linearly used effects: Monadic and CPS transformations into the linear lambda calculus. In *Proc. 6th International Symposium on Functional and Logic Programming (FLOPS)*, volume 2441 of *LNCS*, pages 167–182. Springer, 2002.
- [4] F. W. Lawvere. Ordinal sums and equational doctrines. In *Seminar on Triples and Categorical Homology Theory (ETH, Zürich)*, pages 141–155. Springer, 1969.
- [5] R. E. Møgelberg and S. Staton. Linearly-used state in models of call-by-value. In *Proc. 4th Conference on Algebra and Coalgebra in Computer Science (CALCO)*, volume 6859 of *LNCS*, pages 293–313. Springer, 2011.

New Logical Systems Based on Natural Language

Lawrence S. Moss

abstract for consideration by the Scandinavian Logic Society

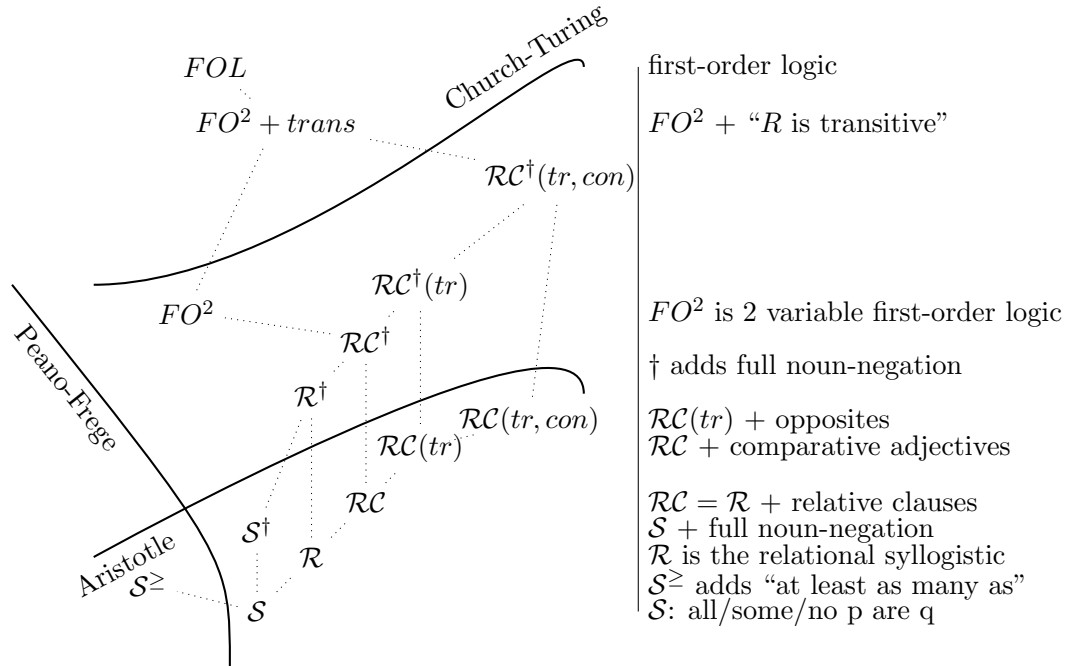
The past Logic is one of the oldest subjects in the western intellectual tradition, initiated by Aristotle and serving as a component of the medieval trivium. Around 1900, the field of logic became heavily mathematical, and indeed today there are whole fields of mathematics which are offshoots of logic. However, in this talk I am not so concerned with the interaction of logic and mathematics but rather with the more primary connection of *logic and language*.

The present I have been constructing logical systems which allow us to represent logical arguments in natural language, or something approximating it, directly, without translation to first-order logic. Indeed, the goal is to work as close to the “surface” forms as possible and to have decidable and complete logical systems. The chart on the next page is a map of the prominent systems. Listed a number of tiny pieces of English with names like \mathcal{S} , \mathcal{R} , etc. The line marked “Aristotle” separates the logical systems above the line, systems which can be profitably studied on their own terms without devices like variables over individuals, from those which cannot. For example, the system \mathcal{S} is Aristotle’s syllogistic; it contains sentences such as *All men are mortals* and *Some woman is a senator*. The system \mathcal{R} extends this to a bigger fragment containing verbs. (“ \mathcal{R} ” stands for *relation*.) So \mathcal{R} would contain *Some dogs chase no cats*. The yet larger system \mathcal{RC} would contain *All who love all animals love all cats*. To see that logic in this system is not exactly trivial, verify that in the most natural semantics

All insects are animals \models *All who fear all who love all insects fear all who love all animals*

The languages in the chart with the dagger such as \mathcal{S}^\dagger and \mathcal{R}^\dagger are further enrichments which allow subject nouns to be negated. This is rather un-natural in standard speech, but it would be exemplified in sentences like *Every non-dog runs*. The point: the dagger fragments are beyond the *Aristotle boundary* in the sense that they cannot be treated by the relatively simpler syllogistic logics. The only known logical systems for them use *variables* in a key way. For example, here is a derivation in $\mathcal{RC}^\dagger(trans)$:

$$\begin{array}{c}
 \frac{\frac{\frac{[\text{bigger}(x, y)]^2}{\text{bigger}(x, z)} \quad \frac{\frac{[\text{kumquat}(z)]^1}{\text{bigger}(y, z)} \quad \frac{[\text{sweet}(y)]^2 \quad \forall(\text{sweet}, \forall(\text{kumquat}, \text{bigger}))}{\forall(\text{kumquat}, \text{bigger})(y)} \quad \forall E}{\forall(\text{kumquat}, \text{bigger})(y)} \quad \forall E}{\text{bigger}(y, z)} \quad \text{trans}}{\forall(\text{kumquat}, \text{bigger})(x)} \quad \forall I^1 \\
 \frac{\frac{[\exists(\text{sweet}, \text{bigger})(x)]^3}{\forall(\text{kumquat}, \text{bigger})(x)} \quad \forall I^2}{\forall(\exists(\text{sweet}, \text{bigger}), \forall(\text{kumquat}, \text{bigger}))} \quad \forall I^3
 \end{array}$$



It corresponds to an inference: assuming that all sweet fruit are bigger than all kumquats, everything which is bigger than some sweet fruit is bigger than all kumquats. The logical system for this language is a natural deduction system. It comes from [6]; an early paper in this direction is Fitch [1].

The biggest language in the chart is $RC^\dagger(tr, con)$. It has transitive verbs, relative clauses, comparative adjective phrases (like *bigger than*) and their converses (*smaller than*). This is already a fairly big fragment. From a logical perspective, one interesting fact about it is that it does not lie in FO^2 , the 2-variable fragment of first-order logic. However, the validity problem for it and all other the systems below the "Church-Turing" boundary are all *decidable*.

The web site <http://logicforlanguage.blogspot.com/> contains more on the topic of natural logic, including a set of lecture notes and accompanying pdf files of slides from a course. One can also see the reference mentioned at the end of this submission.

Specific contribution The talk at SLS will include a discussion of Ivanov and Vakarelov's paper [3]. This paper presents a logical system which goes beyond $RC^\dagger(con)$ in having a full relational repertoire, including *boolean combinations* of binary relations, and also a primitive construct for the *converse of a relation*. In their language, one can directly represent inferences such as

There is some dog who chased every cat \vdash *Every cat was chased and bit by some dog*

with the second sentence interpreted using the subject wide scope. Their system is not a natural deduction system with individual variables, but it uses a method coming from modal

logic; specifically, from the region-based theory of space. My contribution is to re-work their logic along the lines of my earlier paper [6]. There are reasons for doing this re-working. In addition, I will treat comparative adjectives interpreted by transitive relations, as in the “kumquat” example above. The importance of the resulting logic is that it is the *largest known decidable logic for representing linguistic inference*.

The talk (and the area of natural logic) makes connections to algebraic logic (the proof techniques for completeness borrow from it), computational complexity theory (see [7]), proof theory [2], modal logic (via translation to boolean modal logics), the history of logic, and linguistic semantics. The overall point that linguistic inference should be captured in decidable logics is a controversial one that never fails to stimulate a discussion.

Future This talk presents natural logic as a developer of new systems of logic with an eye towards natural language. In a sense, the project goes back to Aristotle’s syllogisms. However, we have in mind a much broader and deeper set of research questions: whereas syllogistic reasoning only applies to arguments in very restricted forms, we would eventually like to study reasoning as close to “surface forms” (real sentences) as possible. Instead of limiting attention to categorical (yes/no) reasoning, we eventually hope to develop connections to default reasoning and reasoning under uncertainty. Indeed, we feel that natural logic should be connected to the field of *textual inference*, where one attempts to automatically infer the meaning of texts and to answer questions. In a sense, natural logic would like to present a version of natural language semantics based on computational linguistics, rather than the tools coming from the logical exploration of mathematics. These topics, however, are for the future.

References

- [1] Frederic B. Fitch. Natural Deduction Rules for English. *Philosophical Studies*, 24:2 (1973), 89–104.
- [2] Nissim Francez and Roy Dyckhoff. Proof-Theoretic Semantics for a Natural Language Fragment. To appear in the *Journal of Logic, Language, and Information*.
- [3] Nikolay Ivanov and Dimiter Vakarelov. A System of Relational Syllogistic Incorporating Full Boolean Reasoning, ms. 2011.
- [4] Lawrence S. Moss. Completeness Theorems for Syllogistic Fragments. In F. Hamm and S. Kepser (eds.) *Logics for Linguistic Structures*, Mouton de Gruyter, 2008, 143–173.
- [5] Lawrence S. Moss. Syllogistic Logics with Verbs, *Journal of Logic and Computation*, special issue on papers from Order, Algebra and Logics, Vol. 20, No. 4, 2010, 947–967.
- [6] Lawrence S. Moss. Logics for Two Fragments Beyond the Syllogistic Boundary, in *Fields of Logic and Computation: Essays Dedicated to Yuri Gurevich on the Occasion of His 70th Birthday*, A. Blass, N. Dershowitz, and W. Reisig (eds.), LNCS, vol. 6300, Springer-Verlag, 2010, 538–563.
- [7] Ian Pratt-Hartmann and Lawrence S. Moss. Logics for the Relational Syllogistic. *Review of Symbolic Logic* Vol. 2, No. 3, 2009, 647–683.

**A.N. Prior's Contributions to the Rise of Temporal Logic
in the 1950s and 1960s:
New Insights Based on the Study of Prior's Nachlass**

Abstract

Peter Øhrstrøm

Aalborg University

E-mail: poe@hum.aau.dk

Arthur Norman Prior (1914-69) must be said to be the founder of modern tense-logic. He revived the medieval attempt at formulating a temporal logic for natural language. Therefore his work also established a paradigm applicable to the exact study of the logic of natural language. Prior held that logic should be related as closely as possible to intuitions embodied in everyday discourse, and his tense logic can indeed account for a large number of linguistic inferences. In the 1950s and 1960s he laid out the foundation of tense-logic and showed that this important discipline was intimately connected with modal logic. Prior also argued that temporal logic is fundamental for understanding and describing the world in which we live. He regarded tense and modal logic as particularly relevant to a number of important existential and theological problems. Using his temporal logic, Prior analysed the fundamental question of determinism versus freedom of choice.

A.N. Prior wife, Mary Prior, has described the first occurrence of this idea of temporal: "I remember his waking me one night, coming and sitting on my bed, and reading a footnote from John Findlay's article on Time, and saying he thought one could make a formalised tense logic." This must have been some time in 1953. Findlay's considerations on the relation between time and logic in this footnote were not very elaborated, but it gave Prior the idea of developing a formal calculus which would capture this relation in detail. For this reason Prior called Findlay "the founding father of modern tense logic". But there are, in our opinion, certainly not sufficient reasons for viewing Findlay as the founder of tense logic. The honour of being the founder must without doubt be attributed to Prior himself. With his many articles and books on questions in tense logic he presented a very extensive and thorough corpus, which still forms the basis of tense logic as a discipline. Findlay's major merit in tense logic is to have had the luck of inspiring Prior to initiate the development of formal tense logic.

During the 1940s and the 1950s Prior often worked on questions in the history of logic. From 1952 to 1955 he had seven articles on the history of logic published. In particular, he became very interested in the Master Argument of Diodorus Kronos and the so-called Diodorean logic, primarily Diodorus' definition of implication. Prior seemed to realise that it might be possible to relate

Diodorus' ideas to contemporary works on modality by developing a calculus which included temporal operators analogous to the operators of modal logic.

A.N. Prior spent 1956 in Oxford, where he had been invited to give the 'John Locke lectures' of that year. These lectures formed the basis of Prior's book *Time and Modality* (published in 1957), the first work in which Prior's logic of time and modality was presented systematically.

The reactions on *Time and Modality* led to the development of the branching time models, first suggested by Saul Kripke. Prior later developed the idea further taking Ockhamistic and Peircean ideas into account.

The Handbook of the History of Logic, vol. 7 [Gabbay and Woods 2006] includes a chapter on "A.N. Prior's logic" [Øhrstrøm and Hasle 2006a] as well as a chapter entitled "Modern Temporal Logic: The Philosophical Background" [Øhrstrøm and Hasle 2006 b]. These chapters offer a general account of the rise of temporal logic. However, further studies of the sources have led to significant new insights regarding the development of temporal logic. In particular, the systematic study of Prior's Nachlass (including his correspondence) kept at the Bodleian Library in Oxford has led to a deeper understanding of how and why temporal logic was developed.

In this paper I shall concentrate on the following topic:

- The early studies of Diodorean logic
- Prior's preparation for the 'John Locke lectures'
- The early discussions on the idea of branching time
- Prior's further development of branching time models using Ockhamistic and Peircean ideas

It will be argued that the study of Prior's Nachlass can provide new insights regarding these topics.

References:

- Gabbay, D. M. and Woods, J. (Eds.) : 2006, *The Handbook of the History of Logic*, vol. 7, North-Holland .
- Øhrstrøm, P. and Hasle P. : 2006a, "A.N. Prior's logic" , in [Gabbay and Woods: 399-446].
- Øhrstrøm, P. and Hasle P. : 2006b, "Modern temporal logic: The philosophical background", in [Gabbay and Woods, 447-498] .

Mathematics and Logic

Ulf Persson
professor of mathematics at
Chalmers University of Technology
Göteborg, Sweden

Mathematical Platonism is intellectually indefensible and psychologically inescapable according to the Russian mathematician Yuri Manin ([5]). The issue is of course metaphysical and can only be approached in a philosophical vein. There is of course a large body of work addressing the subject, and it is not my aim to give a survey, instead I would like to present a personal and idiosyncratic view by virtue of my profession as a mathematician, especially when it comes to the relation between mathematics and logic, where my views resonate with those of C.S.Peirce ([6]).

Platonism is nowadays disparaged as a hopelessly outmoded view, but in my opinion, the arguments against it, especially if phrased in some formal way, as with Benacerraf ([1]) strike me as silly. Obviously you are as unable to argue against Platonism in a strict mathematical way, as you are unable to argue for it by deductive mathematical arguments. It is a matter of faith, and thus to many a source of comfort that many influential mathematicians have supported it ([7]). Although it is assumed that mathematicians are Platonists when doing mathematics, they tend to be divided when it comes to a public stand, as illustrated by a recent debate ([2]).

In mathematical philosophy Platonism is often contrasted against intuitionism, formalism and nominalism. There are of course other modern approaches based on evolution and thus in practical terms on psychology (something that scandalized Frege ([3])). To me this bespeaks a confusion of categories, obviously those approaches are very different in character and have a high degree of overlap. This has made me question whether mathematical philosophers are qualified to expound philosophically on mathematics.

To a mathematician what may shake his or her belief in the timeless Platonic character of mathematics are the higher hierarchies of cardinals, which never enter into serious mathematics. Could it be that set-theory is merely a language for mathematics and that it by itself contains few, if any, serious mathematical problems?([4])

There are of course many variants of Platonism, and the debate is somewhat muddled because it is not always made clear that there is a distinction between the human practice of mathematics and mathematics itself, although I admit that proposing that distinction, by itself presupposes some platonistic persuasions.

I hope that my talk will be somewhat provocative.

References

- [1] Benacerraf, Paul, 1973, Mathematical Truth, *Journal of Philosophy*, 70(19):
- [2] Davies, Brian; Gardner, Martin; Hersh, Reuben; Mazur, Barry; Mumford David 2007-08, in subsequent issues of the Newsletter of the European Mathematical Society.
- [3] Frege, Gottlob, 1953, *Foundations of Arithmetic*, Oxford: Blackwell. Transl. by J.L. Austin.
- [4] Gödel, Kurt, 1964, What is Cantor's Continuum Hypothesis?, In Benacerraf and Putnam *Philosophy of Mathematics: Selected Readings*, Cambridge: Cambridge University Press. Second edition.(1983)
- [5] Manin, Yuri 2010, book review of J.Gray's 'Plato's Ghost' in the Notices of the American Mathematical Society
- [6] Peirce, Charles Sanders, 1940,55. The Nature of Mathematics in "Philosophical Writings of Peirce, selected and ed by J.Buchler. Dover Publications
- [7] Penrose, Roger, 1994, 'Shadows of the Mind", Oxford University Press

Extended Abstract: Qualitative Modeling of Informational Cascades

Rasmus K. Rendsvig

No Institute Given

Humans are social beings and as such, decisions and actions include and affect the decisions and actions of others. This happens both when deliberating groups try to reach consensus via information exchange and when a singular agent acts within a group of interacting agents. A prime example of the latter is the robust social phenomena known from behavioral economics, *informational cascades*. Informational cascades are common. As a classic example goes, we choose to dine at what seems to be the most crowded restaurant [3]. The restaurant is chosen because others before us seem to vouch for its quality by their choice, but in doing so, we help to insure it will have filled tables. Thereby, we again guide others in their choice for supper. Informational cascades may occur in situations where an agent has to choose between a number of alternatives without a strong individual preference. If the earlier choices of others are made available to the agent, the agent may conjecture that these were made on an informed basis, and choose accordingly. An informational cascade occurs when the choosing agent's initial belief are "overridden" by the information extrapolated from the choices of previous actors. The term 'informational cascade' was coined in the 1992 [5], and has been deemed at least a partial reason for college binge drinking [4], technology adoption and "hot" academic topics [3].

The Structure of Informational Cascades. In general terms, the structure underlying informational cascades consists of 1) a set of rational agents that act sequentially, 2) a set of options between which the agents can choose, and 3) a preference order on the outcome of each choice.

In the literature on informational cascades, agents are typically modeled as Bayesian maximizers of expected utility. The decision is made under uncertainty in the sense that no agent knows which action leads to the jointly preferred outcome. That there is a jointly preferred outcome is essential when it comes to the epistemic assumptions made. There is no strategic interaction in the decision problem, so no agent will have an incentive to mislead later agents by choosing in contrary to the best of their knowledge. This in turn means that subsequent agents may base their decision not only on their private information, but also on the action of those that act before them. Specifically, the following epistemic assumptions are in order:

1. The underlying structure is known to all agents,
2. Each agent makes a rational decision based on available information, which consists of
 - (a) A private signal about which action will lead to which outcome, which is known to be more often right than it is wrong.
 - (b) A public signal consisting of the string of actions performed by the previous agents,

3. Knowledge among the agents that their signals are equally likely to be correct, and
4. Knowledge of rationality as described in 2.

Notice that in b. it is only the actions, and not the signals, of previous agents that can be observed. Furthermore, one should notice that the sequence of agents is known to all is in conjunction with b. taken to imply that any agent knows what public signal any previous agent received.

A run of such a model may be conceived as a line of agents, each waiting to make a decision between a (finite) set of choices. In runs where later agents choose to ignore their private information and act on the information conveyed by previous agents' actions, an informational cascade is said to be in effect.

Illustration: Initiating a Cascade. To illustrate, consider a situation where the agents have to make a binary choice between turning left or turning right at a junction in a maze – or just de-boarding a plane. Before receiving their private signal of *left* or *right*, each agent will be indifferent between the two options. When the first agent receives her private signal, say *left*, she will take this to indicate the correct path out of the maze. Given that she has no further information available, she will follow her private signal, thereby conveying a *left* action to all subsequent agents.

When the second agent must choose, the public signal of an executed *left* action in conjunction with knowledge of rationality may be used to deduce that the first agent's signal was *left*. Two situations may now have occurred: one in which the second agent received the private signal *left*, in which case she should choose to go *left*, or one where she received private signal *right*, in which case her available information – a *left* signal from agent 1 and a *right* signal from herself – will suggest opposite responses. Since both signals are known by 2 to be equally likely to be correct, rationality specifies no concrete plan of action. Hence the agent must choose based on some *tie-breaking rule*, e.g., by randomizing, choosing to follow her private signal, etc. For now, assume that the second agent received a *left* signal, and therefore chooses to go left.

The actions of agents 1 and 2 send a public (*left, left*) signal to agent 3. As agent 2, 3 can deduce the private signal of agent 1. Additionally, given suitable assumptions regarding the tie-breaking rule, 3 may also deduce that agent 2 received a *left* signal. As it is known that every private signal is equally likely to be correct, it now does not matter for her action what signal she herself received. If 3 received a *left* signal, she, too, should choose to go left. If she received a *right* signal, the information extrapolated from the public (*left, left*) signal results in *left* still being more probable than *right*. She will therefore choose to ignore her private information and act in accordance with the group behavior. Thereby agent 3 will be the first agent in an informational cascade.

Upon receiving the (*left, left, left*) action string, agent 4 will also choose to ignore her private signal in case this is *right*, and choose to go left. This action will be chosen on the same basis as 3 made her choice, namely the deduction of the private signals of agents 1 and 2. The fourth agent will, however, *not* have

a stronger reason to go left than agent 3 had, since the choice made by agent 3 is uninformative to all subsequent agents. This is a corollary of agent 3 being in cascade: since 4 knows that 3 is rational and received the public signal (*left, left*), 4 can deduce that 3 would have chosen to go left no matter what private signal she received. Hence, agent 4 will base her decision only on the choices of the two first agents, and will also be in cascade. Similar considerations apply to all subsequent agents: they will all be in the cascade, ignoring both their private information, as well as the choices made by previous agents in the cascade.

Previous Works and Our Contribution. As mentioned above, the literature on information cascades typically models agents as Bayesian. This was done in the original papers on the topic, [5,3], as well as in later refinements, e.g. [1,6]. In these treatments, however, a large part of the reasoning of the agents are excluded from the model. Most notably, the way the agents deduce other agents’ signals and their informativeness is not modeled. To remedy this lack, we instead suggest a model of informational cascades where the belief revision steps are explicated using dynamic doxastic-epistemic logic and *multi-agent plausibility models*, cf. [2]. We define a majority-based belief merge operation for groups which can be shown to be ‘equivalent’ with Bayesian updating in the informational cascades case. This operation is used by agents when making their choice, as well as allow them to reason about the belief construction of previous agents. As a result, the informal reasoning gone through above may be re-produced in a formal setting. This has the advantage of forcing the used epistemic assumptions to be made explicit. It can be shown under which conditions certain tie-breaking rules assumed by previous authors produce cascades: some require common knowledge, whereas other requires an assumption of *pluralistic ignorance*. We conclude by evaluating strategies for dissolving informational cascades that have converged to the “wrong” option using public and private announcements.

References

1. L.R. Anderson and C.A. Holt. Information Cascades in the Laboratory. *The American Economic Review*, 87:847–862, 1997.
2. Alexandru Baltag and Sonja Smets. A Qualitative Theory of Dynamic Interactive Belief Revision. In G. Bonanno, W. van der Hoek, and M. Wooldridge, editors, *Texts in Logic and Games*, volume 3, pages 9–58. Amsterdam University Press, 2008.
3. Abhijit V. Banerjee. A Simple Model of Herd Behavior. *The Quarterly Journal of Economics*, 107(3):797–817, 1992.
4. Cristina Bicchieri and Yoshitaka Fukui. The Great Illusion: Ignorance, Informational Cascades, and the Persistence of Unpopular Norms. In M.C. Galavotti and A. Pagnini, editors, *Experience, Reality, and Scientific Explanation*, pages 89–121. Kluwer Academic Publishers, 1999.
5. Sushil Bikhchandani, David Hirshleifer, and Ivo Welch. A Theory of Fads, Fashion, Custom, and Cultural Change as Informational Cascades. *Journal of Political Economy*, 100(5):992–1026, 1992.
6. Lones Smith and Peter Norman Sørensen. Observational Learning. *Econometrica*, 68(2):371–398, 2000.

Two-valued semantics for Łukasiewicz's 3-valued logic L3

Gemma Robles

Łukasiewicz's three-valued logic L3 was defined in a two pages paper in 1920 (cf. [6]). The philosophical motivation of the proposed logic and the interpretation of the third truth value are clearly stated: "The indeterministic philosophy [...] is the metaphysical substratum of the new logic" ([6], p. 88). "The third logical value may be interpreted as "possibility"" ([6], p.87). L3, understood as the set of all three valued formulas, was first axiomatized by M. Wajsberg in 1931 (see [14]). Since then, different axiomatizations of the finite-valued L_n -logics have been proposed. Among those in a Hilbert-style form, the ones by Tokarz and Tuziak are to be remarked (see [10], [11]). Tokarz axiomatizes L_n by adding one-variable axioms to L_ω as axiomatized by Wajsberg. Tuziak, on the other hand, presents particular axiomatic systems for each n by generalizing a completeness theorem by Pogorzelski and Wojtilak. Concerning the axiomatization of L3, Avron provides a simple one which is, in addition, a "well-axiomatization" (cf. [1]). We, on our part, will define a simple axiomatization of L3 by extending with independent axioms Routley and Meyer's basic positive logic B_+ (cf. [8]). On Łukasiewicz logics and many-valued logic in general, cf., e.g., [7], [11].

Now, let us precisely state what "Łukasiewicz's three-valued logic L3" refers to in this paper.

As it is well known, a logic is, according to the Polish logical tradition, equivalent to a consequence relation of some kind, provided it fulfills Tarski's standard conditions. As Wojcicki puts it: "a logic is defined by its derivability relation rather than by its sets of theorems" ([15], p. 202). In this sense, there are essentially (but not exclusively), two ways of defining a (semantical) consequence relation given a set of truth-values (i.e., a set with an ordering relation \leq) and a set of functions F from the set of wffs to v :

1. For any set of wffs X and wff A , $X \models A$ iff for all $f \in F$, if $f(X) \in D$, then $f(A) \in D$ (D is the set of designated values).
2. For any set of wffs X and wff A , $X \models A$ iff for all $f \in F$, $f(X) \leq f(A)$ [$f(X) = \inf \{f(B) : B \in X\}$].

It is commonly understood that Łukasiewicz logics are those determined by the relation defined in (1). But, as Wojcicki remarks, ([16], §13), the Łukasiewicz logics determined by the relation defined in (2) also deserve to

be called Łukasiewicz logics ([16], p.42). Actually, the referred author concludes that there are at least two kind of Łukasiewicz logics: *truth-preserving Łukasiewicz logics* (determined by the relation defined in (1)) and *well-determined Łukasiewicz logics* (determined by the relation defined in (2)). Consequently, Łukasiewicz's three-valued logic L3 is here understood in three different senses:

- i. As the set of three-valued valid formulas according to the matrices ML3 defined by Łukasiewicz.
- ii. As truth-preserving three-valued L3 (determined by the relation (1) in ML3).
- iii. As well-determined three-valued L3 (determined by the relation (2) in ML3).

The aim of this paper is to provide a two-valued semantics for the three versions of L3 recorded above with (equivalently) under-determined or over-determined interpretations. An *under-determined interpretation* is a function from sentences to the proper subsets of the set $\{T, F\}$; and an *over-determined interpretation* is a function from sentences to a non-empty subset of $\{T, F\}$ (T and F represent truth and falsity in the classical sense). Thus, under-determined interpretations assign T , F or neither to sentences; and over-determined interpretations assign T , F or both. It will be shown that L3 (in the sense (i)) can be dually interpreted either by under-determined or else over-determined interpretations. Then, consequence relations equivalent to those defined in (1) and in (2) (referred, of course, to the matrices ML3) will be defined by under(over)-determined interpretations. A consequence of these results is that Łukasiewicz's third-value can legitimately be thought of as equivalently representing either indefiniteness or else contradictoriness.

The two-valued semantics with either "gaps" or "gluts" here defined is based on Dunn's semantics for first degree entailments (see [4], [5]), that goes back to Dunn's doctoral dissertation (see [3]). As noted by Dunn himself ([4], p.150) essentially equivalent semantics are defined in [9] and [13].

On the other hand, L3 is axiomatized as an extension of B_+ following Brady's strategy for axiomatizing many-valued logics by employing two-valued under-determined and/or over-determined interpretations (cf. [2]).

Acknowledgements

-Work supported by research project FFI2011-28494 financed by the Spanish Ministry of Economy and Competitiveness.

-G. Robles is supported by Program Ramón y Cajal of the Spanish Ministry of Economy and Competitiveness.

Dpto. de Psicología, Sociología y Filosofía, Universidad de León
Campus de Vegazana, s/n, 24071, León, Spain
<http://grobv.unileon.es>
gemmarobles@gmail.com

References

- [1] Avron, A. (1991). Natural 3-Valued Logics. Characterization and Proof Theory. *Journal of Symbolic Logic*, 56, 276-294.
- [2] Brady, R. (1982). Completeness Proofs for the Systems RM3 and BN4. *Logique et Analyse*, 9-32.
- [3] Dunn, J. M. (1966). *The Algebra of Intensional Logics*. Doctoral dissertation, University of Pittsburg. (Ann Arbor, University Microfilms).
- [4] Dunn, J. M. (1976). Intuitive semantics for first-degree entailments and coupled trees. *Philosophical Studies*, 29, 149-168.
- [5] Dunn, J. M. (2000). Partiality and its Dual. *Studia Logica*, 65, 5-40.
- [6] Łukasiewicz, J. (1920). On three-valued logic. In J. Łukasiewicz (1970), *Selected Works* (North Holland, Amsterdam), 87-88.
- [7] Malinowski, G. (2007). Many-valued logic and its philosophy. *The many valued and nonmonotonic turn in logic*, Handbook of the History of Logic, 8, Elsevier/North-Holland, Amsterdam, 13-94.
- [8] Routley, R., Meyer, R. K. (1972). Semantics of Entailment III. *Journal of Philosophical Logic*, 1, 192-208.
- [9] Routley, R., Routley, V. (1972). Semantics of first-degree entailment. *Noûs*, 1, 335-359.
- [10] Tokarz, M. (1974). An method of axiomatization of Łukasiewicz logics. *Studia Logica*, 33, 333-338.
- [11] Tuziak, R. (1988). An axiomatization of the finite-valued Łukasiewicz calculus. *Studia Logica*, 47, 49-55.
- [12] Urquhart, A. (2002). Basic Many-Valued Logic. In D. Gabbay and F. Guenther (Eds.), *Handbook of Philosophical Logic*, vol. 2, Springer, 249-295.
- [13] Van Fraassen, B. (1969). Facts and Tautological Entailments. *The Journal of Philosophy*, 67, 477-487.
- [14] Wajsberg, M. (1931). Axiomatization of the 3-valued propositional calculus. In S. McCall (1967), *Polish Logic: 1920-1939* (Oxford University Press), 264-284.
- [15] Wojcicki, R. (1974). The Logics Stronger than Łukasiewicz's Three-Valued Sentential Calculus - The Notion of Degree of Maximality versus the Notion of Degree of Completeness. *Studia Logica*, 33, 201-214.
- [16] Wojcicki, R. (1984). Lectures on Propositional Calculi. *Ossolineum*.

Dynamic Consequences for Soft Information

Extended Abstract for submission to the *Scandinavian Logic Symposium*

Olivier Roy and Ole Thomassen Hjortland

Munich Center for Mathematical Philosophy
LMU, Munich

This paper studies the so-called *dynamic consequence relation* generated by lexicographic updates in dynamic epistemic logic (DEL). Our goal is to provide a sound and complete sequent calculus for this consequence relation, to study its proof-theoretic properties, and to relate it to known sub-structural proof systems.

This extended abstract presents the main ideas behind this project, and states some of the preliminary results already obtained. The main questions remain open, though. Our goal is to present and discuss them with the audience of the *Scandinavian Logic Symposium*.

Dynamics of Soft Information

Dynamic epistemic logic (DEL) is a general logical framework to study dynamic information changes in social situations—see [15] and [11] for an introduction. In its most common form, the gist of DEL is to enrich propositional modal languages with “dynamic operators” that describe the effects or consequences of certain “epistemic actions” in a given situation. Epistemic actions are events that only affect the information available in a given situation, for instance observing certain states of affairs or learning about them from a truthful and trusted source.¹

Here we are interested in the dynamics of so-called *soft* informational attitudes. These are attitudes, like beliefs, which are revisable, might be mistaken, and might not be fully introspective. Our starting point is thus a static modal language expressive enough to encode *conditional beliefs*, i.e. modalities of the form

$B_i^\phi \psi$, to be read “conditional on ϕ , agent i believes that ψ .” Mainly for technical reasons, we work in the following language \mathcal{L}_S , in which conditional beliefs are definable [2].²

$$\phi := p \mid \neg\phi \mid \phi \wedge \phi \mid [\sim]_i\phi \mid [\leq]_i\phi$$

Formulas of the form $[\sim]_i\phi$ and $[\leq]_i\phi$ should be read, respectively, as “agent i knows that ϕ ” and “agent i safely believes that ϕ .” We write $\langle\sim\rangle_i\phi$ for $\neg[\sim]_i\neg\phi$ and $\langle\leq\rangle_i\phi$ for $\neg[\leq]_i\neg\phi$. Conditional beliefs $B_i^\phi\psi$ are then defined as follows:

$$B_i^\phi\psi \Leftrightarrow_{df} \langle\sim\rangle_i\phi \rightarrow \langle\sim\rangle_i(\phi \wedge [\leq]_i(\phi \rightarrow \psi))$$

This language is interpreted in *epistemic-plausibility models*, which are Kripke structures [4] equipped with a collection of partial pre-orders \leq_i and a valuation V assigning to each state $w \in W$ a subset of a given set of atomic propositions. The truth conditions for the modalities then run as follows—see again [2] for details:

- $\mathcal{M}, w \models [\leq]_i\phi$ iff $\mathcal{M}, w \models \phi$ for all $w' \leq_i w$.
- $\mathcal{M}, w \models [\sim]_i\phi$ iff $\mathcal{M}, w \models \phi$ for all w' such that either $w' \leq_i w$ or $w \leq_i w'$.

Just like attitudes, epistemic action can also be “soft”. These are informational events that are reversible, not necessarily public nor truthful. A number of such soft epistemic actions have been studied in the literature, for instance the so-called radical and conservative upgrade mechanisms—see e.g. [11].

Here we work in an extension of general dynamic epistemic logic developed in [2], where soft epistemic

¹Epistemic actions are usually distinguished from “ontic events”, i.e. events, like turning on the lights, that change some non-informational facts. Ontic events have also been studied in DEL, c.f. [12], but we bracket them here.

²All along we assume a finite set N of agents and denote its elements i, j , etc.

actions are encoded in so-called *event models* and update with such actions are computed using a *lexicographic update rule*. Roughly, an event model is a Kripke structure equipped with a collection of pre-orders, where that the elements of its domain are thought of as basic events or programs³ and instead of the standard valuation each event e gets assigned a pre-condition, written $pre(e)$, described by a formula of the static language defined above. The lexicographic update rule then takes pairs of epistemic-plausibility and event models \mathcal{M}, w and \mathcal{E}, e and return the updated model $\mathcal{M} \otimes \mathcal{E}$ where the domain is the set of pairs (w, e) such that \mathcal{M}, w satisfies the pre-condition of e , written $\mathcal{M}, w \models pre(e)$ and the valuation is taken directly from \mathcal{M} , i.e $V'(w, e) = V(w)$. The adjective “lexicographic” comes from the update rule for the pre-orders \leq_i , which gives priority to the events:

$$(w', e') \leq'_i (w, e) \text{ iff either } e' <_i e \text{ or } e \cong_i e' \text{ and } w' \leq_i w.$$

With this in hand the static language \mathcal{L}_S is usually extended with modalities of the form $[\mathcal{E}, e]\phi$, to be interpreted as follows:

- $\mathcal{M}, w \models [\mathcal{E}, e]\phi$ iff if $\mathcal{M}, w \models pre(e)$ then $\mathcal{M} \otimes \mathcal{E}, (w, e) \models \phi$.

This dynamic language has been shown in [2] to be expressive enough to capture all soft update operations that have been so far studied in the DEL literature.

Dynamic Consequence: Idea and Motivations

Peter Gärdenfors once said that belief change and (nonmonotonic) reasoning are “two sides of the same coin” [6]. This idea, by now widely accepted in default logic, also underlies the notion of *dynamic consequence*. Information changes, soft or hard, are seen as licensing specific kinds of inferences, that is inducing a specific consequence relation. For DEL this idea has been first proposed by van Benthem [10] in the context of *public announcement logic*. The idea is this. Public announcement logic contains formula $[\phi]\psi$, to be read “ ψ holds after the announcement of ϕ ”. One can use these formulas to define a consequence relation as follows:

³This domain is usually assumed to be finite.

⁴Of course, not all DEL-like systems are prone to this reduction technique. Well-known examples are public announcements in $S5 + \text{Common knowledge}$ [3] and the logic of “epistemic protocols” [13].

$$\phi_1, \dots, \phi_n \models_{dyn} \psi \text{ iff } \mathcal{M}, w \models [\phi_1] \dots [\phi_n] \psi \text{ for all pointed models } \mathcal{M}, w.$$

The study of such dynamic consequence relations has been taken up by a number of authors. Cerdón-Franco et al. [5], investigate a generalization of the above in which the class of models where the announcements are made is restricted by a certain set of background formulas Γ , and Aucher [1] provides a sound and complete axiomatization of a dynamic consequence relation defined not only for public announcements, but for the general operation of “product update” in full dynamic epistemic logic. [15]

Dynamic consequence relations are interesting for a number of reasons. First, dynamic consequence relations unveil a deep connection between logics for information dynamics and sub-structural logic [8]. Indeed, from a proof-theoretic point of view dynamic consequences are highly sub-structural.

Fact (van Benthem [10]). All classical structural rules [contraction, weakening, uniform substitution, etc] fail for \models_{dyn} .

In the same paper van Benthem shows that \models_{dyn} can nevertheless be completely axiomatized by a number of inference rules reminiscent of rules for non-monotonic reasoning. These rules, however, are not valid for the generalizations investigated by Cerdón-Franco et al., and the sound and complete proof system devised by Aucher indeed lacks all the classical structural rules.

Dynamic consequence relations are also interesting for dynamic epistemic logic itself. By now the standard technique to axiomatize DEL-validities is to use so-called *reduction axioms*, which essentially show how to analyze, compositionally, the *effects* of epistemic action in terms of static conditions in the underlying static (and event) model(s).⁴ This methodology, however, says very little about the logical operations *on the epistemic actions themselves*, except for certain forms of contractions. Valid and invalid structural rules in dynamic consequences, on the other hand, do make explicit logical operations on the epistemic actions. The study of dynamic consequence thus unveil logical principles governing the dynamic of information that were up to now implicit by the reduction axiom methodology. Furthermore, the study of dynamic consequence relations puts DEL in historical perspective. As noted in Cerdón-Franco et al., it

constitutes a return to the original motivation behind *dynamic semantics* [9, 16], arguably at the source of DEL.

Dynamic Consequence for Soft Information

In this paper we take up the task of extending and/or adapting the results in Aucher [1] to dynamics of soft information. More precisely, our goal is, first to provide a sound and complete axiomatization of the dynamic consequence relation defined as follows:

$\phi_1, \phi_2 \vdash \phi_3$ iff $\mathcal{M} \otimes \mathcal{E}, (w, e) \models \phi_3$ for any epistemic-plausibility model \mathcal{M}, w and event models \mathcal{E}, e such that $\mathcal{M}, w \models \phi_1 \wedge pre(e)$ and $\mathcal{E}, e \models \phi_2$.

The formulas ϕ_1 and ϕ_3 in a sequent $\phi_1, \phi_2 \vdash \phi_3$ are formulas of \mathcal{L}_S , and the updated model $\mathcal{M} \otimes \mathcal{E}$ is obtained by lexicographic update. ϕ_2 is a formula of a new language \mathcal{L}_E , whose syntax extends \mathcal{L}_S with a modality $[<]_i$ that describes the strict version of $[\leq]_i$.⁵ This language is interpreted directly on event models. This two-language re-description of standard DEL⁶ comes from Aucher [1], and is natural given our goal of capturing the logical operations on epistemic actions that are allowed by such sequents.

Fact. The set of axioms and rules in Table 1 is sound for the dynamic consequence relation \vdash .

The axioms and rules R2 and R3 are imported directly from Aucher’s calculus. Dynamic consequence for soft information exhibits the same sub-structural phenomena than its correspondent for product update. The rules R4, R5 and R6 capture the case distinction built-in the lexicographic update. It should be noted, furthermore, that read bottom-up these rules also encode a reduction from complex to simpler sequent, echoing the reduction methodology in standard DEL while keeping explicit how these operations also bear epistemic actions.

We conjecture this set of axiom and rule is also complete, but at the moment of submitting this abstract we are still working on the proof. Incidentally, this proof also makes heavy use of so-called “Fine formulas” developed for co-algebraic views on modal logics [7, 17], which are of independent interest.

Bibliography

- [1] G. Aucher. Del-sequents for progression. *Journal of Applied Non-Classical Logics*, 21(3-4), 2011.

⁵This modality is not definable in \mathcal{L}_S . See for instance [14] for details.

⁶On finite event models this presentation is expressively equivalent to the more “standard” one presented above.

- [2] A. Baltag and S. Smets. Conditionl doxastic models: A qualitative approach to dynamic belief revision. In *Electornic notes in theoretical computer science*, volume 165, pages 5 – 21. Springer, 2006.
- [3] A. Baltag, L.S. Moss, and S. Solecki. The logic of public announcements, common knowledge and private suspicions. In *TARK 98*, 1998.
- [4] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2002.
- [5] A. Cerdón-Franco, H. van Ditmarsch, D. Fernández-Duque, E. Gómez-Caminero, and A. Nepomuceno-Fernández. Two mischievous dynamic consequence relations. In *Int. Workshop on Logic and Philosophy of Knowledge, Communication and Action*, pages 197–208, 2010.
- [6] P. Gärdenfors. Belief revision and nonmonotonic logic: two sides of the same coin? *Logics in AI*, pages 52–54, 1991.
- [7] L.S. Moss. Finite models constructed from canonical formulas. *Journal of Philosophical Logic*, 36(6):605–640, 2007.
- [8] G. Restall. *An Introduction to Sub-Structural Logics*. Routledge, London, 2000.
- [9] J. van Benthem. *Exploring logical dynamics*. CSLI publications Stanford, 1996.
- [10] J. van Benthem. Logical dynamics meets logical pluralism? *The Australasian Journal of Logic*, 6:182–209, 2008.
- [11] J. van Benthem. *Logical Dynamics of Information and Interaction*. Cambridge University Press, 2011.
- [12] J. van Benthem, J. van Eijck, and B. Kooi. Logics of communication and change. *Information and Computation*, 204(11):1620 – 1662, 2006.
- [13] J. Van Benthem, J. Gerbrandy, T. Hoshi, and E. Pacuit. Merging frameworks for interaction. *Journal of Philosophical Logic*, 38(5):491–526, 2009.
- [14] J. van Benthem, P. Girard, and O. Roy. Everything else being equal: A modal logic for *ceteris paribus* preferences. *Journal of Philosophical Logic*, 38:83–125, 2009.
- [15] H. van Ditmarsch, W. van de Hoek, and B. Kooi. *Dynamic Epistemic Logic*, volume 337 of *Synthese Library Series*. Springer, 2007.
- [16] F. Veltman. Defaults in update semantics. *Journal of philosophical logic*, 25(3):221–261, 1996.
- [17] Y. Venema. *Modal Logic, Algebra and Coalgebra*. ILLC, Amsterdam, 2012. Downloaded from the author’s website on 01-03-12.

	$\perp, \phi_2 \vdash \phi_3$	A1	$\phi_1, \perp \vdash \phi_3$	A2
	$\phi_1, \phi_2 \vdash \top$	A3	$p, \phi_2 \vdash p$	A4
	$\neg p, \phi_2 \vdash \neg p$	A5	$\neg Pre(p), p \vdash \perp$	A6
R2:	$\frac{\phi_1 \wedge \psi_1, \phi_2 \vdash \phi_3 \quad \neg \psi_1, \phi_2 \vdash \phi_3}{\phi_1, \phi_2 \vdash \phi_3}$			
R3:	$\frac{\phi_1, \phi_2 \wedge \psi_2 \vdash \phi_3 \quad \phi_1, \neg \psi_2 \vdash \phi_3}{\phi_1, \phi_2 \vdash \phi_3}$			
R4:	$\frac{\top, \phi_2 \vdash \phi_3 \quad \phi_1, \phi_4 \vdash \phi_3}{[\leq]_i \phi_1, [\prec]_i^{\mathcal{E}} \phi_2 \wedge [\leq]_i^{\mathcal{E}} \phi_4 \vdash [\leq]_i \phi_3}$			
R5:	$\frac{\top, \phi_2 \vdash \phi_3}{\langle \sim \rangle_i pre(p), \langle \prec \rangle_i^{\mathcal{E}} (\phi_2 \wedge p) \vdash \langle \leq \rangle_i \phi_3}$			
R6:	$\frac{\phi_1, \phi_2 \vdash \phi_3}{\langle \leq \rangle_i (\phi_1 \wedge pre(p)), \langle \leq \rangle_i^{\mathcal{E}} (\phi_2 \wedge p) \vdash \langle \leq \rangle_i \phi_3}$			

Table 1: A sound set of axioms and rules for dynamic consequences for soft information.

Deep relevant logics not included in relevant logic R

F. Salto*, G. Robles**, J. M. Méndez***

As it is well known, according to Anderson and Belnap, the following is a necessary property of any relevant logic S (see [1]):

Definition 1 (Variable-sharing property —vsp) *If $A \rightarrow B$ is a theorem of S , then A and B share at least a propositional variable.*

In [2], Brady strengthens the vsp as follows.

Definition 2 (Depth relevance property —drc) *If $A \rightarrow B$ is a theorem of S , then A and B share at least a propositional variable at the same depth.*

The depth of an occurrence of a subformula B in a formula A “is roughly the number of nested ‘ \rightarrow ’s required to reach the occurrence of B in A ” ([2], p. 63). And logics with the drc are named “deep relevant logics”.

The definition of the drc is motivated as a necessary condition for strong paraconsistent logics. And in the cited paper [2], Brady’s strategy is to restrict with the drc the class of logics with the vsp verified by Meyer’s Crystal Matrix CL. The matrix CL is axiomatized by adding to relevant logic R (see [1]) the following axioms (see [4], pp 85, ff):

$$\begin{aligned} \text{CL1. } & (\neg A \wedge B) \rightarrow [(\neg A \rightarrow A) \vee (A \rightarrow B)] \\ \text{CL2. } & A \vee (A \rightarrow B) \end{aligned}$$

Brady chooses the logic DR (presumably an abbreviation for “Depth Relevance”) as the preferred one among those definable from CL as indicated. The logic DR is the result of adding to Routley and Meyer’s basic logic B (cf., e. g., [5]) the following axioms, rule and metarule (see [2], [3]):

$$\begin{aligned} \text{DR1. } & [(A \rightarrow B) \wedge (B \rightarrow C)] \rightarrow (A \rightarrow C) \\ \text{DR2. } & A \vee \neg A \\ \text{DR3. } & A \Rightarrow \neg(A \rightarrow \neg A) \\ \text{DR4. } & A \Rightarrow B \Rightarrow (C \vee A) \rightarrow (C \vee B) \end{aligned}$$

Now, CL1 and CL2, which are, obviously, acceptable from the vsp point of view, are not admissible from the drc one. Consequently, it is reasonable to

think that all deep relevant logics definable from CL are included in relevant logic R.

The aim of this paper is to generalize Brady's strategy by showing how to define a class of deep relevant logics from each weak relevant matrix. "Weak relevant matrices" are defined in [6] and are characterized as matrices verifying only logics with vsp. It will be proved that there are deep relevant logics not included in R. Actually, not included in R-mingle, which, as it is known, is a logic lacking the vsp axiomatizable by adding to R the axiom "mingle" (see [1]):

$$\text{RM1. } A \rightarrow (A \rightarrow A)$$

That is, it will be shown that there are deep relevant logics well far off the spectrum of standard relevant logics.

Acknowledgements

-Work supported by research project FFI2011-28494 financed by the Spanish Ministry of Economy and Competitiveness.

-G. Robles is supported by Program Ramón y Cajal of the Spanish Ministry of Economy and Competitiveness.

*Dpto. de Psicología, Sociología y Filosofía, Universidad de León
Campus de Vegazana, s/n, 24071, León, Spain
<http://www3.unileon.es/personal/wwdfcfsa>
francisco.salto@unileon.es

**Dpto. de Psicología, Sociología y Filosofía, Universidad de León
Campus de Vegazana, s/n, 24071, León, Spain
<http://grobv.unileon.es>
gemmarobles@gmail.com

***Universidad de Salamanca
Campus Unamuno, Edificio FES, 37007, Salamanca, Spain
<http://web.usal.es/sefus>
sefus@usal.es

References

- [1] Anderson, A. R., Belnap, N. D. Jr., *Entailment. The Logic of Relevance and Necessity*, vol I. Princeton University Press, 1975.
- [2] Brady, R.T., "Depth relevance of some paraconsistent logics", *Studia Logica* 43, (1984), pp. 63–73.
- [3] Brady, R. T., "Hierarchical semantics for relevant logics", *Journal of Philosophical Logic*, 21, (1992), pp. 337–374.
- [4] Brady, R. T. (ed.), *Relevant logics and their rivals*, Vol. II, Ashgate, 2003.

- [5] Robles, G., Méndez, J. M., "A general characterization of the variable-sharing property by means of logical matrices", *Notre Dame Journal of Formal Logic* (accepted).
- [6] Routley, R. Meyer, R. K., Plumwood, V., Brady R. T., *Relevant Logics and their Rivals, vol. 1*, (Atascadero, CA: Ridgeview Publishing Co.), 1982.

DYNAMIC MODALITIES

Extended abstract

Dimitar Vakarelov, Sofia University
e-mail: dvak@fmi.uni-sofia.bg

Introduction.

In this paper we introduce a new modal logic called LDM (Logic with Dynamic Modalities). It contains four modalities of dynamic nature with the following informal readings:

- \Box^\forall – **always necessary**, necessary in all situations,
- \Box^\exists – **sometimes necessary**, necessary in some situations,

and their duals:

- \Diamond^\forall – **always possible**, possible in all situations, and
- \Diamond^\exists – **sometimes possible**, possible in some situations.

Let us note that such modalities are frequently used in ordinary language. For instance, when the doctor prescribes some medicines, for some of them he says that they should be taken every day ("always necessary") and for others, that they should be taken only if some special symptoms occur ("sometimes necessary"). Of course some other names for these modalities are possible, for instance \Box^\forall – "absolute necessity", "strong necessity", and \Box^\exists – "weak necessity" and similar names for their duals. We call such modalities "dynamic" because they are characteristics of changing necessity and possibility, so in this sense LDM is related to propositional dynamic logic (PDL), but at the same time it is quite different from it. Temporal logics with modalities having similar informal interpretation but with different formal semantics, have been studied semantically by Vladimir Rybakov [1]. The full version of this paper is [2].

Syntax and semantics.

The language of LDM is an extension of the language of propositional logic with four unary modalities $\Box^\forall, \Box^\exists, \Diamond^\forall, \Diamond^\exists$ with the standard definition of a formula. We consider standard notations for propositional connectives: $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow, \perp, \top$.

The semantics is based on the so called dynamic frames, which are structures in the following form: $(W, \{R_i : i \in S\})$, where $W \neq \emptyset$, $S \neq \emptyset$ and for each $i \in S$, R_i is a binary relation in W . The elements of W are called, as usual, possible worlds, and the elements of S are considered as *situations*. So, for each situation $i \in S$ we associate a binary relation $R_i \subseteq W^2$ which describes the local necessity and possibility related to i . Note that we do not have in the language formal operators corresponding to these relations sim-

ply because we allow their number (finite or infinite) to change from frame to frame.

A mapping $v(x, p)$ which assigns to each propositional variable p and $x \in W$ the values 0 (falsity) and 1 (truth) is called a valuation and $M = (W, \{R_i : i \in S\}, v)$ is called a model. The extension of v to arbitrary formulas is by induction on the complexity of formulas. The truth conditions of non-modal connectives in a model M are as in the ordinary Kripke semantics and for the modal connectives they are as follows:

$$v(x, \Box^\forall A) = 1 \text{ iff } (\forall i \in S)(\forall y \in W)(xR_i y \rightarrow v(y, A) = 1),$$

$$v(x, \Box^\exists A) = 1 \text{ iff } (\exists i \in S)(\forall y \in W)(xR_i y \rightarrow v(y, A) = 1),$$

and dually for \Diamond^\forall and \Diamond^\exists :

$$v(x, \Diamond^\forall A) = 1 \text{ iff } (\forall i \in S)(\exists y \in W)(xR_i y \text{ and } v(y, A) = 1),$$

$$v(x, \Diamond^\exists A) = 1 \text{ iff } (\exists i \in S)(\exists y \in W)(xR_i y \text{ and } v(y, A) = 1).$$

We adopt the standard definitions of a formula to be true in a model, in a frame, and in a class of frames.

The formal semantics shows that we may take the pair $\Box^\forall, \Box^\exists$ as primitives and the other two to be defined: $\Diamond^\exists A =_{def} \neg \Box^\forall \neg A$, and $\Diamond^\forall A =_{def} \neg \Box^\exists \neg A$, or vice versa.

Axiomatization.

We propose the following axiom system for LDM:

Axiom schemes for LDM.

(I) Axiom schemes for classical propositional logic.

(II) Axiom schemes for \Box^\forall and \Box^\exists :

$$(K_{\Box^\forall}) \quad \Box^\forall(A \Rightarrow B) \Rightarrow (\Box^\forall A \Rightarrow \Box^\forall B),$$

$$(Mono_{\Box^\exists}) \quad \Box^\forall(A \Rightarrow B) \Rightarrow (\Box^\exists A \Rightarrow \Box^\exists B),$$

$$(\Box^\forall \rightarrow \Box^\exists) \quad \Box^\forall A \rightarrow \Box^\exists A.$$

Rules of inference: Modus ponens (MP): $\frac{A, A \Rightarrow B}{B}$,

Necessitation for \Box^\forall (N): $\frac{A}{\Box^\forall A}$.

Lemma 1.(i) The following rules are provable in LDM:

$$(ia) \text{ (MONO } \Box^\forall) \quad \frac{A \Rightarrow B}{\Box^\forall A \Rightarrow \Box^\forall B}, \text{ (MONO } \Box^\exists) \quad \frac{A \Rightarrow B}{\Box^\exists A \Rightarrow \Box^\exists B},$$

(ib) The rule of replacement of equivalents.

(ii) Examples of theorems and non-theorems

(iia) The formula $A = \Box^\exists(p \Rightarrow q) \Rightarrow (\Box^\exists p \Rightarrow \Box^\exists q)$ is not theorem of LDM (and hence \Box^\exists is not normal modality).

(iib) The formulas $\Box^\exists \top$ and $\Box^\forall A \wedge \Box^\exists B \Rightarrow \Box^\exists(A \wedge B)$ are theorems of LDM.

Note that it follows from the axiomatization of LDM and Lemma 1 that \Box^\forall is a normal modality, while \Box^\exists is a non-normal monotonic modality.

Completeness and decidability.

Theorem 1. [Completeness theorem for LDM]. Let Σ be a set of formulas and A be a formula. Then:

- (i) (Strong form) Σ is consistent $\longleftrightarrow \Sigma$ has a model.
- (ii) (Weak form) A is a theorem $\longleftrightarrow A$ is true in all frames.

Idea of the proof. The proof goes through a canonical construction. The fact that \Box^\forall is a normal modality makes possible to use its canonical model construction and to divide in a suitable way the canonical relation R_{\Box^\forall} into a set $\{R_i : i \in S\}$ of binary relations, and to use them for defining the canonical model for the logic.

Theorem 2. [Decidability of LDM] The logic LDM possesses the final model property and hence is decidable.

Idea of the proof. The proof goes through a suitable modification of the classical filtration method simulating the construction of completeness proof.

Concluding remarks.

LDM is the minimal logic corresponding to the class of all dynamic frames. One can consider different classes of dynamic frames and the corresponding logics. So all standard questions for a given class of logics, like definability, completeness, complexity, possible relations to known classes of logics, are open. Applications to some concrete applied domains and systems for practical reasoning and knowledge representation are also possible and are in our plans for the future.

Thanks. This work was partially supported by the contract No176/2012 with Sofia University. Thanks are due to the anonymous referee for pointing out some misprints.

References

- [1] Vladimir V. Rybakov: Temporal logic with interacting agents. *Journal of Applied Non-Classical Logics* 18(2-3): 293-308 (2008)
- [2] Dimitar Vakarelov, Dynamic Modalities. *Studia Logica* 100(1-2):385-397(2012).

Remarks on Proof-Theoretic Semantics of Intensional Transitive Verbs

Bartosz Więckowski
bartosz.wieckowski@uni-greifswald.de
Universität Greifswald, Germany

Abstract

Sentences which contain intensional transitive verbs (e.g., ‘seek’, ‘owe’) give rise to an ambiguity as they allow for both a specific and an unspecific reading. Consider

(1) John seeks a baby-sitter.

On the specific reading John is looking for a particular baby-sitter, whereas on the unspecific reading he is looking for a baby-sitter but not for a particular one. My project for the presentation will be to contribute to the clarification of the semantics and the logical behavior of intensional transitive verbs by means of an MLTT-formalism [6] which integrates subatomic derivation [5]. This abstract outlines the main ideas concerning the analysis of intensional transitives and presents simple examples. For the details of the formalisms the reader is referred to [5; 6].

1. Subatomic derivation. In Prawitz-style proof-theoretic semantics the meaning of an atomic sentence is determined by a set of derivations in an atomic system which contain that sentence as a conclusion [2; 3]. These systems perform a role analogous to that of models in model-theoretic semantics in that they determine the valid atomic sentences. Subatomic systems [5] perform this role as well, but they do so by means of normalizing subatomic introduction and elimination rules for atomic sentences. Roughly, the introduction rule determines when an atomic sentence can be inferred from the term assumptions associated with the terms (i.e., individual (or nominal) constants and atomic predicates) from which it is composed. The term assumptions for a term are, in effect, sets of atomic sentences which contain that term. The elimination rule, by contrast, determines how term assumptions can be inferred from atomic sentences. In view of a simple subatomic normal form theorem, these systems admit a proof-theoretic account of the semantic behavior of first-order atomic sentences and their components.

2. Subatomic MLTT-formalism. In [6] a proof-theoretic semantics for a fragment of English within a type-theoretical formalism is developed that combines subatomic systems for natural deduction with MLTT by stating rules for the formation, introduction, elimination and equality of atomic propositions understood as types (or sets) of subatomic proof-objects. The formalism is extended with dependent types in order to allow for an interpretation of non-atomic sentences. In this formalism subatomic systems replace denotational model-theoretic bases which are traditionally used by MLTT-approaches to natural language (see, e.g., [4]). The basic idea is to explain atomic propositions as constructive sets of tuples of association functions which in a subatomic system serve to assign term assumptions to nominal constants and atomic predicates of a first-order language. In this way atomic sentences and other subatomically sensitive natural language constructions receive a type-theoretical interpretation which does not appeal to denotational bases (possibly accompanied by an epistemology of direct, e.g., visual, verification of atomic sentences), but rests on derivations. In contrast to currently available proof-theoretic semantic frameworks for natural language (e.g., [1], [4]), the formalism developed in [6] for (a fragment of) English allows to explain atomic constructions in a compositional manner which in contrast to [1] is “bottom-up”. Moreover, it is in a position to interpret a wide range of natural language constructions (e.g., sentences containing internally nested proper names, identity sentences, donkey sentences), in a rather fine-grained way. This granularity can be exploited for the interpretation of intensional transitives.

3. Intensional transitives and subatomic natural deduction. In subatomically extended natural deduction [5] the specific/unspecific distinction can be made precise in terms of the term assumptions for nominal constants which are used in the derivations of the symbolizations of the ambiguous sentence. A nominal constant will be said to be unspecific in case the term assumptions for it contain exactly the atomic sentences which are composed from the elementary predicates which occur in the first-order symbolization of the ambiguous sentence, otherwise it will be said to be specific. We may allow the special case in which the term assumptions for a specific constant are only as specific as those for an unspecific one. In such cases the distinction collapses.

Illustration: Let $(\exists x)(Bx \ \& \ S_jx)$ be the first-order symbolization of (1). In subatomic natural deduction the unspecific reading of (1) can be represented by derivation $(1u)$ and the specific reading by $(1s)$:

(1u):

$$\frac{\frac{B\{\dots, Bc, \dots\} \quad c\{Bc, Sjc\}}{Bc} atI \quad \frac{S\{\dots, Sjc, \dots\} \quad j\{\dots, Sjc, \dots\} \quad c\{Bc, Sjc\}}{Sjc} atI}{\frac{Bc \& Sjc}{(\exists x)(Bx \& Sjc)} \exists I} \&I$$

(1s):

$$\frac{\frac{B\{\dots, Ba, \dots\} \quad a\{\dots, Ba, Sja, \dots\}}{Ba} atI \quad \frac{S\{\dots, Sja, \dots\} \quad j\{\dots, Sja, \dots\} \quad a\{\dots, Ba, Sja, \dots\}}{Sja} atI}{\frac{Ba \& Sja}{(\exists x)(Bx \& Sjc)} \exists I} \&I$$

The derivations differ in that, first, the unspecific case (1u) contains the nominal constant c wherever the specific case (1s) contains the constant a and, second, the term assumptions for c in (1u) are confined to exactly the atomic sentences which are composed from the predicates which figure in the conclusion, whereas the term assumptions for a contain more than these two atomic sentences.

4. *Intensional transitives and subatomic MLTT*. The type-theoretic meaning of (1), on the unspecific reading, can be given by the following derivation which is subject to the proviso stated below it:

$$\frac{\frac{b^1 \in \mathbf{b}^1 \quad x \in \mathbf{x}}{\langle b^1, x \rangle \in \mathbf{b}^1 \cdot \mathbf{x}} afI \quad \frac{s^2 \in \mathbf{s}^2 \quad j \in \mathbf{j} \quad r_1(\langle b^1, x \rangle) \in \mathbf{r}_1(\langle \mathbf{b}^1, \mathbf{x} \rangle)}{\langle s^2, j, r_1(\langle b^1, x \rangle) \rangle \in \mathbf{s}^2 \cdot \mathbf{j} \cdot \mathbf{r}_1(\langle \mathbf{b}^1, \mathbf{x} \rangle)} afI}{(\langle b^1, x \rangle, \langle s^2, j, r_1(\langle b^1, x \rangle) \rangle) \in (\Sigma y \in \mathbf{b}^1 \cdot \mathbf{x}) \mathbf{s}^2 \cdot \mathbf{j} \cdot \mathbf{r}_1(\mathbf{y})} \Sigma aI$$

Proviso: x is such that, in addition to the usual conditions for (afI) (these conditions are omitted here), the following condition is satisfied: $x(x) = c$ and $c(c) = \{Bc, Sjc\}$. For the specific reading of (1) the proviso differs in that x is such that: $x(x) = a$ and $\{Ba, Sja\} \subset a(a)$.

Inference (2) is an instance of the upward monotonicity inference pattern for search-verbs where both premiss and conclusion are understood in the unspecific sense. This pattern is validated on the present type-theoretical account.

$$(2) \quad \frac{\text{John seeks a female baby-sitter}}{\text{John seeks a baby-sitter}}$$

This instance corresponds to the type-theoretical derivation in Fig. 1.

In the presentation these ideas will be explained in detail, further applications to intensional transitives will be discussed, and attention will be paid to the proof-theoretic approach to the semantics of intensional transitives sketched in [1].

References

- [1] Francez, N. & Dyckhoff, R. (2010). Proof-theoretic semantics for a natural language fragment. *Linguistics and Philosophy* 33, 447-477.
- [2] Prawitz, D. (1971). Ideas and results in proof theory. In J. E. Fenstad (ed.) *Proceedings of the Second Scandinavian Logic Symposium (Oslo 1970)*. Amsterdam: North-Holland, 235-309.
- [3] Prawitz, D. (1973). Towards a foundation of a general proof theory. In P. Suppes *et al.* (eds.) *Logic, Methodology, and Philosophy of Science IV*. Amsterdam: North-Holland, 225-250.
- [4] Ranta, A. (1994). *Type-Theoretical Grammar*. Oxford: Clarendon Press.
- [5] Więckowski, B. (2011). Rules for subatomic derivation. *The Review of Symbolic Logic* 4, 219-236.
- [6] Więckowski, B. (2012). A constructive type-theoretical formalism for the interpretation of subatomically sensitive natural language constructions. Forthcoming in *Studia Logica*, special issue on *Logic and Natural Language* edited by N. Francez and I. Pratt-Hartmann.

$$\begin{array}{c}
\frac{b^1 \in \mathbf{b}^1 \quad x \in \mathbf{x}}{\langle b^1, x \rangle \in \mathbf{b}^1 \cdot \mathbf{x}} \frac{afI}{afE} \\
\frac{r_1(\langle b^1, x \rangle) \in \mathbf{x}}{+r_1(\langle b^1, x \rangle) \in \mathbf{x}} \frac{spl}{afI} \\
\frac{f^1 \in \mathbf{f}^1}{\langle f^1, +r_1(\langle b^1, x \rangle) \rangle \in \mathbf{f}^1 \cdot \mathbf{x}} \frac{s^2 \in \mathbf{s}^2 \quad j \in \mathbf{j} \quad r_1(\langle f^1, +r_1(\langle b^1, x \rangle) \rangle) \in r_1(\langle \mathbf{f}^1, r_1^+(\langle \mathbf{b}^1, \mathbf{x} \rangle) \rangle)}{\langle s^2, j, r_1(\langle f^1, +r_1(\langle b^1, x \rangle) \rangle) \in \mathbf{s}^2 \cdot \mathbf{j} \cdot r_1(\langle \mathbf{f}^1, r_1^+(\langle \mathbf{b}^1, \mathbf{x} \rangle) \rangle)} \frac{afI}{\Sigma aI} \\
\frac{(\langle f^1, +r_1(\langle b^1, x \rangle) \rangle, \langle s^2, j, r_1(\langle f^1, +r_1(\langle b^1, x \rangle) \rangle) \rangle) \in (\Sigma y \in \mathbf{f}^1 \cdot \mathbf{x}) \mathbf{s}^2 \cdot \mathbf{j} \cdot r_1(\mathbf{y})}{p(\langle f^1, +r_1(\langle b^1, x \rangle) \rangle, \langle s^2, j, r_1(\langle f^1, +r_1(\langle b^1, x \rangle) \rangle) \rangle) \in \mathbf{f}^1 \cdot \mathbf{x}} \frac{\Sigma aE1}{afE} \\
\frac{b^1 \in \mathbf{b}^1}{\langle b^1, r_1(p(\langle f^1, +r_1(\langle b^1, x \rangle) \rangle, \langle s^2, j, r_1(\langle f^1, +r_1(\langle b^1, x \rangle) \rangle) \rangle)) \in \mathbf{b}^1 \cdot \mathbf{x}} \frac{s^1 \in \mathbf{s}^1 \quad j \in \mathbf{j} \quad r_1(h) \in r_1(\mathbf{h})}{\langle s^1, j, r_1(h) \rangle \in \mathbf{s}^1 \cdot \mathbf{j} \cdot r_1(\mathbf{h})} \frac{afI}{\Sigma aI} \\
\frac{(h, \langle s^1, j, r_1(h) \rangle) \in (\Sigma z \in \mathbf{b}^1 \cdot \mathbf{x}) \mathbf{s}^1 \cdot \mathbf{j} \cdot r_1(\mathbf{z})}{}
\end{array}$$

Where: $h =_{df} \langle b^1, r_1(p(\langle f^1, +r_1(\langle b^1, x \rangle) \rangle, \langle s^2, j, r_1(\langle f^1, +r_1(\langle b^1, x \rangle) \rangle) \rangle)) \rangle$.

Figure 1

(Side conditions omitted in Fig. 1.)