Executable Multicore Model of RISC-V in HolBA

Henrik Karlsson
KTH Royal Institute of Technology, Sweden

Outline

- Problems of executable multicore models
- How we implement multicore in HolBA
- Summary

Problems of executable multicore models

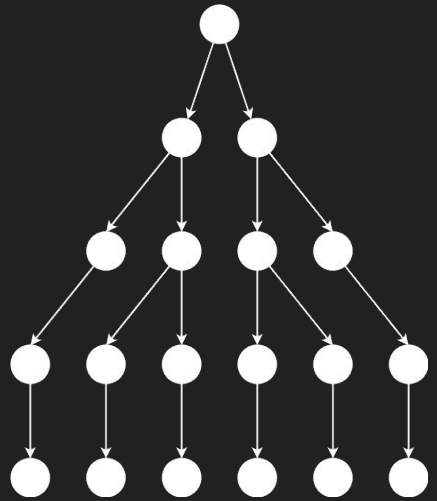
- Interleaving
- Weak-Memory Model
- Not compositional

```
Thread 1     Thread 2
---          ---
r1 := x      r1 := y
y  := 1      x  := r1
```

Problems of executable multicore models

- Interleaving
- Weak-Memory Model
- Not compositional

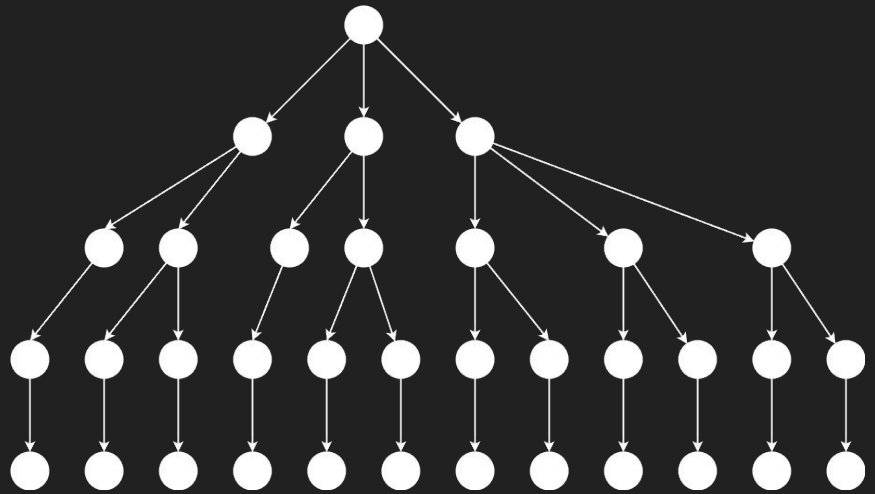
```
Thread 1      Thread 2
---           ---
r1 := x       r1 := y
y  := 1       x  := r1
```



Problems of executable multicore models

- Interleaving
- Weak-Memory Model
- Not compositional

```
Thread 1      Thread 2
---           ---
r1 := x       r1 := y
y  := 1       x  := r1
```



Implementing multicore in HolBA

- Extend HolBA with Promising RISC-V*
  - Stores: Promise (Write) + Fulfil
  - Memory: History of writes
  - Loads: Non-deterministic read of writes
- Relational HolBA multicore model

*C. Pulte, J. Pichon-Pharabod, J. Kang, S.-H. Lee, and C.-K. Hur, “Promising-ARM/RISC-V: a simpler and faster operational concurrency model,” *Apollo (University of Cambridge)*, Jun. 2019, doi: https://doi.org/10.1145/3314221.3314624.

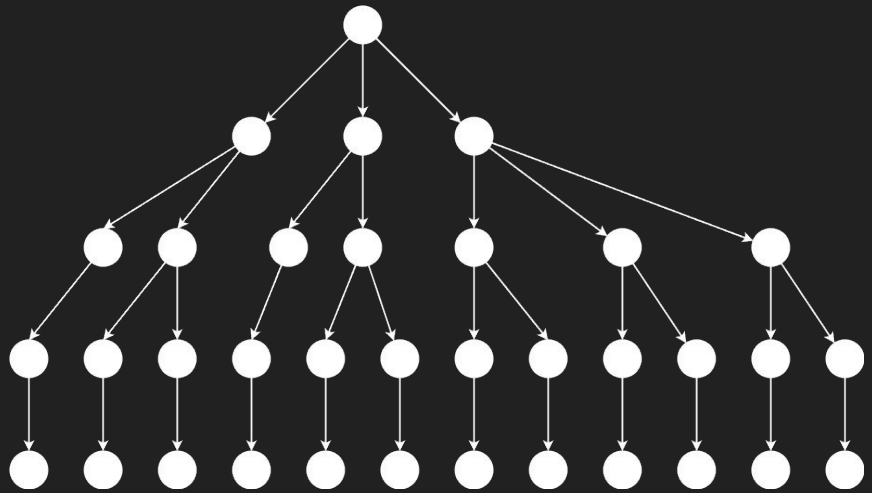
Implementing multicore in HolBA

- Executable Model
  - Bounded executions of RISC-V binary
  - Execute stores first (promises)
  - Core-local + loads done sequentially
- Added atomic memory operations
- Validated using litmus tests*
- Ongoing: Bisimulation proof with relational model

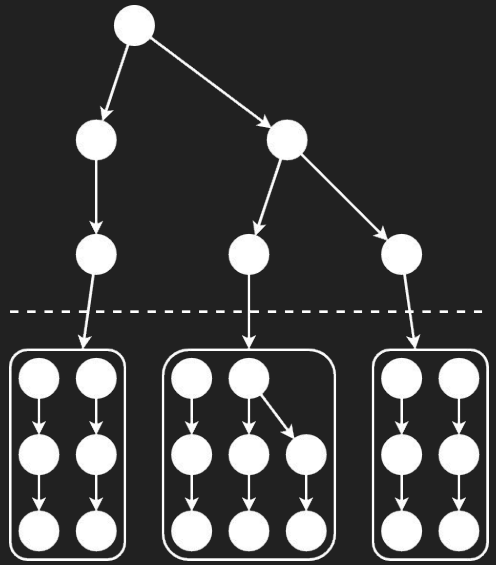
*J. Alglave, L. Maranget, and M. Tautschnig, “Herding Cats,” ACM Transactions on Programming Languages and Systems, vol. 36, no. 2, pp. 1–74, Jul. 2014, doi: https://doi.org/10.1145/2627752.

Implementing multicore in HolBA

Naive Execution



Promising Execution



Summary

- Interleaving of **ALL** operations is unnecessary
- Promising semantics => Global phase + local phase
- HolBA + Promising Semantics
  - Relational model - for proofs
  - Executable model - for tests and validation
- Added atomic memory operations
- Ongoing: Bisimulation proof