Approaches to Mechanising Behavioural Types

António Ravara
with the contributions in the spreadsheet

NOVA-LINCS and Dep of Informatics
School of Science and Technology, NOVA University of Lisboa, PT
Univ of Malta, Malta

June 4, 2020

Representations of Binders

Five approaches

- de Bruijn indices
- polymorphic HOAS
- proof-assistant binders / separation logic
- nominal sets
- syntactic

de Bruijn indices

Common cons: not necessarily "human-readable"
(should we care? it's to be checked by machines...)

- type level (Uma & Ornela)
  well-scoped by construction, easily mechanised
- tree-shaped (James & Robert)
  metatheory just like STλC (well-known)
- co-indices (Robert)
  simple in the linear case, readily translated to BCI combinators
- Agda-binders in types (Luca[2])
  + just Agda functions - heavy notation, limited inference
- proof of context membership in types (Edwin)
  + well-scoped by construction, type-driven implementation of context manipulation - sometimes costly at compile-time

Other Representations of Binders

- polymorphic HOAS (Uma & Ornela – why another approach?)
  + transparent to the user
  - cannot reason about in host language
- proof-assistant binders (Jonas et al.)
  + Binders like HOAS, no meta theory needed ("apart" from SL)
  - semantic approach
- nominal sets (Kirstin)
  + binders treated just like in paper proofs
  - requires equivariance proofs, not "easily" portable
- syntactic (Petros / Antonio et al.)
  + quick & easy, no libraries required
  - nightmare to reason about

Dealing with linearity - I

(almost) as many approaches as entries in the sheet...

- leftover typing on partial commutative monoids (Uma & Ornela)
  + no extrinsic context splits
  - needs to be cancelative
- predicate on processes with polymorphic channels (Uma & Ornela)
  + trivial to define
  - user loses access to channel type info
- semiring usage annotations (James & Robert / Lucas)
  + standard linear algebra
  - no clear general approach, users have to supply proofs
- linear co-de Bruijn indices (Robert)
  + no algebra
  - (no algebra) only syntactic linearity, user supplies permutations

Dealing with linearity - II

(almost) as many approaches as entries in the sheet...

- separation logic (Iris) (Jonas et al.)
  + resource reasoning encapsulated
  - unclear how to do other properties than safety
- quantitative type theory (Edwin)
  + session types definable directly
  - "read only"operations are syntactically noisy
- linear logic (Petros)
  + correct-by-construction
  - allowed patterns severely limited
- addition relation + proof of well-formedness (Antonio et al.)
  + operations on types are partial functions, natural to implement
  - operations on type environments require proofs; non-determinism in the type system complicates soundness proofs

Questions / Challenges

Are each of the approaches scalable?

- from linear types to binary session types
- from binary to multiparty session types
- from protocol to functional correctness
- from safety to liveness properties
- ...

How to relate these approaches?

- Encodings?
- Sharing definitions and results
- Characterisations to find the right setting to the problem of interest
- ...