Linear Logic, Adjoint Logic And Session Types

Cut Elimination, Substructural Rules, Linear Logic, Adjoint Logic, Message Passing, Sessions, Scribble

[Sample: link hub] Data types have played a pivotal role in software engineering for the last 50 years and many believe session types will have an equally profound effect for the next 50. Software platforms are getting more complicated, larger, have more moving parts and errors are more costly to remedy. It is clear that the software engineering community needs more robust tooling and modern mathematics can certainly help.

A number of potentially very useful mathematical logics have already been defined, the most interesting from a communications and concurrency perspective is linear logic. Multiple logics can be combined using modal operators based on adjoint logic. These can be used where we need to mathematically represent and accurately reason about processes/threads and their concurrency and message passing constructs and use-once or other richer semantics.

Due to its many practical usage scenarios, the juxtaposition of mathematics, concurrency and communication is a very active research area. We are beginning to see these ideas have an impact on new specialist tooling and extensions to existing frameworks.

	Contents of One-Day Training Course	
Target Audience Mathematicians and software engineers interested in taking a rigorous mathematical approach to concurrency and communication **Prerequisites** Good knowledge of the fundamentals of mathematical logic.	**High Level Goals** Deadlock freedom during communication Distributed garbage collector Shared memory at scale Mathematically sound concurrency **Preliminaries** Cut elimination Sequent calculus What is a sequent? The idea of a process calculus (e.g. π-calculus) **Evolution of Propositions As Types** "propositions as session types, proofs as processes, and cut elimination as communication" [link] **Substructural Logic** Substructural logic – tweaking with the structural rules, what effect that brings? **Linear Logic** What does it describe? Mathematically describe a process (a thread of activity) as a series of steps Mathematical duals In contrast to normal mathematical logic, which focuses on truth (constant), linear logic focuses on resources (e.g. use once) Highly useful for writing deadlock-free code and communication specifications **Adjoint Logic: Intro** Fundamentals of adjoint logic Practical uses Review of modal operators Adjoint pairs of modal operators	**Adjoint Logic: Details** Relationship to weakening and contraction Combining a variety of logics, such as LNL, linear, s4, lax, ... Modes: * linear, * affine, * strict, * unrestricted **Message Passing** Interesting: message passing interpretation Using adjoint logic to structure messaging **Session types** Practical uses of session types in modern programming - intro type discipline to code Managing multiple communicating and concurrent sessions Binary and multi-party **Scribble** "Scribble is a language to describe application-level protocols among communicating systems. " [link] How Scribble works Generating Finite State Machines (FSMs) on both sides **Usage** Some of these ideas are being to be used in real products and development tooling Survey of what is available and how to use **Project** Expanding on practice topics explored earlier with a project looking at practical deployment of these ideas in production