Developments in concurrent Kleene algebra. [Zbl 1344.68149]

Summary: This report summarises the background and recent progress in the research of its co-authors. It is aimed at the construction of links between algebraic presentations of the principles of programming and the exploitation of concurrency in modern programming practice. The signature and laws of a Concurrent Kleene Algebra (CKA) largely overlap with those of a Regular Algebra, with the addition of concurrent composition and a few simple laws for it. They are re-interpreted here in application to computer programs. The inclusion relation for regular expressions is re-interpreted as a refinement ordering, which supports a stepwise contractual approach to software system design and to program debugging.

The laws are supported by a hierarchy of models, applicable and adaptable to a range of different purposes and to a range of different programming languages. The algebra is presented in three tiers. The bottom tier defines traces of program execution, represented as sets of events that have occurred in a particular run of a program; the middle tier defines a program as the set of traces of all its possible behaviours. The top tier introduces additional incomputable operators, which are useful for describing the desired or undesired properties of computer program behaviour. The final sections outline directions in which further research is needed.

MSC:
68Q70 Algebraic theory of languages and automata
68N30 Mathematical aspects of software engineering (specification, verification, metrics, requirements, etc.)
68Q85 Models and methods for concurrent and distributed computing (process algebras, bisimulation, transition nets, etc.)

Keywords:
concurrent Kleene algebra; laws of programming; trace algebra; semantic models; refinement; unifying theories

Software:
SystemC; Prover9; Mace4; Isabelle/HOL; PTSC

Full Text: DOI

References:
Kleene, S. C., Representation of events in nerve nets and finite automata, (Shannon, C. E.; McCarthy, J., Automata Studies, Zbl 1125.68039
Brink, C.; Rewitzky, I., A paradigm for program semantics: power structures and duality, (2001), CSLI Publications
Chandy, K. Mani; Misra, Jayadev, Systolic algorithms as programs, Distrib. Comput., 1, 3, 177-183, (1986) - Zbl 0669.68024
Conway, J. H., Regular algebra and finite machines, (Chapman and Hall, 1971), Zbl 0231.94041
Dijkstra, E. W., Guarded commands, nondeterminacy and formal derivation of programs, Commun. ACM, 18, 8, 453-457, (1975) - Zbl 0308.68017
Goltman, A.; Cook, B.; Parkinson, M. J.; Vafeiadis, V., Proving that non-blocking algorithms don’t block, (Shao, Z.; Pierce, B. C., POPL 2009, (2009), ACM), 16-28 - Zbl 1315.68093
Kleene, S. C., Representation of events in nerve nets and finite automata, (Shannon, C. E.; McCarthy, J., Automata Studies, (1956), Princeton University Press), 3-41
Kung, H.; Leiserson, C. E., Algorithms for VLSI processor arrays, (Mead, Carver; Conway, Lynn, Introduction to VLSI Systems, (1979), Addison-Wesley), 271-292


[57] Pratt, V. R., Some constructions for order-theoretic models of concurrency, (Parikh, R., Logic of Programs, LNCS, vol. 193, (1985), Springer), 269-283


[59] Roscoe, A. W., A mathematical theory of communicating processes, (1982), Oxford University, PhD thesis

[60] Salomaa, A., Two complete axiom systems for the algebra of regular events, J. ACM, 13, 1, 158-169, (1966) · Zbl 0149.24902


This reference list is based on information provided by the publisher or from digital mathematics libraries. Its items are heuristically matched to zbMATH identifiers and may contain data conversion errors. It attempts to reflect the references listed in the original paper as accurately as possible without claiming the completeness or perfect precision of the matching.