technische universität
dortmund

Fakultät für Informatik
Lehrstuhl XIV – Software Engineering
Prof.Dr. Falk Howar
Abschlussarbeit

Abschlussarbeit

Implementierung und Verifikation eines Software-Systems in OCaml mit Session Types.

Ansprechpartner: Richard Stewing (richard.stewing@tu-dortmund.de)

Situation

Software-Verifikation ist eines der klassischen Probleme der Informatik. Ein viel diskutierter Lösungsansatz sind Type-Systeme um Eigenschaften von Programmen zu kodieren. Session Types kodieren das Kommunikationsverhalten von nebenläufigen Prozessen und können interpretiert werden als kommunizierende, endliche Zustandsmaschinen. Auf die entsprechenden Zustandsmaschinen können dann Model-Checking Ansätze angewendet werden, um weitergehende Eigenschaften der Systeme zu zeigen. Im Rahmen der Abschlussarbeit soll ein Software-System modelliert und in OCaml mit Session Types umgesetzt werden.

Aufgabe

Die Abschlussarbeit schließt folgende Aufgaben ein:

Auswahl eines Software-System In Zusammenarbeit mit dem Betreuer wird ein Software-System von geeigneten Umfang ausgewählt. (Dieser Schritt erfolgt vor Anmeldung der Arbeit.)

Spezifikation Die Spezifikation des Software-Systems leistet der Studierende in Eigenarbeit. Der Studierende entwickelt mit Blick auf die Literatur geeignete kommunizierende, endliche Zustandsmaschinen. Außerdem werden Eigenschaften spezifiziert, die in dem Modell erfüllt werden sollen.

Implementierung Das Software-System wird durch den Studierenden in OCaml mit Session Types umgesetzt. Die Typen kodieren die endlichen Zustandsmaschinen.

Verifikation Die zuvor spezifizierten Eigenschaften werden durch Model-Checking verifiziert. Eine Auswahl der notwendigen Theorien erfolgt in Absprache zwischen Betreuer und Studierendem.

Voraussetzungen

Für die Arbeit sind gute Programmierkenntnisse von Nöten. Insbesondere der Umgang mit funktionalen Programmiersprachen (z.B. Haskell) sollte bekannt sein. Kenntnis von OCaml ist nicht notwendig, aber von Vorteil. Vorkenntnisse im Bereich des Model-Checkings sind ebenfalls hilfreich.

Links und Literatur

- „Foundations of Session Types and Behavioural Contracts" von Hüttel et. al.
- „On Communicating Finite-State Machines" von Brand und Zafiropulo
- `https://ocaml.org`
- „`kmclib`: Automated Inference and Verification of Session Types from OCaml Programs" von Imai et. al.