

発表概要

混合選択とタイムアウトで拡張された マルチパーティセッション型の軽量検査

今井 敬吾^{1,a)}

2021年1月13日発表

本発表は、混合選択 (mixed choice) とタイムアウトによるマルチパーティセッション型 (MPST) の拡張とその正しさを議論する。MPST は、多者間の通信プログラムのデッドロックフリー性をプロトコルに基づく型付けにより保証する枠組みである。これまでの多くのセッション型では、外部選択 (受信) と内部選択 (送信) を通信プロトコルの 1 つの状態に混在させることはできず、表現力が低かった。混合選択により、受信メッセージが一定期間到達しなかった場合にタイムアウトを送信するといった振舞いを扱えるようになる。本手法は構文的で軽量な方法により混合選択を実現するので、モデル検査のような計算量の大きい外部ツールを必要としない点が新しく、また大きな特徴である。この軽量性を活かし、OCaml における通信ライブラリ OCaml-MPST の混合選択とタイムアウト拡張によるを示す。拡張の正しさの証明の方針として、型の構文的検査が従来の通信オートマトン (CFMS) における multiparty compatible の拡張されたクラスに相当することと、これがデッドロックフリー性を満たすことを議論する。

Presentation Abstract

Lightweight Checking of Multiparty Session Types with Mixed Choice and Timeout

KEIGO IMAI^{1,a)}

Presented: January 13, 2021

Multiparty session types (MPST) are a type system that can ensure the safety of communication programs including deadlock-freedom. A mixed choice is a communication behaviour that mixes external choice (input) and internal choice (output) in a communication step, which is useful to handle various behaviours such as sending a timeout when a message is not received for a while. However, most MPST theories do not allow mixed choices, limiting the expressiveness of communication patterns. In this presentation, we discuss an extension of multiparty session types (MPST) with mixed choice and timeout, in a “syntactic” form, i.e. without resorting to state exploration like in model checking algorithms. To the author’s knowledge, a syntactic checking of mixed choice is not investigated. Taking advantage of its lightweight form, we implement the mixed choice and timeout in OCaml-MPST, our communication library for OCaml. We discuss the new syntactic checking of MPST well-formedness corresponds to an extended class of multiparty compatible in communication automata (CFMS), which also guarantees deadlock-freedom.

This is the abstract of an unrefereed presentation, and it should not preclude subsequent publication.

¹ 岐阜大学

Gifu University, Gifu 501-1112, Japan

^{a)} keigo@gifu-u.ac.jp