

An algebra for interaction of cyber-physical components Lion, B.

Citation

Lion, B. (2023, June 1). An algebra for interaction of cyber-physical components. Retrieved from https://hdl.handle.net/1887/3619936

Version: Publisher's Version

Licence agreement concerning inclusion of doctoral thesis License:

in the Institutional Repository of the University of Leiden

Downloaded from: https://hdl.handle.net/1887/3619936

Note: To cite this publication please use the final published version (if applicable).

Stellingen

behorende bij het proefschrift

An Algebra for Interaction of Cyber-Physical Components van Benjamin Lion

- 1. An algebra of interacting components with parametrized composition operators provides a uniform way of expressing diverse forms of interaction exhibited by cyber-physical systems: terms are components of the system, and parametrized operators capture how parts interact. Algebraic properties of operations, therefore, directly reflect properties of interaction. (Ch2)
- 2. Cyber-physical interactions are different in nature from cyber-cyber interactions, as they introduce unpredictability, missed events, and time dependent values. Sequences of observations and co-inductive constraints are tools that enable local modeling of constraints in cyber-physical interactions and global extension of those constraints on infinite sequences of observations. (Ch2, Ch3)
- 3. Rewriting logic is an expressive framework for executable specification of cyber-physical systems. Each part of a system is modeled as a separate rewrite theory which, after composition, collectively offer simulation and verification facilities. Maude is an expressive and powerful framework to run and analyse rewriting logic specifications. (Ch4, Ch5)
- 4. Discrete observations of physical media don't allow to capture all events. The safety analysis on sequences of observables of a module may not be sufficient to conclude that the cyber-physical system is safe: unsafe events may be unobserved by the program under verification. (Ch 5)
- 5. In the real world, it is crucial to study how a program affects its surrounding. Every executing program uses physical resources, and the effects that a program has on its environment may preempt the logic that the program computes.
- 6. Does there exist a cyber-physical Turing machine? Answering this question could bring new understanding on the relation between the theory of computability and physics.
- 7. Computer science is seen as a tool for most other sciences to analyse data via programming. The development of formal methods for programming is therefore crucial for producing stable scientific results and sound analysis.
- 8. Tools that automate the generation of (executable) content may change the job of a programmer from generating a program to verifying that the program has the desired behavior. In this paradigm, formal methods has a long expertise and might serve an important role.
- 9. Truth is usually highly valued as opposed to falsehood. However, from false, one can negate the result and obtain a true statement. Consequently, both truth and falsehood work towards unwinding general confusion. Silence, however, preserves the confusion between true and false.