Burrow

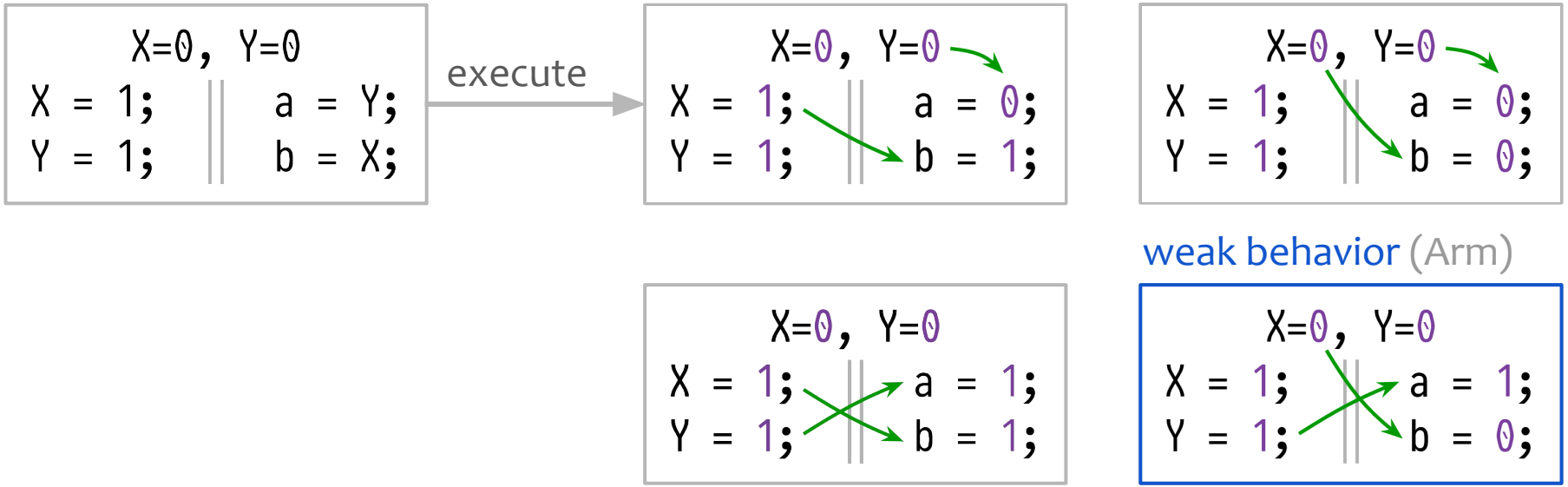
A Proof Framework for Weak Memory

Dennis Sprokholt, Soham Chakraborty

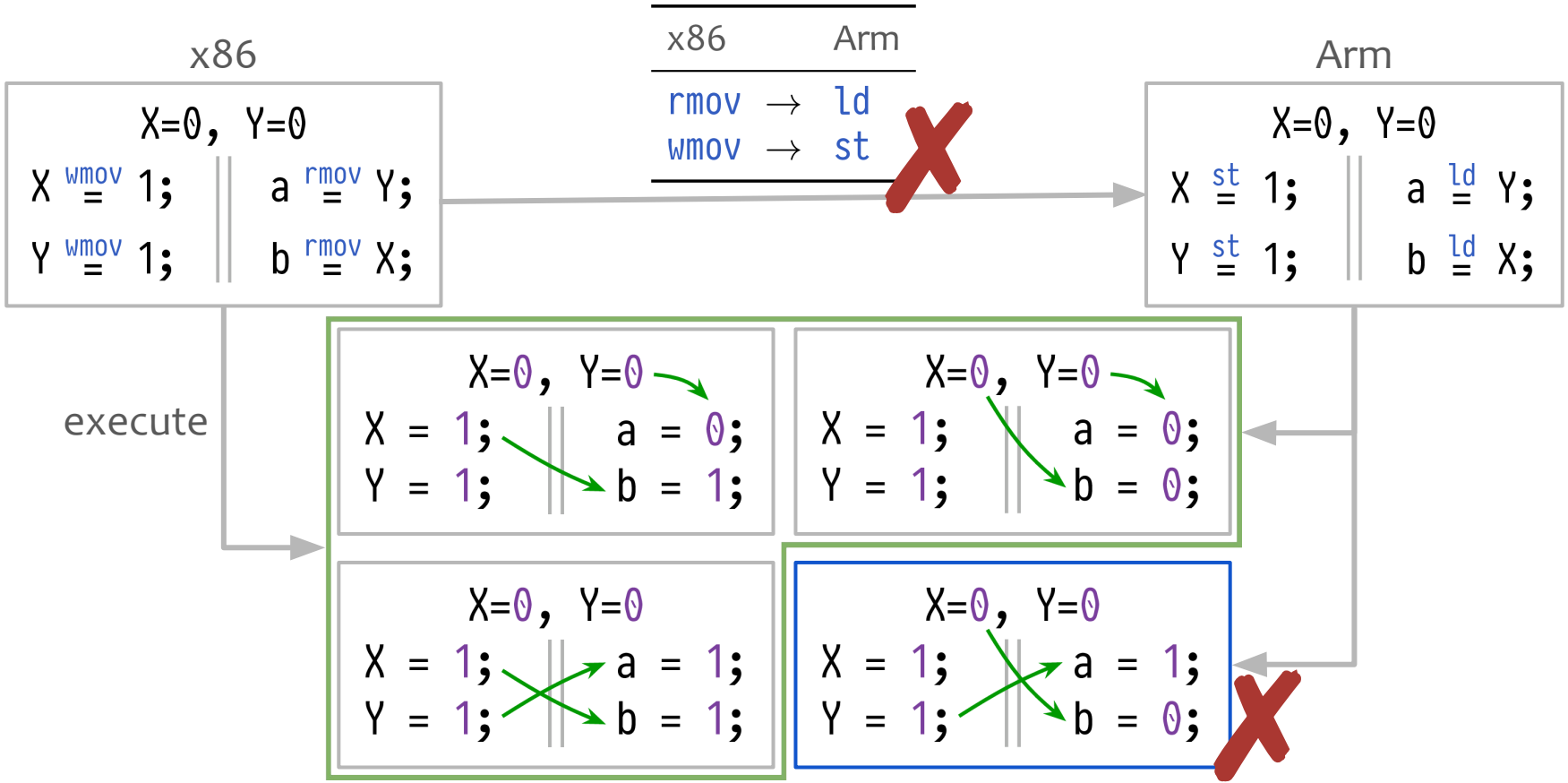
CAV 2026, Lisbon

Weak Memory

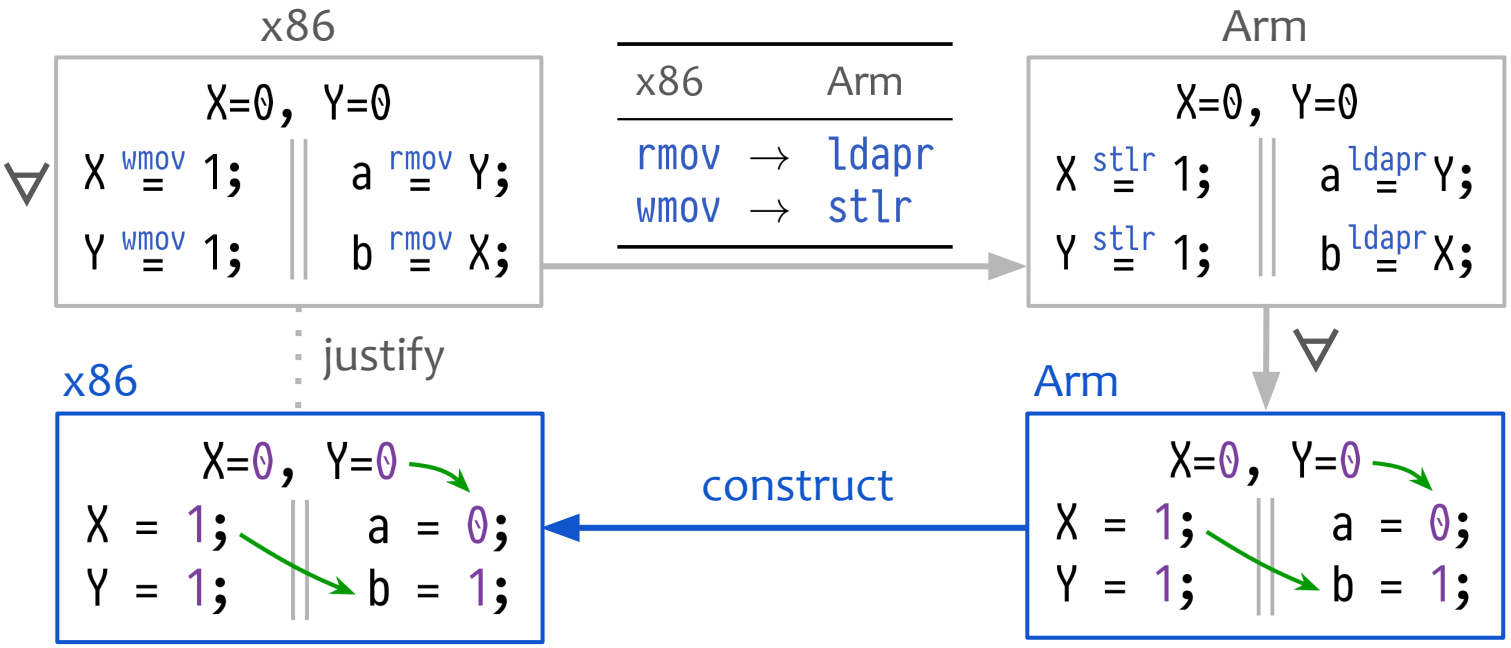
CPUs execute instructions out-of-order



Weak Memory Mapping (Incorrectly)



Proving Weak Memory Mappings



Ideally, mechanized proof in a **proof assistant**

Problem

Mechanizing these proofs is *challenging* and *time-consuming*

1. For each mapping, programmers **reprove many properties**

2. Proof assistant must be **explicitly told** that properties transfer along mapping

Solution

Burrow: A Proof Framework for Weak Memory

1. For each mapping, programmers **reprove many properties**
    - Burrow proves many common properties generally
2. Proof assistant must be **explicitly told** that properties transfer along mapping
    - Burrow primitives translate common properties

Implemented in Agda

1. Common properties

- Properties ensure *well-formedness*

- Example properties:
  - every read reads-from *exactly* one write
  - the reads-from observes the written *value*
- 25 more properties

1. Proving common properties generally

Well-formedness follows from simple conditions

- Reads map to reads
- Writes map to writes

} while preserving locations and values

(and a few more)

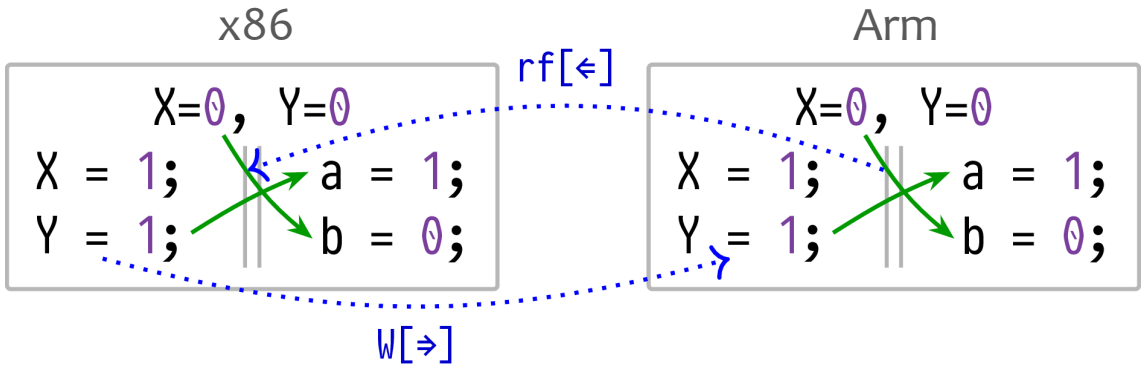
- These hold for our example mapping:

x86		Arm
`rmov`	→	`ldapr`
`wmov`	→	`stlr`

Burrow proves well-formedness generally

2. Primitives that translate common properties

- Proof by contradiction often translates properties



Burrow primitives translate common properties

Case Study

- Mapping from x86-to-Arm

x86		Arm
`rmov`	→	`ldapr`
`wmov`	→	`stlr`

- Our prior work on binary translation
  - Lasagne [PLDI'22]
  - Risotto [ASPLOS'23]
  - Arancini [ASPLOS'26]

Burrow: A Proof Framework for Weak Memory

Problem

Mechanizing weak memory mapping proofs is *challenging* and *time-consuming*

repository

Solution

Burrow brings **abstractions** and **primitives** to easily write such proofs in Agda

Contact:
dennis.sprokholt@tum.de