**A DECENTRALIZED APPROACH TO RELEASE/ACQUIRE CONCURRENCY**

Yotam Dvir, Tel Aviv University, yotamdvir.github.io

**GOAL** Moggi-style Brookes semantics for the Release/Acquire relaxed memory model

**NEW CHALLENGES ABOUND**
- More abstract and nuanced traces
- More closure rules
- First-class parallelism with causal propagation

**JUSTIFIED TRANSFORMATIONS**
- Laws of Parallel Programming
  - Symmetry: $M \parallel N \leftrightarrow N \parallel M$
  - Generalized Sequencing: $\langle \alpha, (M \parallel N) \rangle \leftrightarrow \langle \alpha, N \parallel M \rangle$
- Eliminations
  - Irrelevant Read: $\langle \alpha, \ell \parallel \ell \rangle \leftrightarrow \langle \alpha, \ell \parallel \ell \rangle$
  - Write-Read: $\langle \alpha, \ell \parallel \ell \rangle \leftrightarrow \langle \alpha, \ell \parallel \ell \rangle$
  - Write-FAA: $\langle \alpha, \ell \parallel \ell \parallel \ell \rangle \leftrightarrow \langle \alpha, \ell \parallel \ell \parallel \ell \rangle$
  - Read-Read: $\langle \alpha, \ell \parallel \ell \rangle \leftrightarrow \langle \alpha, \ell \parallel \ell \rangle$
  - Read-FAA: $\langle \alpha, \ell \parallel \ell \parallel \ell \rangle \leftrightarrow \langle \alpha, \ell \parallel \ell \parallel \ell \rangle$
  - FAA-Read: $\langle \alpha, \ell \parallel \ell \parallel \ell \rangle \leftrightarrow \langle \alpha, \ell \parallel \ell \parallel \ell \rangle$
  - FAA-FAA: $\langle \alpha, \ell \parallel \ell \parallel \ell \rangle \leftrightarrow \langle \alpha, \ell \parallel \ell \parallel \ell \rangle$
- Others
  - Irrelevant Read Introduction: $\langle \alpha, \ell \parallel \ell \rangle \leftrightarrow \langle \alpha, \ell \parallel \ell \rangle$
  - Read to FAA: $\langle \alpha, \ell \parallel \ell \rangle \leftrightarrow \langle \alpha, \ell \parallel \ell \rangle$
  - Write-Read Deorder: $\langle \ell \parallel \ell \parallel \ell \rangle \leftrightarrow \langle \ell \parallel \ell \parallel \ell \rangle$
  - Write-Read Reorder: $\langle \ell \parallel \ell \parallel \ell \rangle \leftrightarrow \langle \ell \parallel \ell \parallel \ell \rangle$

**RA state invariants, e.g.**
- $\text{view } w \text{ point to msg } \sigma \Rightarrow \text{view } \leq \sigma$
- Admissible step: $\text{ADVANCE } \Rightarrow \text{pretend to load}$

**Trace-based Denotational Semantics**
- By Brookes
- Sequences of guarantees to/from the environment
-riangle guarantees $\Rightarrow$ return
- Denotations
- Sequences of guarantees closed under REWRITE RULES

**REWRITE RULES**
1. **Concrete:**
   - Without only the CONCRETE rules the traces have an operational interpretation
   - Mumble (Mu) omits a guarantee and relies on it internally

2. **Abstract:**
   - Challenge: non-operational traces. Solution: percolate ABSTRACT rules out to induce

**MAIN RESULTS**
A denotational semantics for Release/Acquire based on linear traces that is:
- Standard (monad base, truly compositional)
- Adequate
- Abstract (supports known transformations)

**Monad-based Denotational Semantics**
- Moggi
- Modular framework for effectful semantics
- $\text{[(k := 1 ; m := 1)] } = \text{[(k := 1 ; m := 1)] } \equiv \text{[(k := 1 ; m := 1)] }$ (monad base)
- $\text{[(k := 1 ; m := 1)] } \equiv \text{[(k := 1 ; m := 1)] }$ (compositionality)

**Release/Acquire Interleaving Semantics**
- Fragment of the C/C++ model of causal propagation

**Brookes semantics**
- Moggi-style
- Denotational Semantics
- Moggi semantics
- Effects denote monads
- Brookes semantics
- Traces denote behaviors

**Relaxed memory**
- Weakly consistent concurrent shared state
- Weakly consistent concurrently shared state
- RA state invariants, e.g.