Runtime Monitoring for Contract Automata

Praxis der Forschung

Background Embedded systems are prelevant in many industry applications. For example, they appear as Programmable Logic Controllers (PLCs) controlling automated production systems and medical devices, or as controllers (motor control, automatic break systems, assisted driving, ...) in cars. These systems are often reactive and specially tailored to safety-critical real-time environments. A malfunction may cause severe damage to the system itself or to the payload, or even harm persons within the reach of the system. Verifying such systems formally is a worthwile goal.

Contract automata are a formalism that can be used to specify such reactive systems (given as software programs). They are a form of finite automata describing the behaviour of the reactive system through contracts (pre- and postcondition) at every step.

The validity of a program / system under a given contract automata can be verified statically using model-checking techniques. However, such static techniques suffer from bad scalability and often require interaction by the programmer or an expert logician or require complex code annotations. Even then, not all conforming programs can be verified like this, because e.g. their validity might rely on a complex run time condition that is not statically tangible. These issues limit the practical applicability of contract automata and verification techniques in general.

As an alternative runtime verification / monitoring techniques exist to observe a running system and report its adherance to a given specification / contract.

Goal The primary goals of this work are:

- Research existing solutions for runtime verification and asses their applicability to the presented problem.
- Develop and implement an approach for performing runtime monitoring of software programs based on contract automata, both in a simulated and production environment.

Your Profile Programming skills are required. You should be interested in model checking, temporal logics, automata theory and runtime verification. Interest in practical applicability is a plus. You should have completed the Formal Methods (Formale Systeme) course at KIT or equivalent.

Contact

Joshua Bachmeier j.bachmeier@fzi.de

FZI Research Center for Information Technology | SE-SEQ
KIT | KASTEL Research Group for Application-Oriented Formal Verification
Joshua Bachmeier | Email: j.bachmeier@fzi.de