MR2796004 (2012d:03063) 03B70 03B60 68Q55 68Q60

Bergstra, Jan A. (NL-AMST-IIF); **Ponse, Alban** (NL-AMST-IIF)

Proposition algebra. (English. English summary)

ACM Trans. Comput. Log. **12** (2011), *no.* 3, *Art.* 21, 36 *pp.*

A simple way of presenting information about an environment is by representing its basic facts by means of atomic propositions and combining them within propositional logic: to check whether some information holds simply amounts to evaluating its corresponding formula with respect to a valuation. In propositional logic, a valuation assigns the same value to different occurrences of the same atomic proposition in a single formula. However, in many contexts, such as sequential programs with side-effects, different occurrences correspond to different moments in time and should possibly return different values. This paper generalizes propositional logic by allowing valuations that evolve through time and may thus return different Boolean values for the same atomic proposition during the sequential evaluation of a formula. The resulting logic is studied under the name *proposition algebra*.

Among the class of all possible valuations, six varieties ranging from the *free* valuations satisfying no restrictions to the *static* valuations that correspond to propositional logic are singled out. The bulk of the paper is devoted to providing complete axiomatizations for the congruence notions associated to each of those varieties. Most interestingly, these axiomatizations are built using the ternary connective if-then-else; binary connectives are later shown to lack the necessary expressive power in this framework. The exposition ends with some remarks about the definition of infinite formulas which, due to the repetition of atomic propositions that may be evaluated differently, are more expressive than finite ones.

The paper is clearly written and, even though the details in Section 12 demand a lot of attention, is a nice read. As the authors acknowledge, there remains the challenge to find convincing examples for these free valuations as well as some discussion about the merits of proposition algebra with respect to other logics that could be of use to tackle the same problem, such as temporal or dynamic logic. *Miguel Palomino*

[References]

1. Baeten, J. C. M. and Weijland, W. P. 1990. *Process Algebra.* Cambridge Tracts in Theoretical Computer Science, vol. 18. Cambridge University Press, Cambridge. MR1102458
2. Belnap, N. D. 1977. A useful four-valued logic. In *Modern Uses of Multiple-Valued Logic,* J. Dunn and G. Epstein, Eds. D. Reidel, Dordrecht, 8–37. MR0485167
3. Bergstra, J. A., Bethke, I., and Ponse, A. 2007. Thread algebra and risk assessment services. In *Proceedings of the Logic Colloquium.* C. Dimitracopoulos, L. Newelski, and D. Normann, Eds. Cambridge Lecture Notes in Logic, vol. 28. 1–17. MR2395800
4. Bergstra, J. A., Bethke, I., and Rodenburg, P. H. 1995. A propositional logic with 4 values: true, false, divergent and meaningless. *J. Appl. Non-Classi. Logics 5,* 2, 199–218. MR1362876
5. Bergstra, J. A. and Klop, J. W. 1984. Process algebra for synchronous communication. *Inf. Cont. 60,* 1–3, 109–137. MR0764282
6. Bergstra, J. A. and Middelburg, C. A. 2007. Thread algebra for strategic interleaving. *Form. Asp. Comput. 19,* 4, 445–474.
7. Bergstra, J. A. and Ponse, A. 1998a. Bochvar-McCarthy logic and process algebra. *Notre Dame J. Form. Logic 39,* 4, 464–484. MR1776220
8. Bergstra, J. A. and Ponse, A. 1998b. Kleene's three-valued logic and process

algebra. *Inf. Proc. Lett. 67,* 2, 95–103. MR1638158

9. Bergstra, J. A. and Ponse, A. 2005. A bypass of Cohen's impossibility result. In *Proceedings of the European Grid Conference (EGC).* Lecture Notes in Computer Science, vol. 3470. Springer-Verlag, Berlin, 1097–1106.
10. Bergstra, J. A. and Ponse, A. 2007. Execution architectures for program algebra. *J. Appl. Logic 5,* 1, 170–192. MR2293336
11. Bloom, S. L. and Tindell, R. 1983. Varieties of "if-then-else". *SIAM J. Comput. 12,* 4, 677–707. MR0721007
12. Cohen, F. 1987. Computer viruses–theory and experiments. *Comput. Sec. 6,* 22–35. MR0932259
13. Cohen, F. 2001. Reply to 'Comment on "A Framework for Modelling Trojans and Computer Virus Infection."' *Comput. J. 44,* 4, 326–327.
14. Fokkink, W. J. 2000. *Introduction to Process Algebra.* Texts in Theoretical Computer Science, An EATCS Series. Springer-Verlag, Berlin. MR1764014
15. Hähnle, R. 2005. Many-valued logic, partiality, and abstraction in formal specification languages. *Logic J. IGPL 13,* 4, 415–433. MR2163140
16. Hayes, I. J., Jifeng, H., Hoare, C. A. R., Morgan, C. C., Roscoe, A. W., Sanders, J. W., Sorensen, I. H., Spivey, J. M., and Sufrin, B. A. 1987. Laws of programming. *Comm. ACM 3,* 8, 672–686. MR0899396
17. Hoare, C. A. R. 1985a. *Communicating Sequential Processes.* Prentice-Hall, Englewood Cliffs. MR0805324
18. Hoare, C. A. R. 1985b. A couple of novelties in the propositional calculus. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik 31,* 2, 173–178. MR0786294
19. Konikowska, B. 1996. Recursive functions of symbolic expressions and their computation by machine. *Fund. Inf. 26,* 2, 167–203. MR1398085
20. McCarthy, J. 1963. A basis for a mathematical theory of computation. In *Computer Programming and Formal Systems,* P. Braffort and D. Hirshberg, Eds. North-Holland, Amsterdam, 33–70. MR0148258
21. Mittelstaedt, P. 2004. Quantum logic and decoherence. *Int. J. Theoret. Phys. 43,* 6, 1343–1354. MR2103809
22. Ponse, A. and van der Zwaag, M. B. 2007. Belnap's logic and conditional composition. *Theoret. Comput. Sci. 388,* 1–3, 319–336. MR2368302
23. Ponse, A. and van der Zwaag, M. B. 2008. Risk assessment for one-counter threads. *Theory Computi. Syst. 43,* 563–582. MR2461285
24. Regenboog, B. C. 2010. Reactive valuations. MS thesis, University of Amsterdam (arXiv:1101.3132vl[cs.LO].)
25. Rehder, W. 1980. Quantum logic of sequential events and their objectivistic probabilities. *Int. J. Theoret. Physics 19,* 3, 221–237. MR0575077
26. Vu, T. D. 2008. Denotational semantics for thread algebra. *J. Logic Alg. Prog. 74,* 94–111. MR2378712

Note: This list reflects references listed in the original paper as accurately as possible with no attempt to correct errors.