

Knowist Link Hub

Linear Logic, Adjoint Logic And Session Types

<https://www.knowist.ac>

After attending our training course, it is recommended attendees study these primary sources and additional content to help deepen their knowledge.

Papers by Frank Pfenning et al.

Two papers by: Klaas Pruiksma / Frank Pfenning / William Chargin / Jason Reed:

<https://www.cs.cmu.edu/~fp/papers/adjoint18.pdf>

Adjoint Logic and Its Concurrent Operational Interpretation

“Adjoint logic provides a schematic way to combine multiple logics, some of which may be substructural, through modal operators that are adjoint to each other.”

<https://www.cs.cmu.edu/~fp/papers/adjoint18b.pdf>

Adjoint Logic

“We show that multiple intuitionistic logics with varying structural properties among weakening and contraction can be combined conservatively by adjoint pairs of modal operators. .. We provide three different formulations of adjoint logic and show their equivalence: one with explicit structural rules, a second with implicit structural rules, and a third with focused rules. The first two provide the foundation for proofs-as-programs interpretations, while the third is well-suited for logic programming and logical frameworks”

A paper by: Klaas Pruiksma and Frank Pfenning:

<https://arxiv.org/abs/1904.01290#>

A Message-Passing Interpretation of Adjoint Logic

“We present a system of session types based on adjoint logic which generalize standard binary session types. Our system allows us to uniformly capture several new behaviors in the space of asynchronous message-passing communication, including multicast, where a process sends a single message to multiple clients, replicable services, which have multiple clients and replicate themselves on-demand to handle requests from those clients, and cancellation, where a process discards a channel without communicating along it. We provide session fidelity and deadlock-freedom results for this system, from which we then derive a logically justified form of garbage collection.”

A paper by: Stephanie Balzer, Bernardo Toninho and Frank Pfenning

<http://ctp.di.fct.unl.pt/~btoninho/esop19.pdf>

Manifest Deadlock-Freedom for Shared Session Types

“Shared session types generalize the Curry-Howard correspondence between intuitionistic linear logic and the session-typed π -calculus with adjoint modalities that mediate between linear and shared session types, giving rise to a programming model where shared channels must be used according to a lock discipline of acquire-

release. .. In this paper, we develop a type system for logically-shared sessions in which types capture not only the interactive behavior of processes but also constrain the order of resources (i.e., shared processes) they may acquire. This typelevel information is then used to rule out cyclic dependencies among acquires and synchronization points, resulting in a system that ensures deadlock-free communication for well-typed processes in the presence of shared sessions, higher-order channel passing and recursive processes”

nCatLab

<https://ncatlab.org/nlab/show/adjoint+logic>

Adjoint Logic

“Adjoint logic or adjoint type theory is formal logic or type theory which natively expresses adjunctions of modal operators, adjoint modalities.”

<https://ncatlab.org/nlab/show/modal+type+theory>

modal type theory

“Modal type theory is a flavor of type theory with type formation rules for modalities, hence type theory which on propositions reduces to modal logic. .. modal type theory is specifically understood as being a type theory equipped with (co-)monads on its type system, representing the intended modalities.”

<https://ncatlab.org/nlab/show/linear+logic>

Linear Logic

“Linear logic is sometimes thought of as being a logic for arguing about resource sensitive issues, but it can also be thought of categorically .. . A bit more formally: Linear logic is a substructural logic in which the contraction rule and the weakening rule are omitted, or at least have their applicability restricted.”

Dan Licata & Mike Shulman

<http://dlicata.web.wesleyan.edu/pubs/ls15adjoint/ls15adjoint.pdf>

Adjoint Logic with a 2-Category of Modes

“We generalize the adjoint logics of Benton and Wadler (1996) and Reed (2009) to allow multiple different adjunctions between the same categories. This provides insight into the structural proof theory of cohesive homotopy type theory, which integrates the synthetic homotopy theory of homotopy type theory with the synthetic topology of Lawvere’s axiomatic cohesion.”

Stanford

<https://plato.stanford.edu/entries/logic-linear/>

Linear Logic

“Linear logic is a refinement of classical and intuitionistic logic. Instead of emphasizing truth, as in classical logic, or proof, as in intuitionistic logic, linear logic emphasizes the role of formulas as resources. To achieve this focus, linear logic does not allow the usual structural rules of contraction and weakening to apply to all formulas but only those formulas marked with certain modals.”

Wadler

Long list of topics: <http://homepages.inf.ed.ac.uk/wadler/topics/linear-logic.html>

<http://homepages.inf.ed.ac.uk/wadler/papers/propositions-as-types/propositions-as-types.pdf>

Propositions as Types

"propositions as types / proofs as programs /
simplification of proofs as evaluation of programs"

<http://homepages.inf.ed.ac.uk/wadler/papers/propositions-as-sessions/propositions-as-sessions-jfp.pdf>

Propositions as sessions

"propositions as session types / proofs as processes
/ cut elimination as communication."

"this paper presents CP, a calculus, in which propositions of classical linear logic correspond to session types. .. this paper presents GV, a linear functional language with session types, and a translation from GV into CP. The translation formalises for the first time a connection between a standard presentation of session types and linear logic, and shows how a modification to the standard presentation yields a language free from races and deadlock, where race and deadlock freedom follows from the correspondence to linear logic"

ABCD

<https://groups.inf.ed.ac.uk/abcd/>

From Data Types to Session Types

"Concurrency and distribution are computing's most pressing problem. .. The data type is one of computing's most successful ideas. .. Session types will play a crucial role in all aspects of software. Today, architects model systems using types that are directly supported in the programming language, whereas they model communications using protocols that have no direct support in the programming language; tomorrow, they will model communication using session types that are directly supported in the programming language. "

<http://groups.inf.ed.ac.uk/abcd/session-implementations.html>

Session Types in Programming Languages: A Collection of Implementations

<http://groups.inf.ed.ac.uk/abcd/meeting-december2017/index.html>

Interesting collection of talks (slides+papers)

University Of Glasgow

http://www.dcs.gla.ac.uk/research/betty/summerschool2016.behavioural-types.eu/programme/DardhaIntroBST.pdf/at_download/file.pdf

Introduction to Session Types

- In complex distributed systems communicating participants agree on a protocol to follow, specifying type and direction of data exchanged.
- Session types are a type formalism used to model structured communication-based programming.

- Guarantee privacy, communication safety and session fidelity.”

Scribble

<http://www.scribble.org/>

The Scribble Language

“Scribble is a language to describe application-level protocols among communicating systems. A protocol represents an agreement on how participating systems interact with each other. Without a protocol, it is hard to do meaningful interaction: participants simply cannot communicate effectively, since they do not know when to expect the other parties to send data, or whether the other party is ready to receive data. However, having a description of a protocol has further benefits. It enables verification to ensure that the protocol can be implemented without resulting in unintended consequences, such as deadlocks.”