Computational (Higher and Lower) Type Theory

Robert Harper

MURI Meeting October 15, 2021

New and Coming PhD's

Evan Cavallo defended February, 2021, post-doc at Stockholm w/Mörtberg.
Higher Inductive Types and Parametricity in Cubical Type Theory.

Jon Sterling defended September, 2021, post-doc at Aarhus w/Birkedal.
First Steps in Synthetic Tait Computability.

Yue Niu proposing soon.
Cost-Aware Logical Framework

Two Themes

Synthetic Tait Computability

- Jon: Normalization for Cubical Type Theory.
- Yue: Cost-Aware Logical Framework.
- [Jon and B: Parametricity for Program Modules.]

Parametricity = Relational Invariance

- Carlo: Data Abstraction and Univalence.
- Evan: Higher Parametricity and Cohesion.

First Steps in STC: Normalization for Cartesian Cubical Type Theory.

- Normal cubes are unstable under substitution: eg compose to x, or endpoints.
- Universes require structure, not property.
- Synthetic separation: open/closed modalities $\bigcirc_{\mathsf{syn}} A$, $\bullet_{\mathsf{syn}} A$ corresponding to a subterminal syn.

Direct application to CoolTT implementation of CCTT.

- Practical decision method for judgmental equality.
- cf Favonia and Reed's talks.

Synthetic Parametricity for Program Modules

Sterling and H: J.ACM October 2021

Second steps in STC: Rep Ind for Program Modules. [cf Crary]

- Program modules are phase separated:
  - Static: types $=$ elements of universe.
  - Dynamic: programs as elements of types.
- Dependent types with $\mathcal{U}$, Σ, Π, plus
  - Extension types: $\{\, A \mid \mathsf{stat} \hookrightarrow V \,\}$. (cf Riehl and Shulman, Stone and H)
  - Open/closed modalities: $\bigcirc_{\mathsf{stat}}\, A$, $\bullet_{\mathsf{stat}}\, A$.

Parametricity structures:

- Internal language of topos constructed by glueing.
- Phase-separated, proof-relevant logical relations.
- Open/closed modalities: $\bigcirc_{\mathsf{syn}}\, A$, $\bullet_{\mathsf{syn}}\, A$.

Two, Three, Many Phase Distinctions

Sterling and H 2021 MLW; Niu, et al PoPL 2022

Phase distinctions abound!

- Syntax/semantics: computability structures / glueing.
- Static/dynamic: type checking vs execution time.
- Development/compilation: respecting/violating abstraction.
- Public/private: ensuring confidentiality/integrity.
- Extension/intension: behavior/resource usage.

Calf: Cost-Aware Logical Framework

Niu, Sterling, Grodin, and H: PoPL 2022

Internal/instrinsic account of both behavior and cost.

- Extensional: $\mathtt{mergesort} = \mathtt{inssort} : \mathtt{seq} \to \mathtt{seq}$.
- Intensional:
  - $\mathtt{mergesort} : \mathtt{seq} \xrightarrow{n \lg n} \mathtt{seq}$.
  - $\mathtt{inssort} : \mathtt{seq} \xrightarrow{n^2} \mathtt{seq}$.

Idea: use synthetic methods to isolate extension from intension.

- Open modality $\bigcirc_{\mathsf{ext}}\, A$: disregard cost.
- Closed modality $\bullet_{\mathsf{ext}}\, A$: isolate cost.

Notion of cost is abstract!

- Number of comparisons, number of hcom's (potentially).
- Steps of execution: adequacy wrt operational semantics (cf. Niu and H 2021)

Representation Independence and Univalence

Angiuli, Cavallo, Mörtberg, Zeuner: PoPL 2021

Reynolds: relational invariance / parametricity ensures representation independence.

- eg, relate two implementations of queues.
- show that operations preserve relation.
- Thm: no client can distinguish them.

Smells a lot like Structure Identity Principle!

- Use quotients via HIT's to represent simulation relation as an equivalence.
- Apply univalence to ensure interchangeability in all contexts.

Parametricity and Cohesion

Cavallo and H: Logical Methods in Computer Science; Cavallo thesis

Cubical, proof-relevant parametricity expresses relational invariance:

- Bridges correspond to paths.
- Relativity corresponds to univalence.

Associativity of smash product is readily proved via parametricity.

- Relational invariance gives rise to paths.
- But it is a strong requirement (natural for computing though).

Cohesive modalities distinguish pointwise from parametric, and allow transfer of paths.

References

Carlo Angiuli, Evan Cavallo, Anders Mörtberg, and Max Zeuner. Internalizing representation independence with univalence. *Proceedings of the ACM on Programming Languages*, 5(POPL), January 2021. doi: 10.1145/3434293.

Evan Cavallo. *Higher Inductive Types and Internal Parametricity for Cubical Type Theory*. PhD thesis, Carnegie Mellon University, 2021.

Evan Cavallo and Robert Harper. Internal Parametricity for Cubical Type Theory. In Maribel Fernández and Anca Muscholl, editors, *28th EACSL Annual Conference on Computer Science Logic (CSL 2020)*, volume 152 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 13:1–13:17, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-95977-132-0. doi: 10.4230/LIPIcs.CSL.2020.13. URL `https://drops.dagstuhl.de/opus/volltexte/2020/11656`.

Evan Cavallo and Robert Harper. Internal parametricity for cubical type theory. To appear, *Logical Methods in Computer Science*, 2021.

Yue Niu and Robert Harper. Cost-aware type theory. Manuscript, 2020.

Yue Niu, Jonathan Sterling, Harrison Grodin, and Robert Harper. A cost-aware logical framework. To appear, POPL 2022., 2021.

Jonathan Sterling. *First Steps in Synthetic Tait Computability: The Objective Metatheory of Cubical Type Theory*. PhD thesis, Carnegie Mellon University, 2021. Forthcoming.

Jonathan Sterling and Carlo Angiuli. Normalization for cubical type theory. In *Proceedings of the 36th Annual ACM/IEEE Symposium on Logic in Computer Science*, New York, NY, USA, 2021. Association for Computing Machinery. doi: 10.1109/LICS52264.2021.9470719.

Jonathan Sterling and Robert Harper. Logical Relations As Types: Proof-Relevant Parametricity for Program Modules. *Journal of the ACM*, 6(1), October 2021.

Jonathan Sterling, Carlo Angiuli, and Daniel Gratzer. A cubical language for Bishop sets. Under review, 2020.