BACHELOR'S THESIS – AUTOMATED DESIGN-SPACE EXPLORATION USING QUANTIFIER-FREE BIT-VECTOR LOGIC (M/F/D)

Context

With cyber-physical systems becoming increasingly complex w.r.t. both hardware and software, model-based systems engineering has proven successful in coping with the intricacy of such software-intensive systems. However, with the separation of hardware and software comes the problem of system integration, which due to the growing scale has become increasingly time-consuming and error-prone when performed manually. While efforts to automate this step have successfully applied design-space exploration to synthesize optimal deployments of tasks onto execution units, finding a scalable formulation of the NP-hard deployment problem using satisfiability modulo theories has proven itself to be challenging.

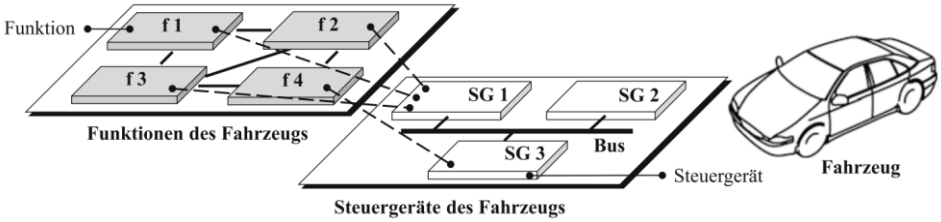

Figure 1: Illustration of the deployment of software components onto a vehicle's hardware platform (M. Fuchs et al.: Neues Rollenverständnis für die Entwicklung verteilter Systemverbunde in der Karosserie- und Sicherheitselektronik, *2001).*

Problem

The so-called deployment problem, consisting of the allocation of tasks onto execution units based on resource and safety constraints, is currently solved in AutoFOCUS 3 (https://af3.fortiss.org/main-features/design-space-exploration/) by formulating it as a satisfiability modulo theory (SMT) problem using free (uninterpreted) functions and quantifiers. This approach results in human-readable representations of the constraint-satisfaction problem and has been successfully applied to industry-size deployment problems. However, there is reason to expect considerable performance improvements by formulating SMT problems using solely quantifier-free bit-vector logic.

```
(declare-datatypes ((Task 0)) (((t1) (t2))))
(declare-datatypes ((Core 0)) (((c1) (c2))))

(declare-fun deployment (Task) Core)

(assert
    (forall ((t Task))
        (exists ((c Core))
            (= (deployment t) c)
        )
    )
)

(check-sat)
(eval (deployment t1))
(eval (deployment t2))
```

Basic deployment problem expressed using free (uninterpreted) functions and quantifiers.

```
(assert
  (let ((sum_deployment_ecu_01
      (bvadd
        (ite
          (= #b1 ((_ extract 0 0) deployment_ecu_01))
          #b01 #b00
        )
        (ite
          (= #b1 ((_ extract 1 1) deployment_ecu_01))
          #b01 #b00
        )
      )
  ))
  (= #b01 sum_deployment_ecu_01))
)
```

Excerpt of the basic deployment constraint expressed using only quantifier-free bit-vector logic.

Unlike propositional logic, first-order logic is undecidable, i.e. there is no decision procedure that can correctly determine whether an arbitrary decision problem is satisfiable or not. Hence, while SMT solvers offer such decision procedures for a combination of quantifier-free theories, once a problem includes quantifiers (and thereby first-order logic) the solver might not be able to determine its satisfiability. Moreover, the definition of uninterpreted functions over infinite sets leads to unbounded search spaces – again leading to undecidability and requiring sophisticated automated reasoning techniques instead of the powerful, yet efficient methods available for bounded model checking.

About fortiss

fortiss is the research institute of the Free State of Bavaria for software-intensive systems and services with headquarters in Munich. The institute currently employs around 180 employees, who collaborate on research, development and transfer projects with universities and technology companies in Bavaria, Germany and Europe. Research is focused on state of the art methods, techniques and tools of software development, systems & service engineering and their application to reliable, secure cyber-physical systems, such as the Internet of Things (IoT). fortiss has the legal structure of a non-profit limited liability company (GmbH). Its shareholders are the Free State of Bavaria (as majority shareholder) and the Fraunhofer Society for the Promotion of Applied Research. www.fortiss.org

Your tasks:

- Introduction to the concepts of DSE (Design-Space Exploration), SMT (Satisfiability Modulo Theory.
- Implementation of the quantifier-free bit-vector-based DSEML-to-SMT translator in AutoFOCUS 3.
- Evaluation of the approach by comparison with the existing translations based on uninterpreted functions.

Your profile:

- You are a computer science (or similar) bachelor's student.
- You enjoy solving challenging problems.
- You have practical experience with Python and Java development, software versioning (Git in particular).
- You have basic knowledge of logic in the context of computer science (such as propositional and first-order logic).
- You have excellent communication skills in English.

Did we catch your interest?

Please submit your application with a motivational statement, a detailed CV and a current transcript of records to career@fortiss.org.

Notice that this code needs to be mentioned in the subject line of your application:

Job-ID: MbSEP-IDEA-BAM-01-2019

Contact: Tiziano Munaro