

# Tools for exploring and understanding memory models

John Wickerson, Imperial College

Joint work with

Jade Alglave (UCL), Mark Batty (Cambridge), Alastair Donaldson (Imperial),  
Ganesh Gopalakrishnan (Utah), Luc Maranget (INRIA),  
Daniel Poetzl (Oxford) and Tyler Sorensen (Utah)

# Outline

- Memory models
- Lem
- Herd

# Memory models

- Memory model:
  - which values might be read from memory
  - assembly code / higher-level language
  - operational / axiomatic

# Axiomatic memory model

- Defines predicates over program executions
- Contract between programmer and language

The value of an atomic object  $M$ , as determined by evaluation  $B$ , shall be the value stored by some operation in the visible sequence of  $M$  with respect to  $B$ .

C standard §5.1.2.4:22

The execution of a program contains a *data race* if it contains two conflicting actions in different threads, at least one of which is not atomic, and neither happens before the other. Any such data race results in undefined behavior.

C standard §5.1.2.4:25

# Lem

A *visible side effect*  $A$  on an object  $M$  with respect to a value computation  $B$  of  $M$  satisfies the conditions:

- $A$  happens before  $B$ , and
- there is no other side effect  $X$  to  $M$  such that  $A$  happens before  $X$  and  $X$  happens before  $B$ .

C standard §5.1.2.4:19

```
let visible_side_effect_set actions hb =  
  { (A,B) | forall ((A,B) IN hb) |  
    is_write A && is_read B && (loc_of A = loc_of B) &&  
    not ( exists (X IN actions). not (X IN {A;B}) &&  
      is_write X && (loc_of X = loc_of B) &&  
      (A,X) IN hb && (X,B) IN hb ) }
```

Lem

# Herd

A *visible side effect*  $A$  on an object  $M$  with respect to a value computation  $B$  of  $M$  satisfies the conditions:

- $A$  happens before  $B$ , and
- there is no other side effect  $X$  to  $M$  such that  $A$  happens before  $X$  and  $X$  happens before  $B$ .

C standard §5.1.2.4:19

```
let hbl = hb & loc  
let vse = [W];hbl;[R] & ~(hbl;[W];hbl)
```

**Herd**

# Herd

$$\frac{(a,b) \in R \quad (b,c) \in S}{(a,c) \in R ; S}$$

$$\frac{a \in s}{(a,a) \in [s]}$$

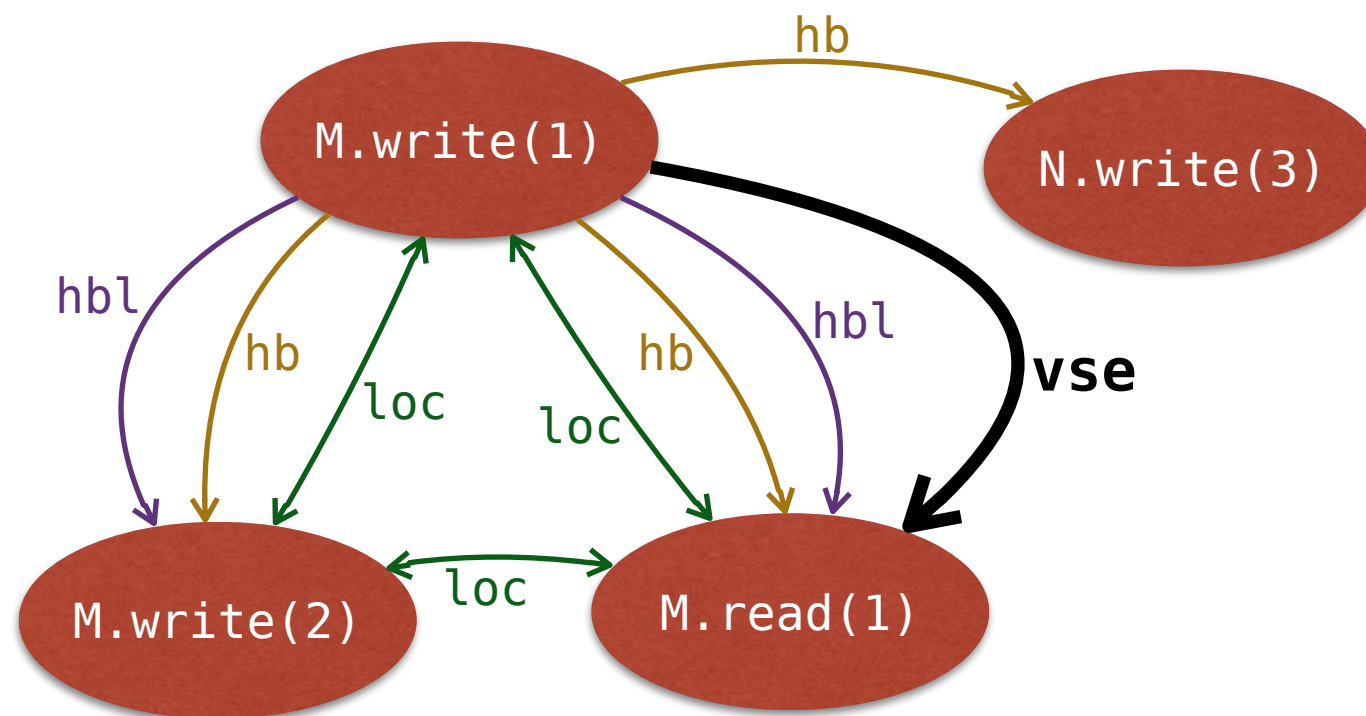
$R \ \& \ S$      $R \mid S$      $R ; S$      $[s]$

$\sim R$      $R^{-1}$      $R^*$      $R^+$      $R^?$

$[zone-4] ; northern ; [step-free] ; central ; [zone-6]$

Herd

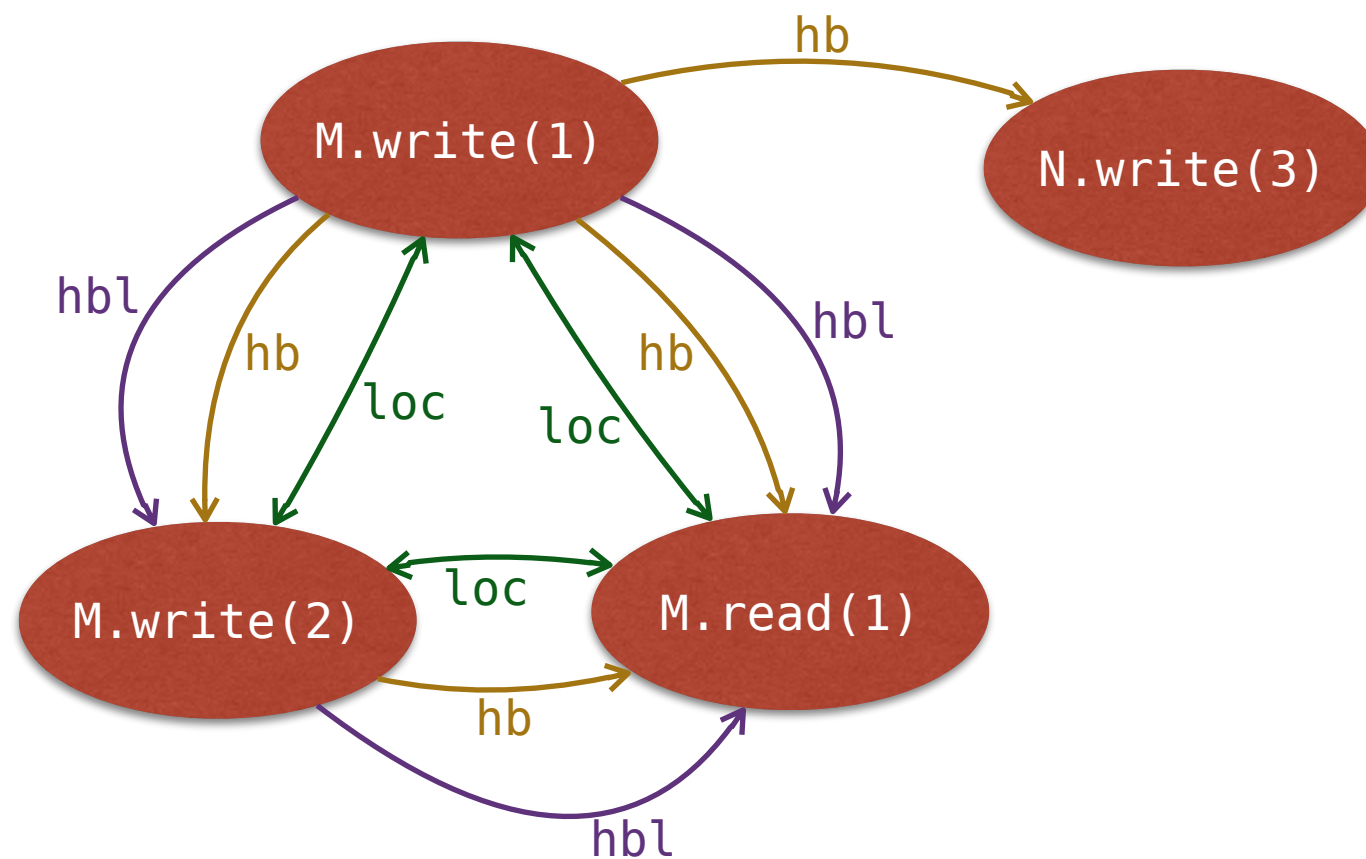
# Herd



```
let hbl = hb & loc  
let vse = [W];hbl;[R] & ~(hbl;[W];hbl)
```



# Herd



```
let hb1 = hb & loc  
let vse = [W];hb1;[R] & ~(hb1;[W];hb1)
```

# The POWER of Herd

# Lem vs. Herd

- Lem:
  - Expressive language → match prose
- Herd:
  - Concise → optimise model, compare models
  - Efficient simulation

# Our work

- C/C++ memory model in Herd, prove equivalent to existing Lem version
- OpenCL memory model in Lem and Herd
- NVIDIA's PTX memory model

# Tools for exploring and understanding memory models

John Wickerson, Imperial College

Joint work with

Jade Alglave (UCL), Mark Batty (Cambridge), Alastair Donaldson (Imperial),  
Ganesh Gopalakrishnan (Utah), Luc Maranget (INRIA),  
Daniel Poetzl (Oxford) and Tyler Sorensen (Utah)