On Propositional Dynamic Logic and Concurrency

Matteo Acclavio[1], Fabrizio Montesi[2], and Marco Peressotti[2]

[1] University of Sussex, Brighton, UK
[2] University of Southenr Denmark, Odense, DK

Dynamic logics are families of logics where programs are part of the language of formulas itself, which enables the direct use of the logic to reason about the semantics of programs [11]. At the syntactical level, each program a defines the modalities $[a]$ and $\langle a \rangle$ and a formula $[a]\,\phi$ is interpreted as "every state reached after executing a satisfies the formula ϕ" while a formula $\langle a \rangle\,\phi$ is interpreted as "there is a state reached after executing a satisfying the formula F". Instances of dynamic logic include the modal μ-calculus [17], the Hennessy-Milner logic [12], and the propositional dynamic logic (PDL) [11], and provide solid foundations for the study of program verification and model checking [31, 7].

PDL and the concurrency problem

While PDL has been successfully applied to the study of sequential programs, extending this approach to concurrent programs has been proved to be challenging. In standard PDL, a program is represented by a regular expression that describes its set of possible traces. In other words, programs are elements of a free Kleene algebra. This representation of programs is satisfactory when reasoning about sequential programs, because one obtains that the theory of equational reasoning for Kleene algebras is a complete system for reasoning about *trace equivalence* [14, 18, 15, 30]. Trace equivalence is therefore captured by logical equivalence in PDL:

$$a \text{ and } b \text{ have the same traces} \quad \text{iff} \quad \vdash_{\mathsf{PDL}} [a]\,\phi \Leftrightarrow [b]\,\phi \text{ for any formula } \phi\,. \tag{1}$$

However, the case of concurrent programs with an interleaving semantics is more problematic. In the presence of interleaving, one expects traces differing by interleaving to be equivalent modulo equations of the form $a; b = b; a$ (called *commutations*). Unfortunately, the word problem in a Kleene algebra enriched with an equational theory containing such commutations is known to be undecidable[1] which makes undecidable checking whether two modalities in PDL are equivalent same.

As a consequence of this problem, applications of PDL to concurrency fall short of the expected level of expressivity from established theories, like CCS [23] and the π-calculus [24]. For example, previous works lack nested parallel composition, synchronisation, or recursion [21, 5, 28, 29, 27, 4]. In general, adding any new concurrency feature (e.g., a construct in the language of programs or a law defining its semantics) requires great care and effort in establishing the meta-theoretical properties of the logic. The result: a literature of various PDL, all independently useful, but with different limitations and dedicated technical developments.

In this talk

We discuss the result in [2], where we develop *operational propositional dynamic logic* (OPDL). The key innovation of OPDL is to distinguish and separate reasoning on programs from reasoning

[1] In [16] is proven that the word problem in a star-continuous Kleene algebra can be reduced to an instance of the Post correspondence problem, by combining sequential composition, iteration, and commutations. This result has been recently extended to the general case of Kleene algebras [3].

on their traces. Thanks to this distinction, we circumvent previous limitations and finally obtain a PDL that can be applied to established concurrency models, such as CCS [23] and choreographic programming [25]. Crucially, OPDL is a general framework: it is parameterised on the operational semantics used to generate traces from programs, yielding a simple yet reusable approach to characterise trace reasoning.

After recalling the axiomatization and semantics of PDL, we provide a proof of its soundness and completeness with respect to the non-wellfounded sequent calculus introduced in [8]. For this purpose, we provide the first cut-elimination result for this non-wellfounded calculus, by adapting the technique developed in [1].[2] This allows us to prove our results by reasoning on the axiomatisation and the sequent system, without directly relying on semantic arguments.

Then, we extend PDL with an additional axiom allowing us to encapsulate an operational semantics for a set of programs into the trace reasoning.

$$\mathsf{A}_{\mathcal{O}} \ : \ [\alpha]\,\phi \Leftrightarrow \left(\bigwedge_{\alpha \xrightarrow{b} \gamma} [b]\,[\gamma]\,\phi \right) \quad \text{with } \alpha \xrightarrow{b} \gamma \text{ in the operational semantics } \mathcal{O} \tag{2}$$

We call the resulting logic *operational propositional dynamic logic* (or OPDL), providing a general framework encompassing various previous works [21, 5, 10], and we provide instantiations of OPDL for Milner's CCS [23] and Montesi's latest presentation of choreographic programming [26].

We conclude by discussing the open questions about the axiomatizaiton of algebraic models for OPDL, and we provide a roadmap for future research in this area.

References

[1] Matteo Acclavio, Gianluca Curzi, and Giulio Guerrieri. Infinitary cut-elimination via finite approximations. *CoRR*, abs/2308.07789, 2023.

[2] Matteo Acclavio, Fabrizio Montesi, and Marco Peressotti. On propositional dynamic logic and concurrency, 2024.

[3] Arthur Azevedo de Amorim, Cheng Zhang, and Marco Gaboardi. Kleene algebra with commutativity conditions is undecidable, 2024. Available at: https://arxiv.org/abs/2411.15979.

[4] Mario Benevides. Bisimilar and logically equivalent programs in pdl with parallel operator. *Theoretical Computer Science*, 685:23–45, 2017. Logical and Semantic Frameworks with Applications.

[5] Mario R.F. Benevides and L. Menasché Schechter. A propositional dynamic logic for concurrent programs based on the π-calculus. *Electronic Notes in Theoretical Computer Science*, 262:49–64, 2010. Proceedings of the 6th Workshop on Methods for Modalities (M4M-6 2009).

[6] Luís Caires and Frank Pfenning. Session types as intuitionistic linear propositions. In Paul Gastin and François Laroussinie, editors, *CONCUR 2010 - Concurrency Theory, 21th International Conference, CONCUR 2010, Paris, France, August 31-September 3, 2010. Proceedings*, volume 6269 of *Lecture Notes in Computer Science*, pages 222–236. Springer, 2010.

[7] Sjoerd Cranen, Jan Friso Groote, Jeroen J. A. Keiren, Frank P. M. Stappers, Erik P. de Vink, Wieger Wesselink, and Tim A. C. Willemse. An overview of the mcrl2 toolset and its recent advances. In Nir Piterman and Scott A. Smolka, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 19th International Conference, TACAS 2013, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2013, Rome, Italy, March*

[2] A cut-elimination result for another sequent calculus for PDL is provided in [13], but that calculus is fundamentally different: it employs nested sequents and contains rules with an infinite number of premises.

16-24, 2013. Proceedings, volume 7795 of *Lecture Notes in Computer Science*, pages 199–213. Springer, 2013.

[8] Anupam Das and Marianna Girlando. Cyclic proofs, hypersequents, and transitive closure logic. In Jasmin Blanchette, Laura Kovács, and Dirk Pattinson, editors, *Automated Reasoning*, pages 509–528, Cham, 2022. Springer International Publishing.

[9] Jean-Yves Girard, Paul Taylor, and Yves Lafont. *Proofs and types*, volume 7. Cambridge university press Cambridge, 1989.

[10] D. Harel and R. Sherman. Propositional dynamic logic of flowcharts. *Information and Control*, 64(1):119–135, 1985. International Conference on Foundations of Computation Theory.

[11] David Harel, Dexter Kozen, and Jerzy Tiuryn. *Dynamic Logic*, pages 99–217. Springer Netherlands, Dordrecht, 2002.

[12] Matthew Hennessy and Robin Milner. On observing nondeterminism and concurrency. In *International Colloquium on Automata, Languages, and Programming*, pages 299–309. Springer, 1980.

[13] Brian Hill and Francesca Poggiolesi. A contraction-free and cut-free sequent calculus for propositional dynamic logic. *Studia Logica*, 94(1):47–72, 2010.

[14] John E Hopcroft, Rajeev Motwani, and Jeffrey D Ullman. Introduction to automata theory, languages, and computation. *Acm Sigact News*, 32(1):60–65, 2001.

[15] Tobias Kappé, Paul Brunet, Jurriaan Rot, Alexandra Silva, Jana Wagemaker, and Fabio Zanasi. Kleene algebra with observations. In Wan J. Fokkink and Rob van Glabbeek, editors, *30th International Conference on Concurrency Theory, CONCUR 2019, August 27-30, 2019, Amsterdam, the Netherlands*, volume 140 of *LIPIcs*, pages 41:1–41:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[16] Dexter Kozen. Kleene algebra with tests and commutativity conditions. In Tiziana Margaria and Bernhard Steffen, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 14–33, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.

[17] Dexter Kozen. Results on the propositional μ-calculus. *Theoretical Computer Science*, 27(3):333–354, 1983. Elsevier.

[18] Dexter Kozen. Kleene algebra with tests. *ACM Trans. Program. Lang. Syst.*, 19(3):427–443, 1997.

[19] Dexter Kozen and Rohit Parikh. An elementary proof of the completeness of pdl. *Theoretical Computer Science*, 14(1):113–118, 1981.

[20] John W Lloyd. *Foundations of logic programming.* Springer Science & Business Media, 2012.

[21] Alain J. Mayer and Larry J. Stockmeyer. The complexity of pdl with interleaving. *Theoretical Computer Science*, 161(1):109–122, 1996.

[22] Dale Miller, Gopalan Nadathur, Frank Pfenning, and Andre Scedrov. Uniform proofs as a foundation for logic programming. *Annals of Pure and Applied Logic*, 51(1):125–157, 1991.

[23] Robin Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer, 1980.

[24] Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes, i. *Information and Computation*, 100(1):1–40, 1992.

[25] Fabrizio Montesi. *Choreographic Programming.* Ph.D. thesis, IT University of Copenhagen, 2013. https://www.fabriziomontesi.com/files/choreographic-programming.pdf.

[26] Fabrizio Montesi. *Introduction to Choreographies.* Cambridge University Press, 2023.

[27] David Peleg. Communication in concurrent dynamic logic. *Journal of Computer and System Sciences*, 35(1):23–58, 1987.

[28] David Peleg. Concurrent dynamic logic. *J. ACM*, 34(2):450–479, apr 1987.

[29] David Peleg. Concurrent program schemes and their logics. *Theoretical Computer Science*, 55(1):1–45, 1987.

[30] Todd Schmid, Tobias Kappé, and Alexandra Silva. A complete inference system for skip-free

guarded kleene algebra with tests. In Thomas Wies, editor, *Programming Languages and Systems - 32nd European Symposium on Programming, ESOP 2023, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2023, Paris, France, April 22-27, 2023, Proceedings*, volume 13990 of *Lecture Notes in Computer Science*, pages 309–336. Springer, 2023.

[31] Colin Stirling and David Walker. Local model checking in the modal mu-calculus. *Theoretical Computer Science*, 89(1):161–177, 1991.

[32] Philip Wadler. Propositions as types. *Commun. ACM*, 58(12):75–84, 2015.