1. Record Nr. UNINA9910830829303321

Titolo	Modeling and verification of real-time systems [[electronic resource]] : formalisms and software tools / / edited by Stephan Merz, Nicolas Navet
Pubbl/distr/stampa	London, : ISTE Hoboken, NJ, : John Wiley, 2008
ISBN	1-282-16492-9 9786612164927 0-470-61101-4 0-470-39359-9
Descrizione fisica	1 online resource (395 p.)
Collana	ISTE ; ; v.16
Classificazione	ST 170 ST 234
Altri autori (Persone)	NavetNicolas MerzStephan
Disciplina	004.01/51 004.0151
Soggetti	Real-time data processing Computer software - Verification Formal methods (Computer science)
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Description based upon print version of record.
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Modeling and Verification of Real-Time Systems: Formalisms and Software Tools; Contents; Preface; Chapter 1. Time Petri Nets - Analysis Methods and Verification with TINA; 1.1. Introduction; 1.2. Time Petri nets; 1.2.1. Definition; 1.2.2. States and the state reachability relation; 1.2.3. Illustration; 1.2.4. Some general theorems; 1.3. State class graphs preserving markings and LTL properties; 1.3.1. State classes; 1.3.2. Illustration; 1.3.3. Checking the boundedness property on-the-fly; 1.3.4. Variations; 1.3.4.1. Multiple enabledness; 1.3.4.2. Preservation of markings (only) 1.4. State class graphs preserving states and LTL properties1.4.1. Clock domain; 1.4.2. Construction of the SSCG; 1.4.3. Variants; 1.5. State class graphs preserving states and branching properties; 1.6.

Computing firing schedules; 1.6.1. Schedule systems; 1.6.2. Delays (relative dates) versus dates (absolute); 1.6.3. Illustration; 1.7. An implementation: the Tina environment; 1.8. The verification of SE-LTL formulae in Tina; 1.8.1. The temporal logic SE-LTL; 1.8.2. Preservation of LTL properties by tina constructions; 1.8.3. selt: the SE-LTL checker of Tina; 1.8.3.1. Verification technique
1.8.3.2. The selt logic1.9. Some examples of use of selt; 1.9.1. John and Fred; 1.9.1.1. Statement of problem; 1.9.1.2. Are the temporal constraints appearing in this scenario consistent?; 1.9.1.3. Is it possible that Fred took the bus and John the carpool?; 1.9.1.4. At which time could Fred have left home?; 1.9.2. The alternating bit protocol; 1.10. Conclusion; 1.11. Bibliography; Chapter 2. Validation of Reactive Systems by Means of Verification and Conformance Testing; 2.1. Introduction; 2.2. The IOSTS model; 2.2.1. Syntax of IOSTS; 2.2.2. Semantics of IOSTS; 2.3. Basic operations on IOSTS
2.3.1. Parallel product2.3.2. Suspension; 2.3.3. Deterministic IOSTS and determinization; 2.4. Verification and conformance testing with IOSTS; 2.4.1. Verification; 2.4.1.1. Verifying safety properties; 2.4.1.2. Verifying possibility properties; 2.4.1.3. Combining observers; 2.4.2. Conformance testing; 2.5. Test generation; 2.6. Test selection; 2.7. Conclusion and related work; 2.8. Bibliography; Chapter 3. An Introduction to Model Checking; 3.1. Introduction; 3.2. Example: control of an elevator; 3.3. Transition systems and invariant checking; 3.3.1. Transition systems and their runs
3.3.2. Verification of invariants3.4. Temporal logic; 3.4.1. Linear-time temporal logic; 3.4.2. Branching-time temporal logic; 3.4.3. -automata; 3.4.4. Automata and PTL; 3.5. Model checking algorithms; 3.5.1. Local PTL model checking; 3.5.2. Global CTL model checking; 3.5.3. Symbolic model checking algorithms; 3.6. Some research topics; 3.7. Bibliography; Chapter 4. Model Checking Timed Automata; 4.1. Introduction; 4.2. Timed automata; 4.2.1. Some notations; 4.2.2. Timed automata, syntax and semantics; 4.2.3. Parallel composition; 4.3. Decision procedure for checking reachability
4.4. Other verification problems

Sommario/riassunto

This title is devoted to presenting some of the most important concepts and techniques for describing real-time systems and analyzing their behavior in order to enable the designer to achieve guarantees of temporal correctness. Topics addressed include mathematical models of real-time systems and associated formal verification techniques such as model checking, probabilistic modeling and verification, programming and description languages, and validation approaches based on testing. With contributions from authors who are experts in their respective fields, this will provide the reader with