

Priorities for CP and GV

Ornela Dardha

School of Computing Science, University of Glasgow, United Kingdom

In this talk we will explore the world of priorities and how they can be added to CP—a session-typed process calculus based on linear logic— and to GV—a session-typed linear concurrent lambda calculus, in order to guarantee deadlock freedom for more expressive programs in these two languages.

Originally, both CP and GV guarantee deadlock freedom by restricting communication to a tree topology. To achieve this, CP combines channel creation and parallel composition under the cut rule, and GV combines channel creation and thread spawning under the fork operation.

We present PCP and PGV, which respectively extend CP and GV, by allowing cyclic topology of communication to be typed. To guarantee deadlock freedom we then use priorities, which are natural numbers capturing the order of operations in a program, thus ruling out bad interleavings. Consequently, we are able to decouple channel creation from parallel composition or thread spawning in CP and GV.