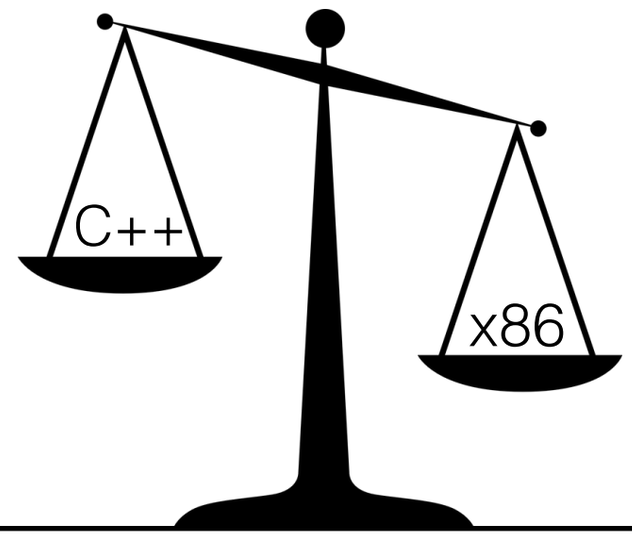


Automatically Comparing Memory Consistency Models

John Wickerson Imperial
Mark Batty U Kent
Tyler Sorensen Imperial
George A. Constantinides Imperial

S-REPLS @ Imperial
Tuesday 27 September 2016

Contents

- Context: memory consistency models (MCMs)
- Where our work fits in
- Key Ideas
- Applications

Contents

- Context: memory consistency models (MCMs)
- Where our work fits in
- Key Ideas
- Applications

Relaxed-memory behaviours

```
          x = y = 0;
x = 1;    ||  y = 1;
r0 = y;   ||  r1 = x;
```

Relaxed-memory behaviours

```
          x = y = 0;
x = 1;    ||
r0 = y;   ||
          ||  y = 1;
          ||  r1 = x;
```

r0 = 0, r1 = 1

Relaxed-memory behaviours

```
                x = y = 0;
x = 1;      ||
            ||  y = 1;
r0 = y;     ||
            ||  r1 = x;
```

r0 = 0, r1 = 1

r0 = 1, r1 = 1

Relaxed-memory behaviours

```
x = y = 0;
                ||  y = 1;
                ||  r1 = x;
x = 1;          ||
r0 = y;         ||
```

r0 = 0, r1 = 1

r0 = 1, r1 = 1

r0 = 1, r1 = 0

Relaxed-memory behaviours

```
        x = y = 0;
x = 1;     ||  y = 1;
r0 = y;    ||  r1 = x;
```

```
r0 = 0, r1 = 0
r0 = 0, r1 = 1
r0 = 1, r1 = 1
r0 = 1, r1 = 0
```

Much confusion!

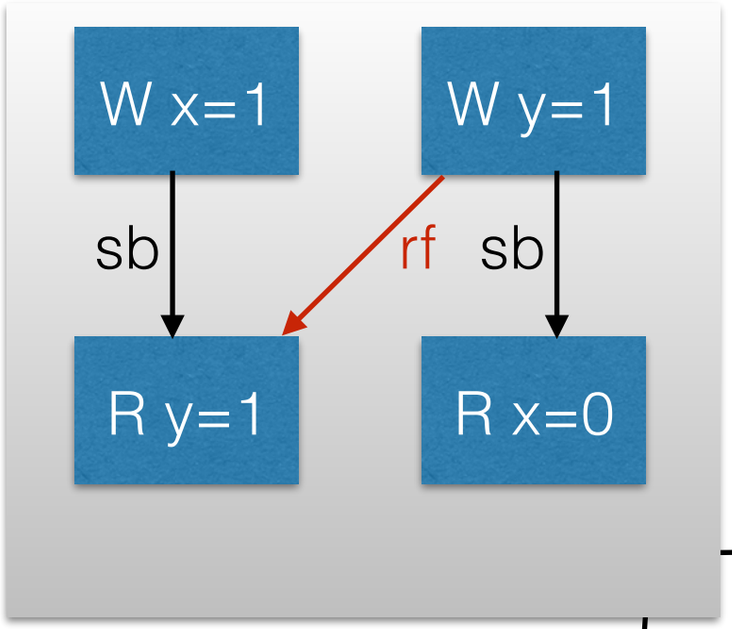
Subtleties related to relaxed memory have led to bugs in...

- **programming language specifications** [Batty+ POPL'11, Batty+ ESOP'13],
- **deployed processors** [Alglave+ CAV'10];
- **compilers** [Morisset+ PLDI'13, Sevcik+ ECOOP'08], and
- **vendor-endorsed programming guides** [Alglave+ ASPLOS'15].

Axiomatic models

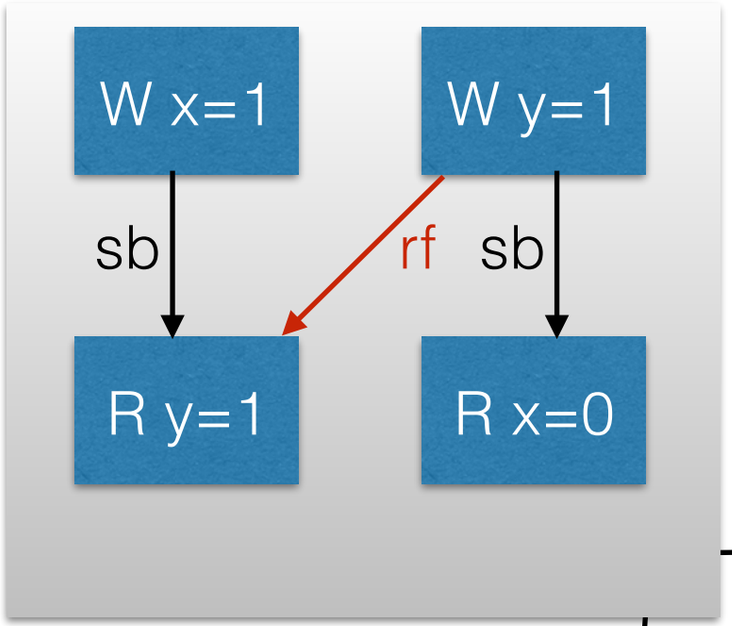
```
x = 1;   ||  y = 1;
r0 = y;  ||  r1 = x;
```

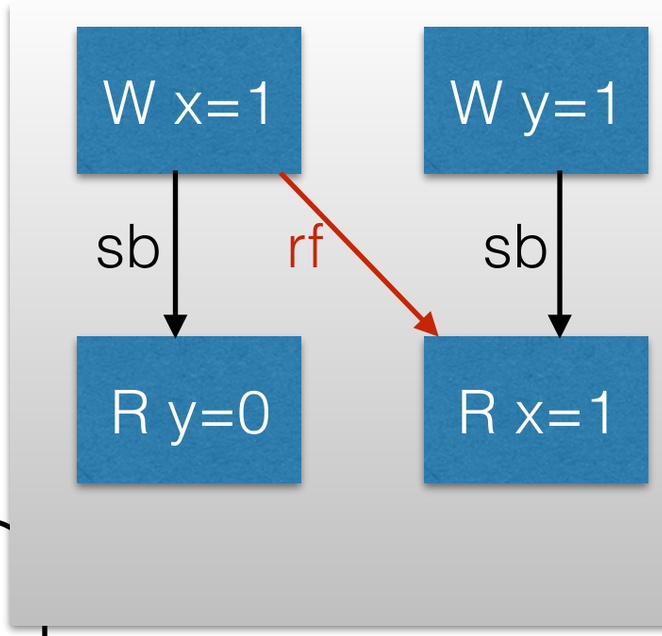
Axiomatic models




```
x = 1;  ||  y = 1;
r0 = y; ||  r1 = x;
```

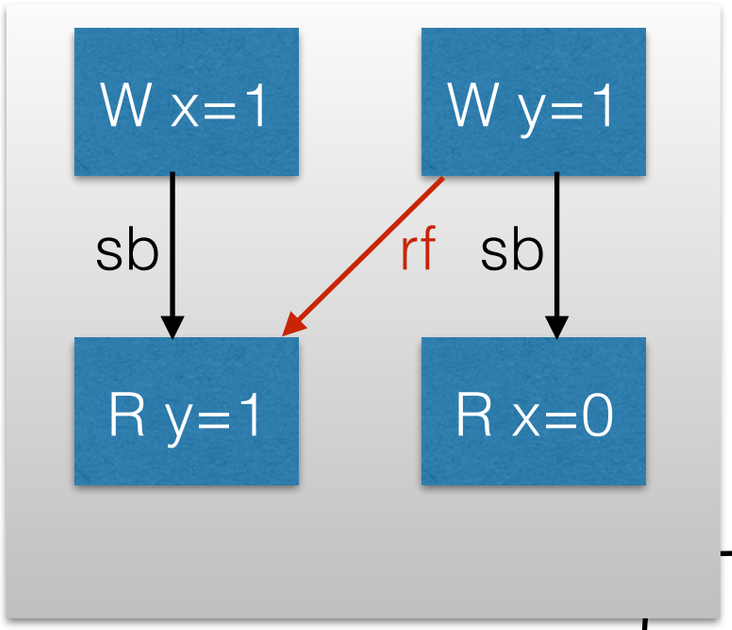
Axiomatic models

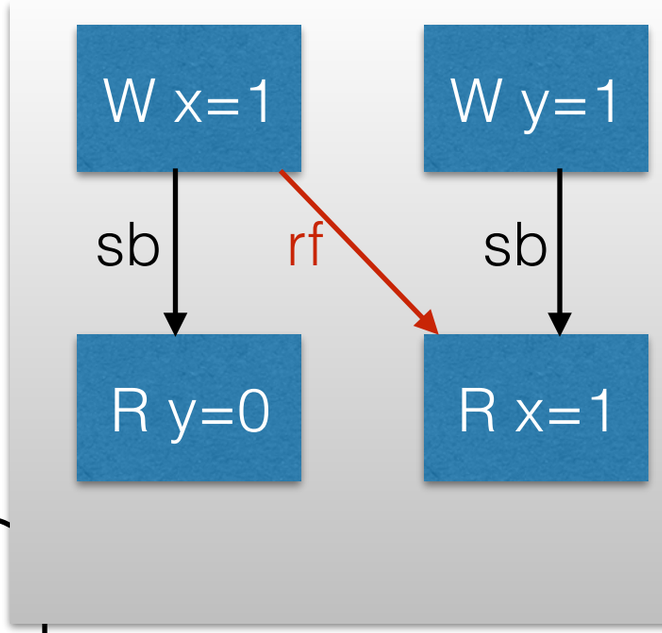




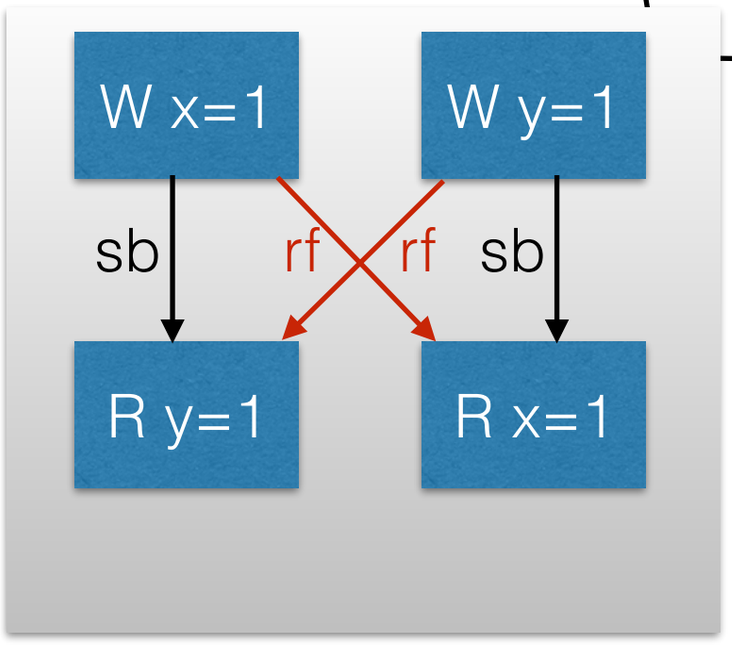

```
x = 1;  ||  y = 1;
r0 = y; ||  r1 = x;
```

Axiomatic models

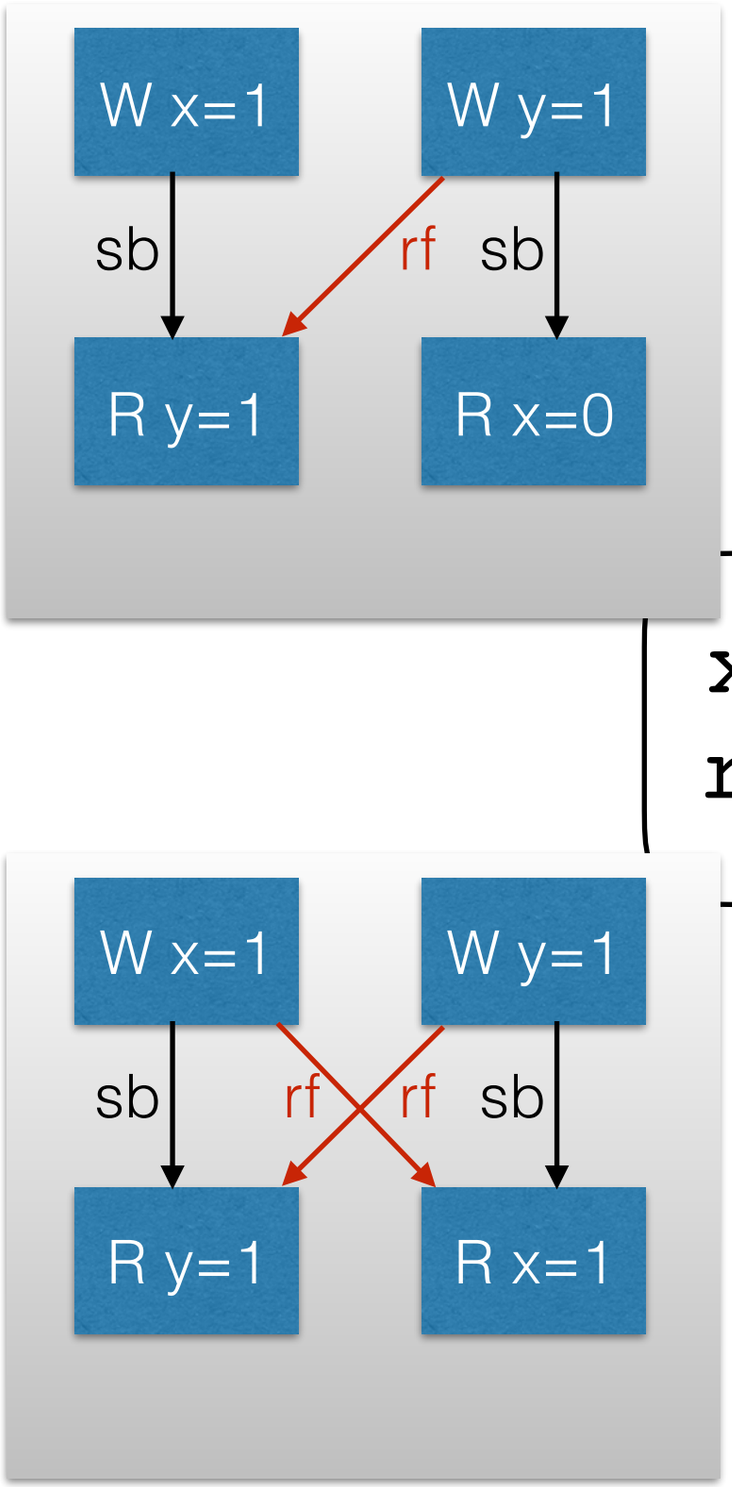


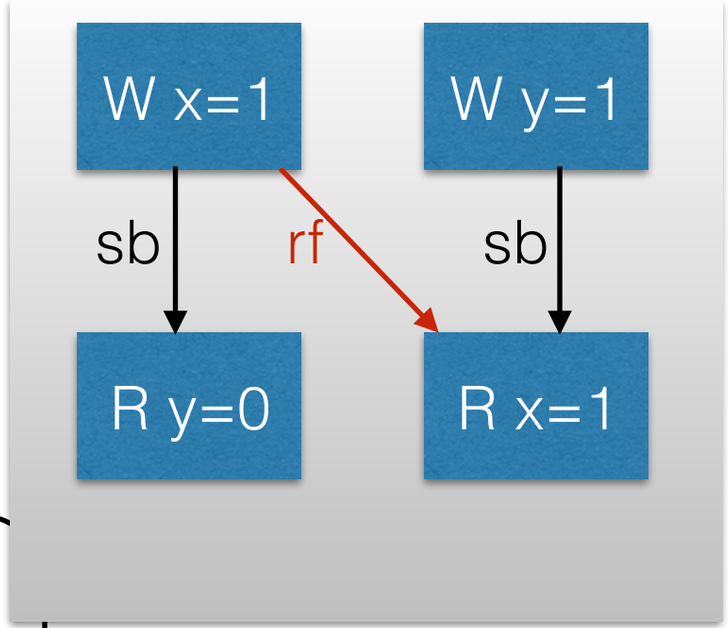



```
x = 1;  ||  y = 1;
r0 = y; ||  r1 = x;
```

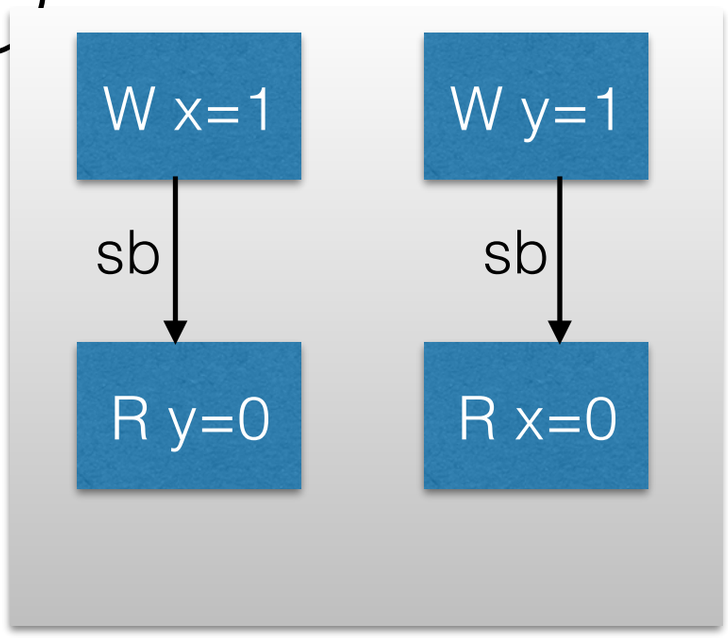


Axiomatic models

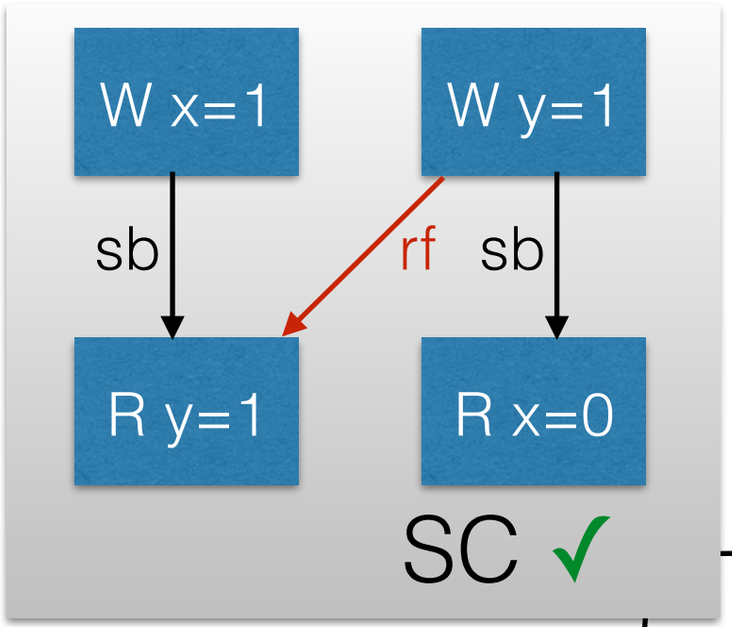


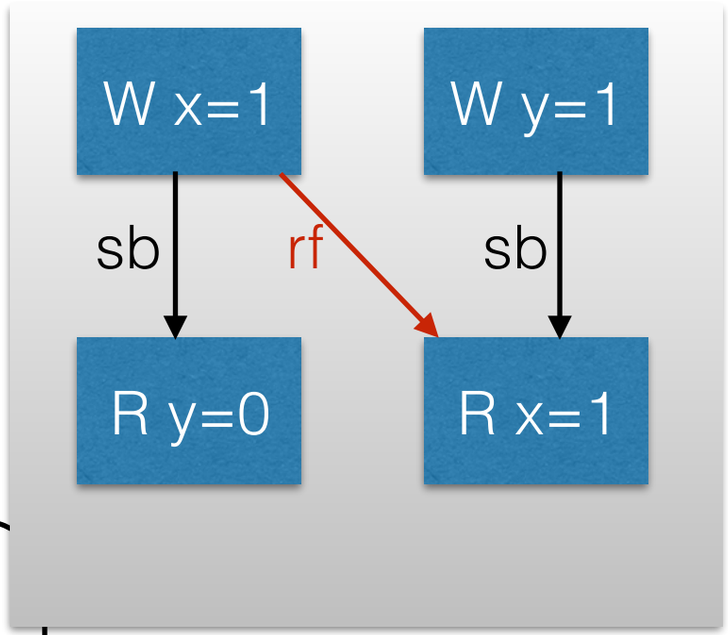



```
x = 1;  ||  y = 1;
r0 = y;     r1 = x;
```

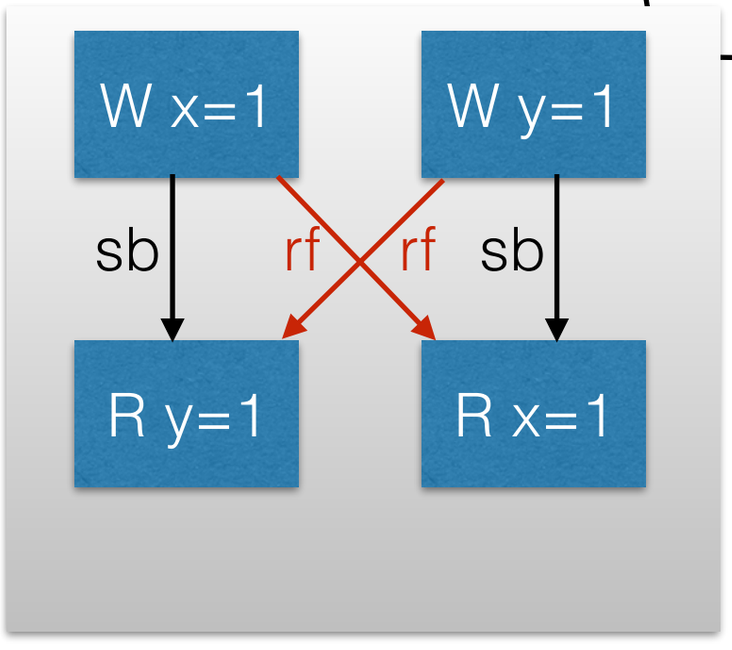



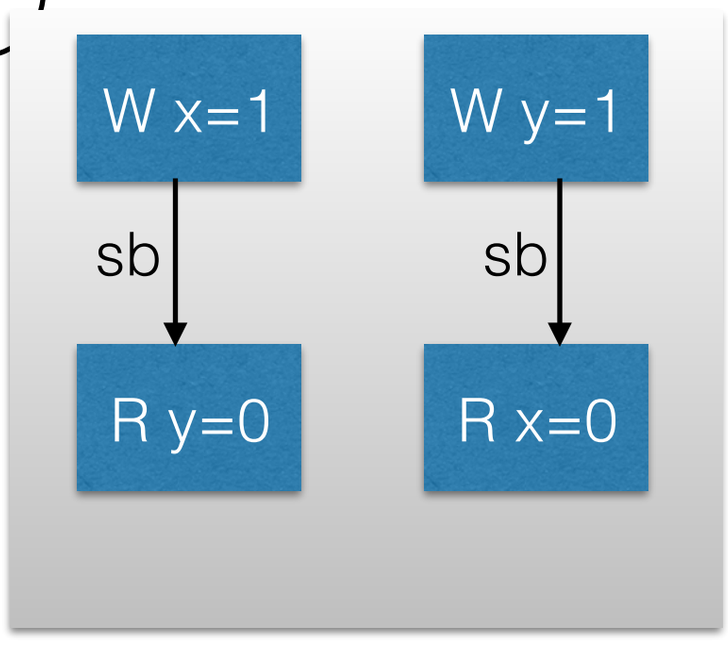
Axiomatic models



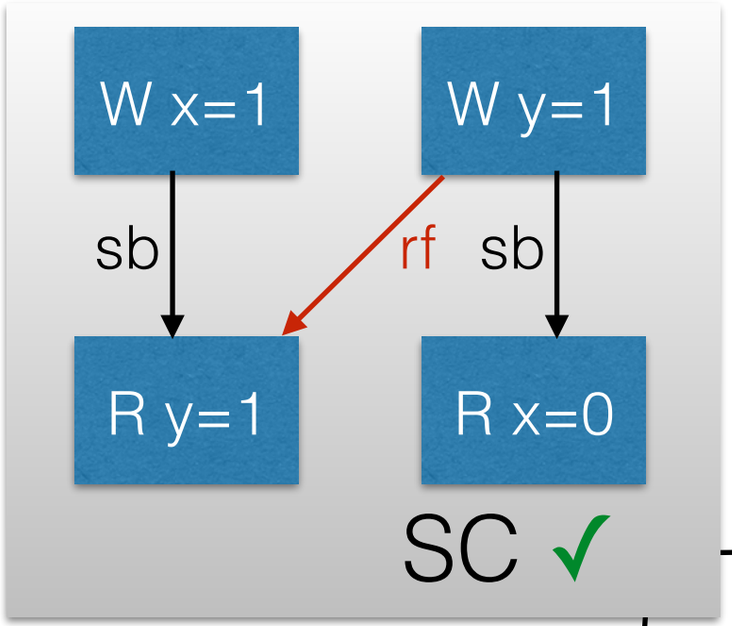


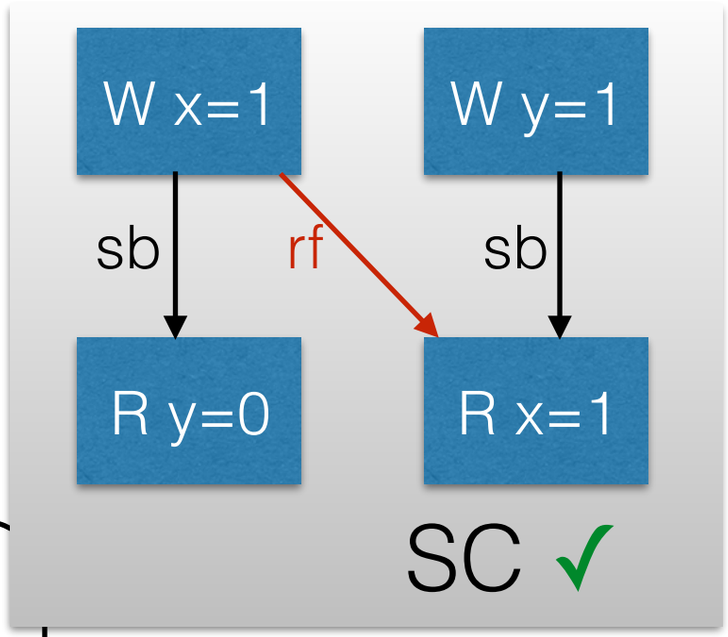

```
x = 1;  ||  y = 1;
r0 = y;  ||  r1 = x;
```



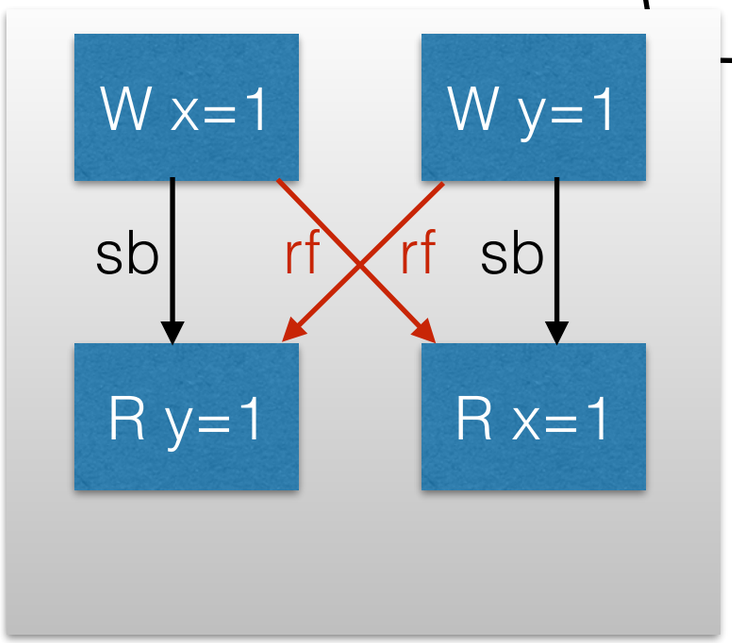


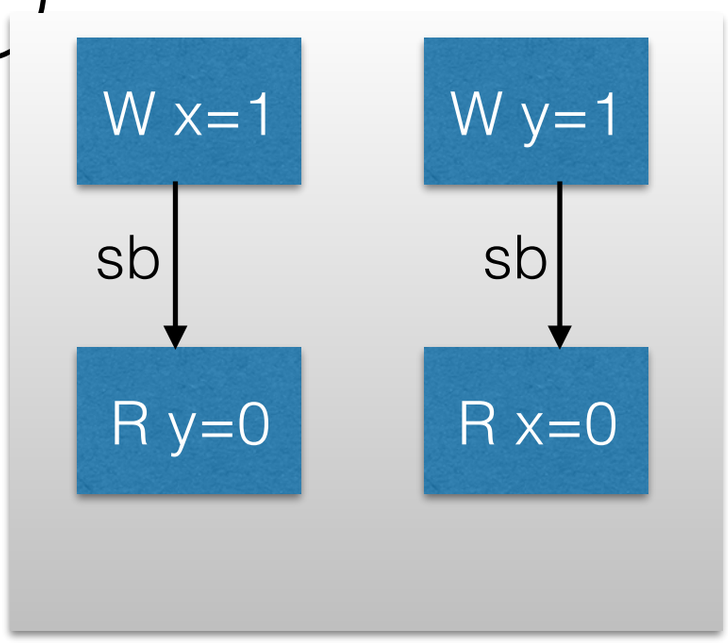
Axiomatic models



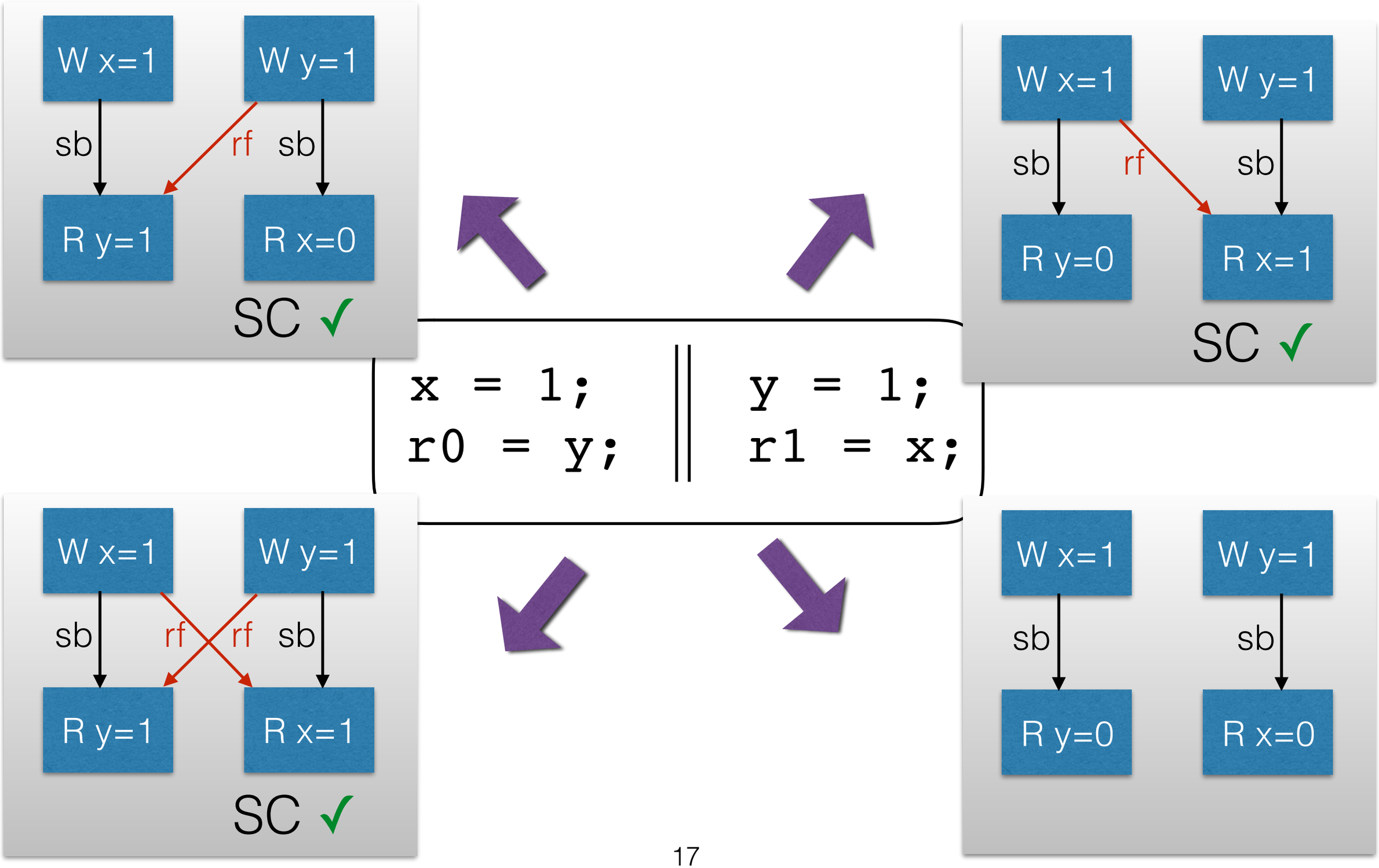



```
x = 1;  ||  y = 1;
r0 = y; ||  r1 = x;
```

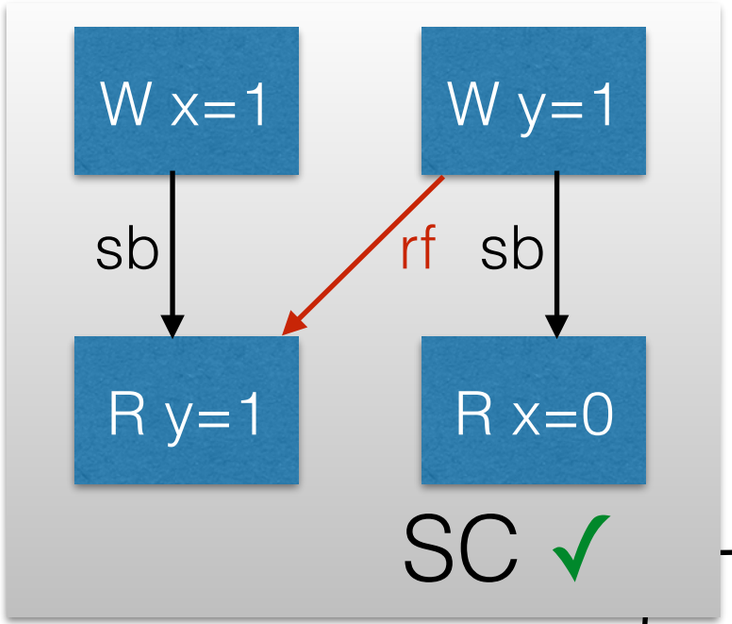


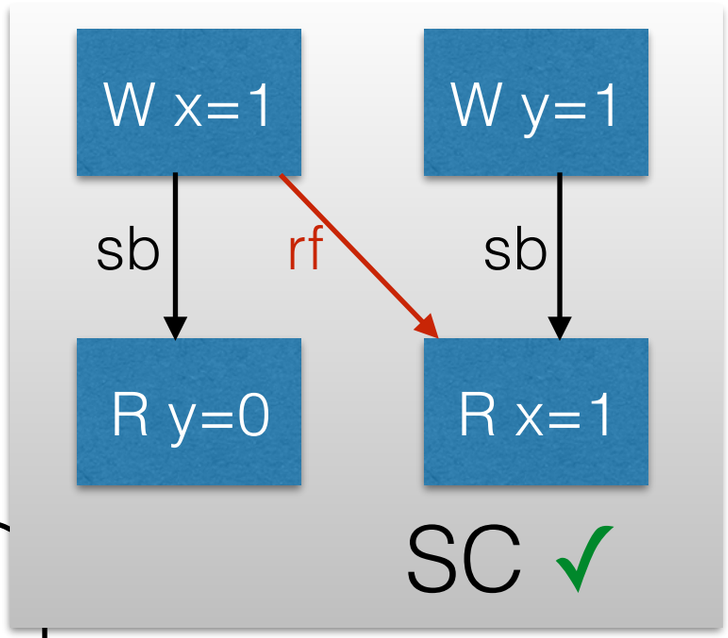


Axiomatic models

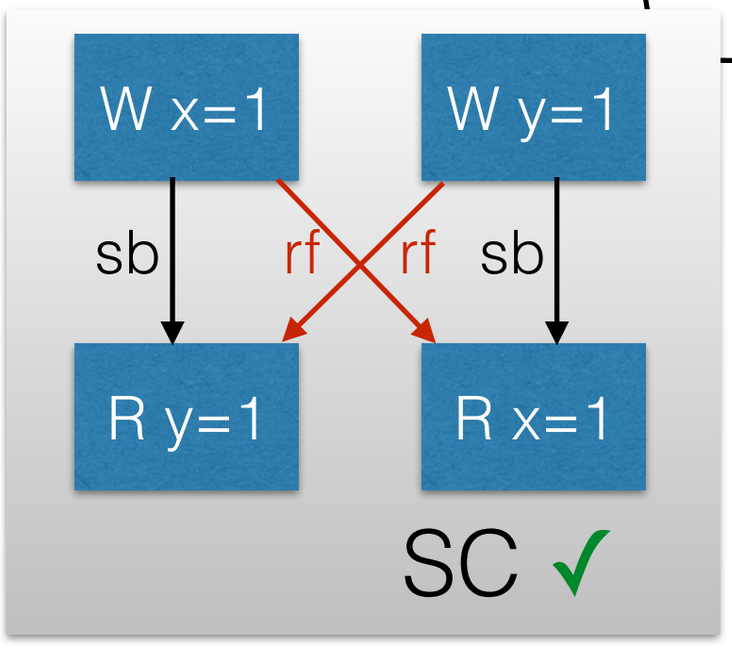


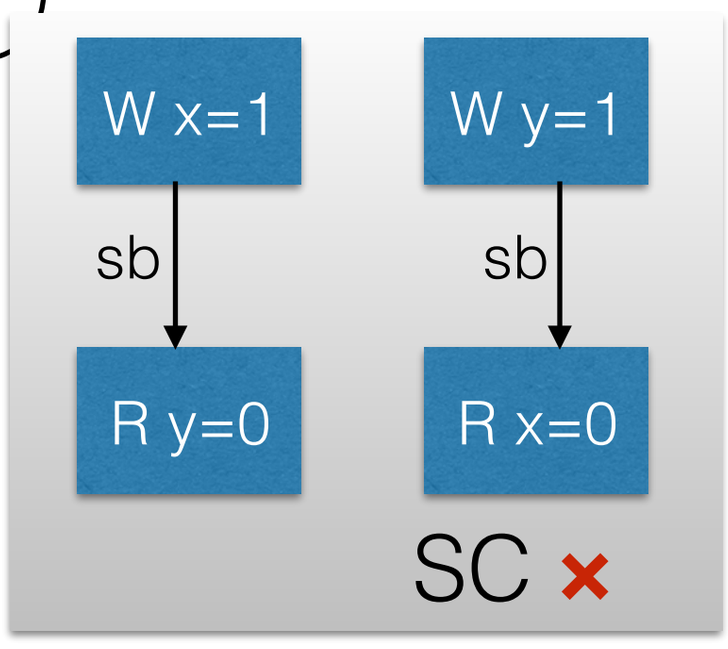
Axiomatic models



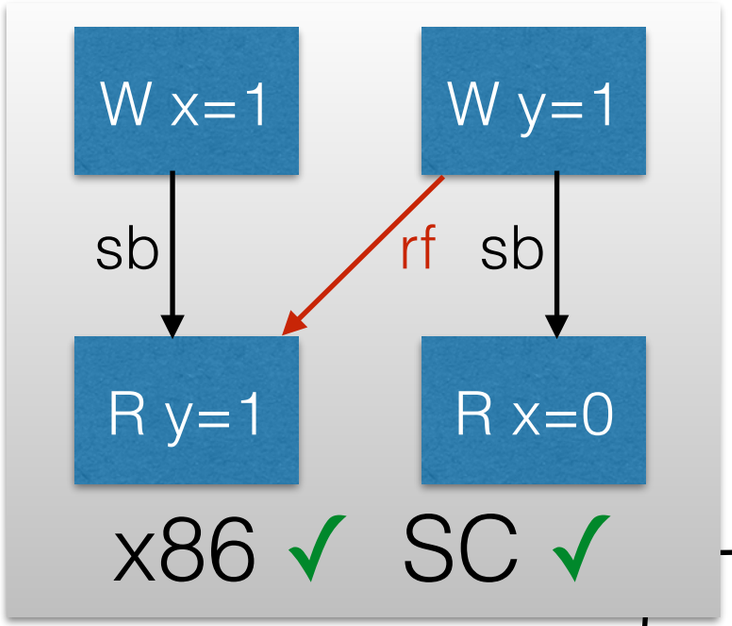


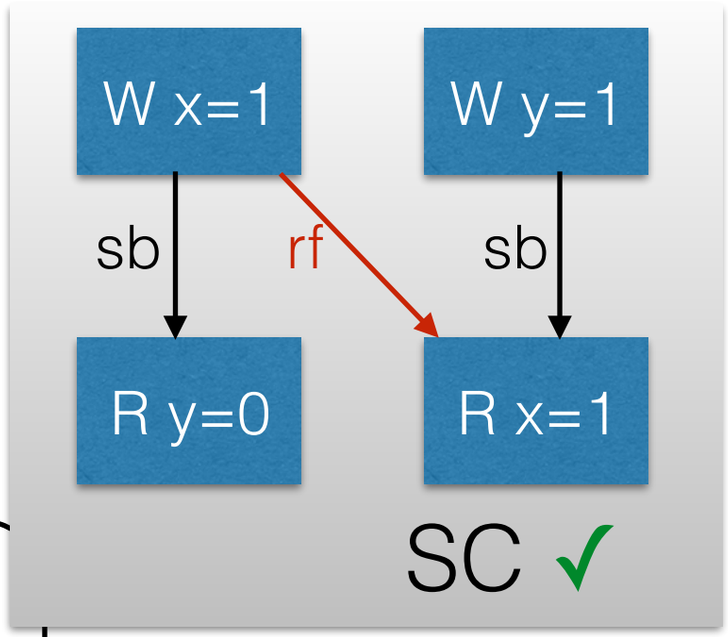

```
x = 1;  ||  y = 1;
r0 = y;  ||  r1 = x;
```



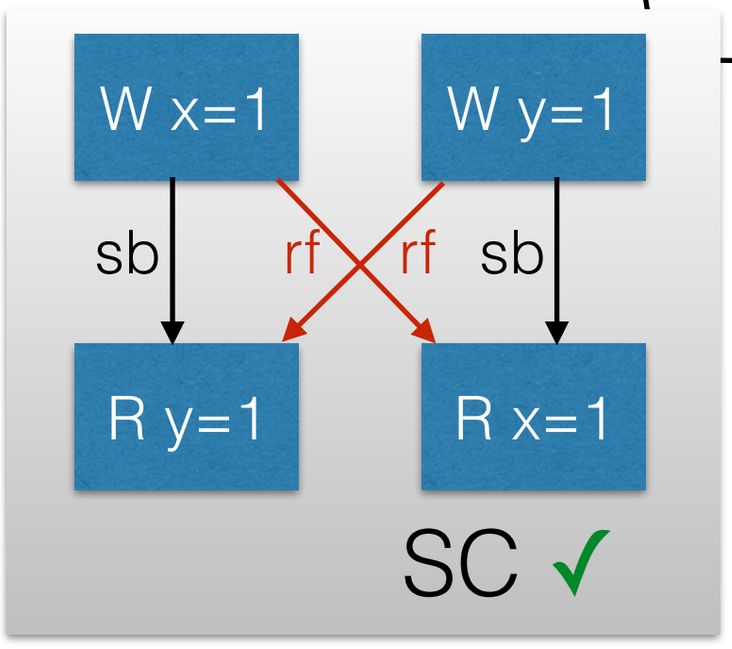


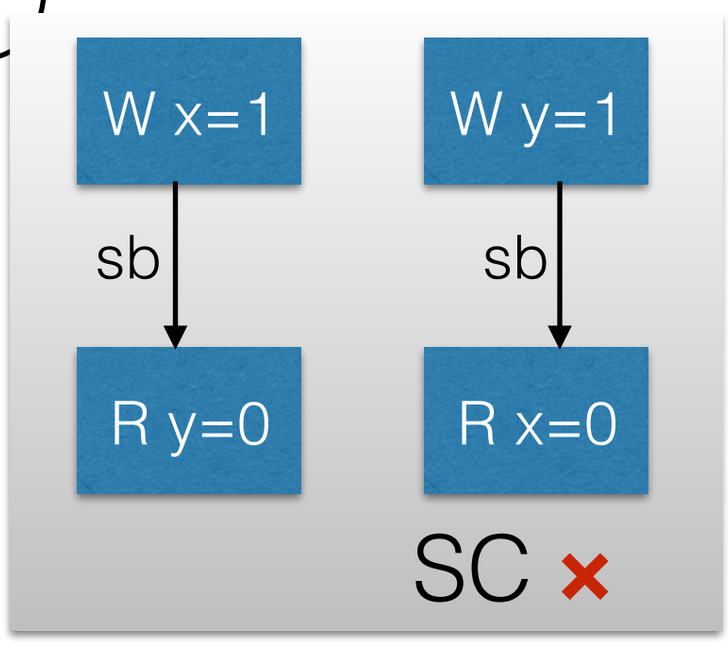
Axiomatic models



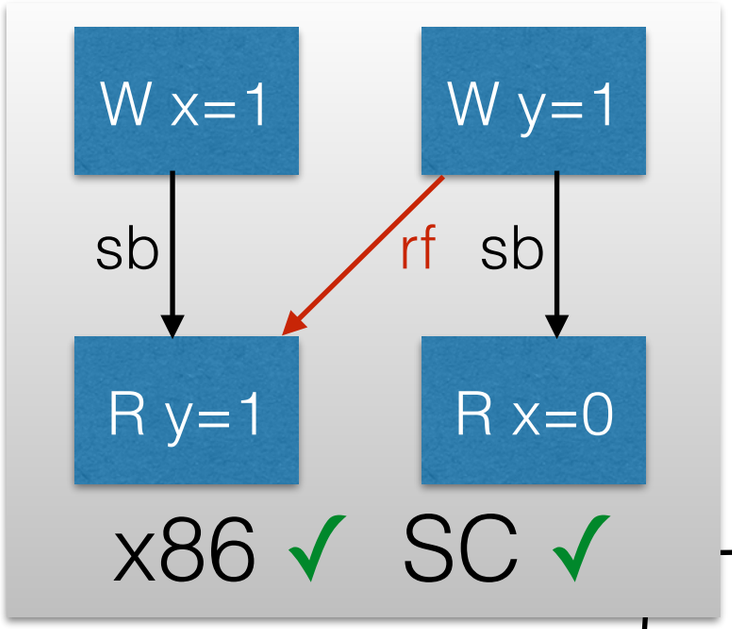


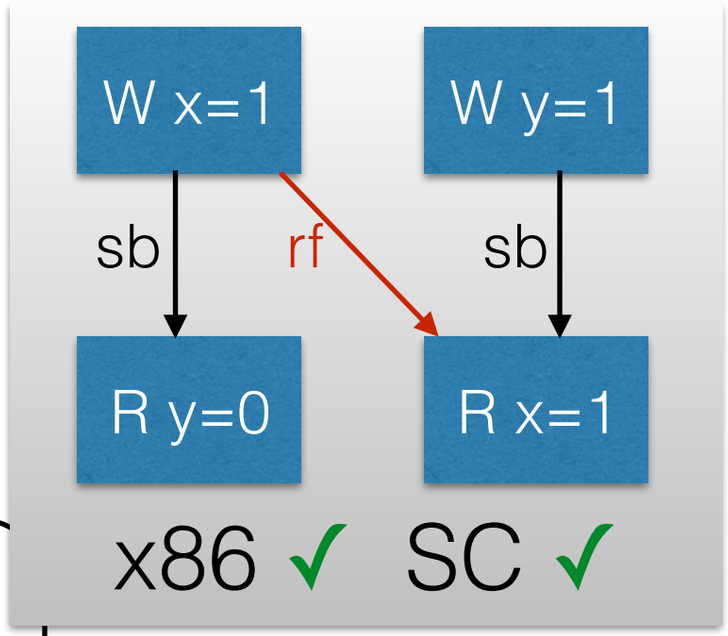

```
x = 1;  ||  y = 1;
r0 = y;     r1 = x;
```



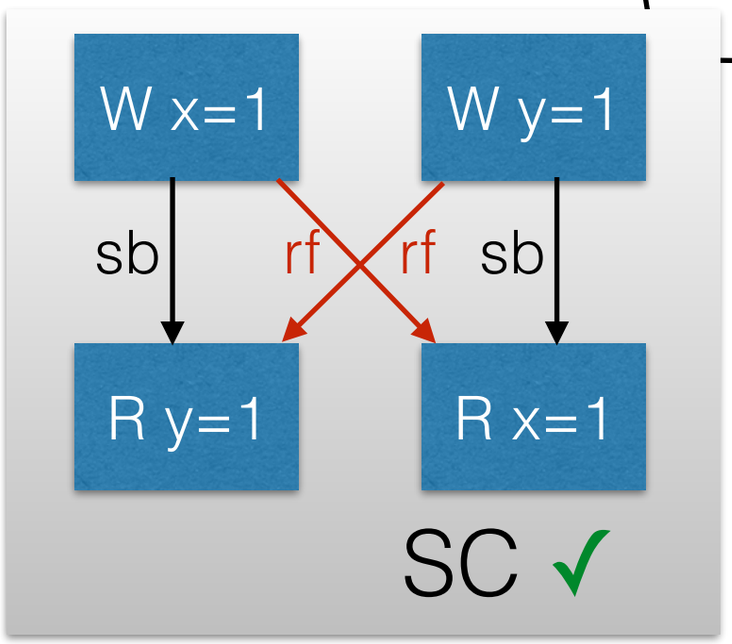


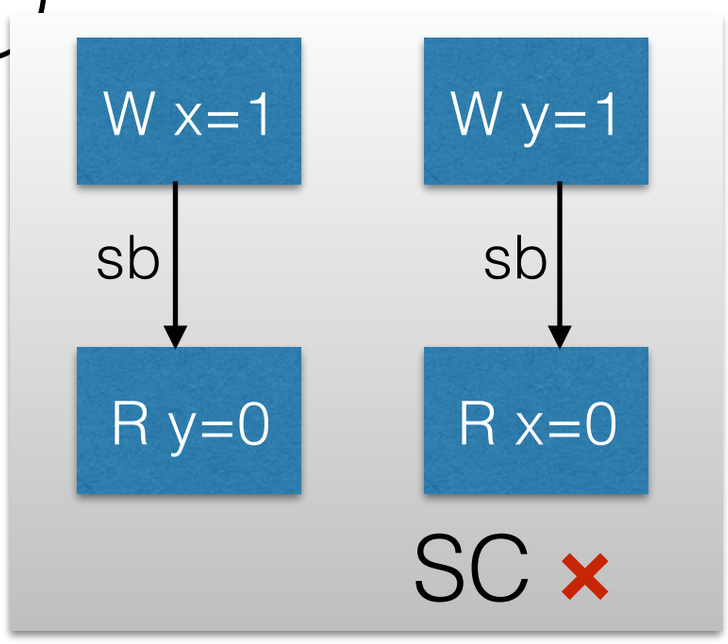
Axiomatic models



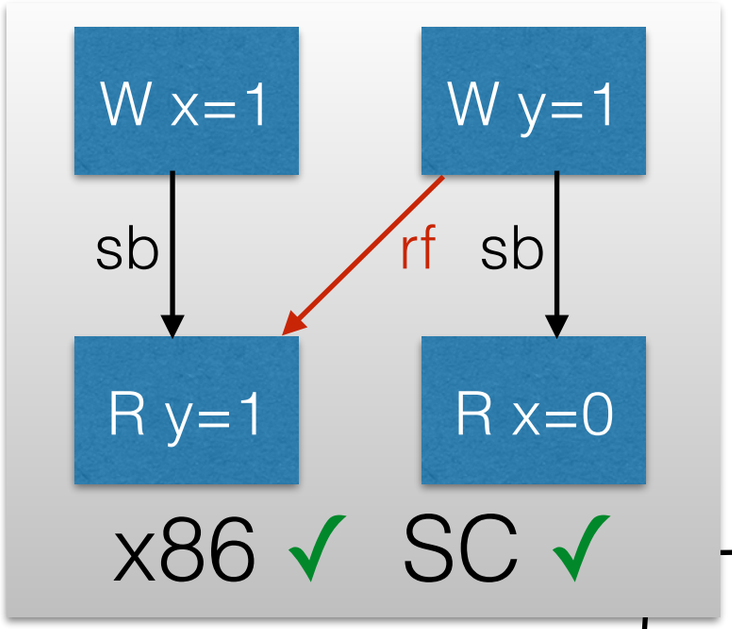


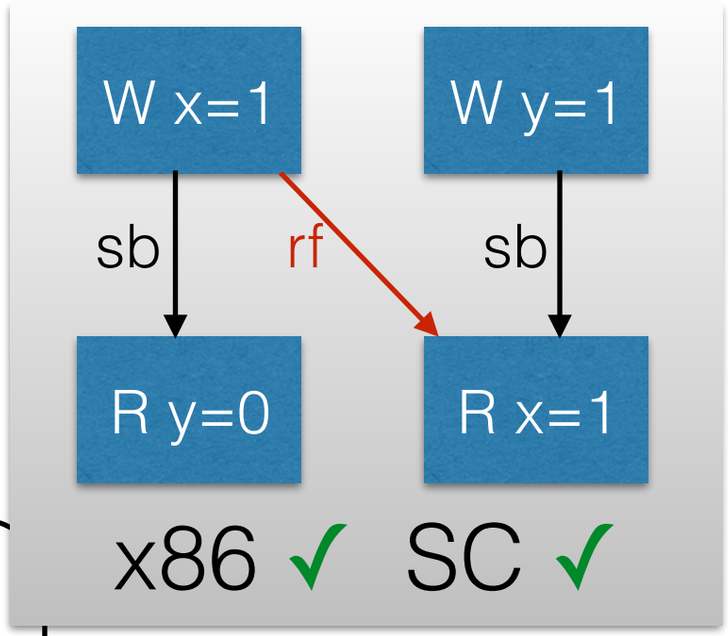

```
x = 1;  ||  y = 1;
r0 = y;     r1 = x;
```



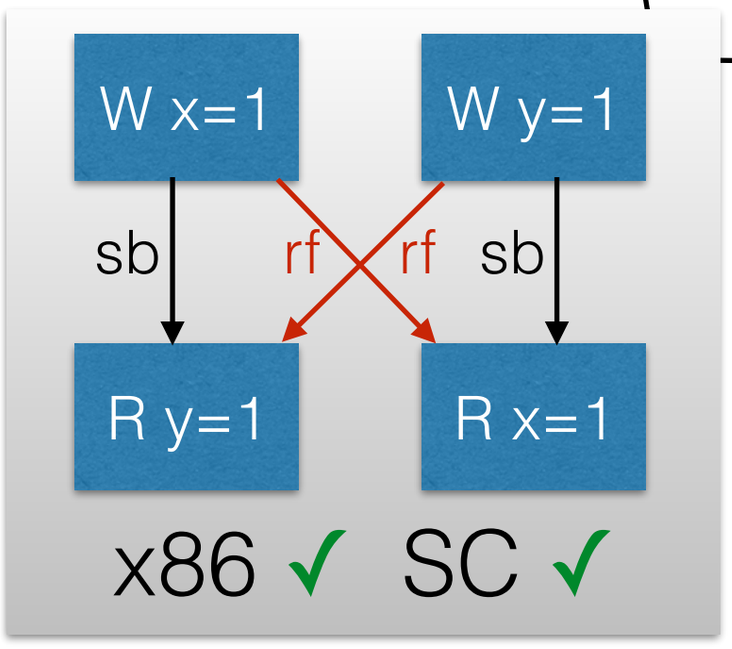


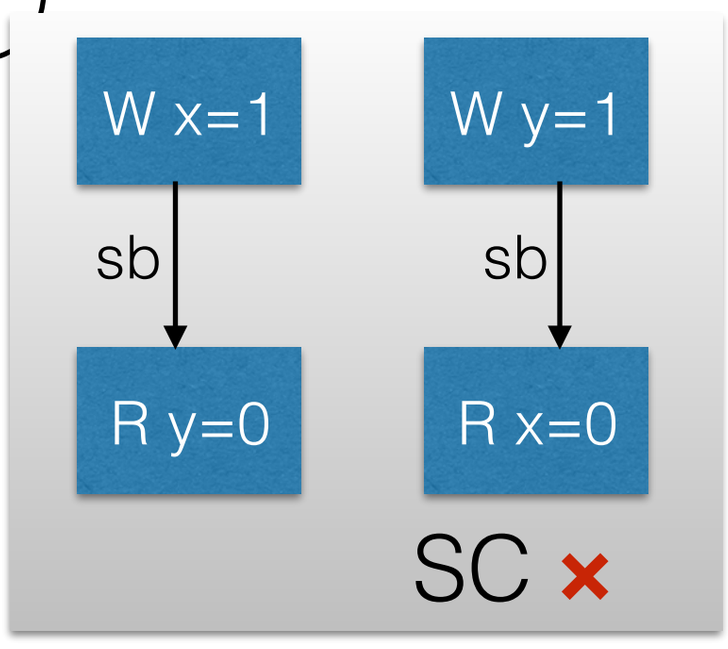
Axiomatic models



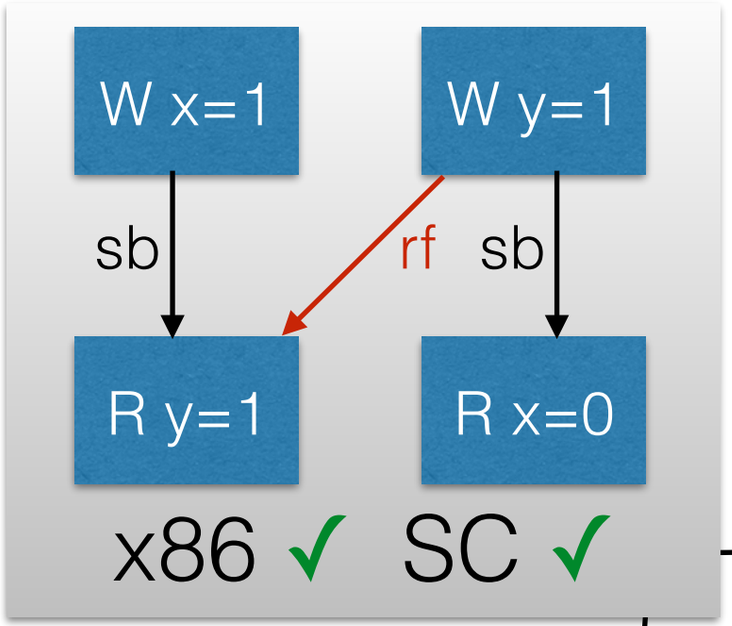


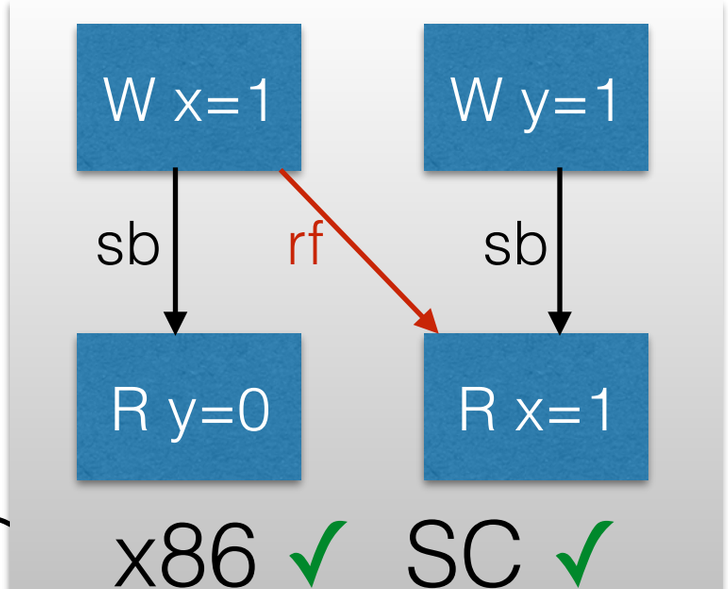

```
x = 1;  ||  y = 1;
r0 = y;     r1 = x;
```



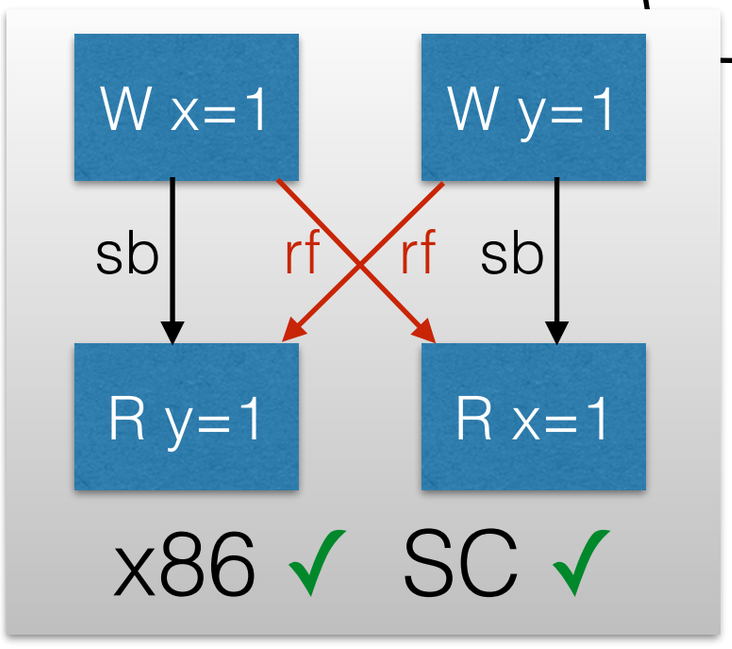


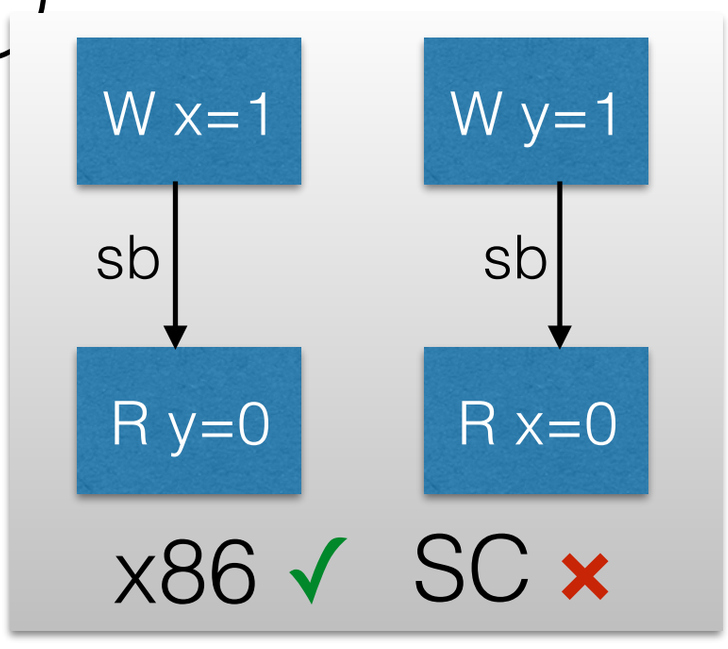
Axiomatic models






```
x = 1;  ||  y = 1;
r0 = y; ||  r1 = x;
```





Contents

- Context: memory consistency models (MCMs)
- Where our work fits in
- Key Ideas
- Applications

Some challenges and current approaches to tackling them

Question	Existing work

Some challenges and current approaches to tackling them

Question	Existing work
Can a given litmus test pass under a given MCM?	

Some challenges and current approaches to tackling them

Question	Existing work
Can a given litmus test pass under a given MCM?	CppMem, Herd, MemSAT, Nemos, ...

Some challenges and current approaches to tackling them

Question	Existing work
Can a given litmus test pass under a given MCM?	CppMem, Herd, MemSAT, Nemos, ...
Which litmus tests can be run to check whether a machine conforms to a given MCM?	

Some challenges and current approaches to tackling them

Question	Existing work
Can a given litmus test pass under a given MCM?	CppMem, Herd, MemSAT, Nemos, ...
Which litmus tests can be run to check whether a machine conforms to a given MCM?	semi-automatic generation with DIY [Alglave+ CAV'10];

Some challenges and current approaches to tackling them

Question	Existing work
Can a given litmus test pass under a given MCM?	CppMem, Herd, MemSAT, Nemos, ...
Which litmus tests can be run to check whether a machine conforms to a given MCM?	semi-automatic generation with DIY [Alglave+ CAV'10];
Is one MCM more permissive than another?	

Some challenges and current approaches to tackling them

Question	Existing work
Can a given litmus test pass under a given MCM?	CppMem, Herd, MemSAT, Nemos, ...
Which litmus tests can be run to check whether a machine conforms to a given MCM?	semi-automatic generation with DIY [Alglave+ CAV'10];
Is one MCM more permissive than another?	manual proof; manual examples [Batty+ POPL'16]; semi-automatic checking with DIY+Herd;

Some challenges and current approaches to tackling them

Question	Existing work
Can a given litmus test pass under a given MCM?	CppMem, Herd, MemSAT, Nemos, ...
Which litmus tests can be run to check whether a machine conforms to a given MCM?	semi-automatic generation with DIY [Alglave+ CAV'10];
Is one MCM more permissive than another?	manual proof; manual examples [Batty+ POPL'16]; semi-automatic checking with DIY+Herd;
Does my MCM allow a given compiler optimisation?	

Some challenges and current approaches to tackling them

Question	Existing work
Can a given litmus test pass under a given MCM?	CppMem, Herd, MemSAT, Nemos, ...
Which litmus tests can be run to check whether a machine conforms to a given MCM?	semi-automatic generation with DIY [Alglave+ CAV'10];
Is one MCM more permissive than another?	manual proof; manual examples [Batty+ POPL'16]; semi-automatic checking with DIY+Herd;
Does my MCM allow a given compiler optimisation?	manual c'examples [Vafeiaidis+ POPL'15]; manual proof [Sevcik PLDI'11];

Some challenges and current approaches to tackling them

Question	Existing work
Can a given litmus test pass under a given MCM?	CppMem, Herd, MemSAT, Nemos, ...
Which litmus tests can be run to check whether a machine conforms to a given MCM?	semi-automatic generation with DIY [Alglave+ CAV'10];
Is one MCM more permissive than another?	manual proof; manual examples [Batty+ POPL'16]; semi-automatic checking with DIY+Herd;
Does my MCM allow a given compiler optimisation?	manual c'examples [Vafeiaidis+ POPL'15]; manual proof [Sevcik PLDI'11];
Does my MCM allow a given compiler mapping?	

Some challenges and current approaches to tackling them

Question	Existing work
Can a given litmus test pass under a given MCM?	CppMem, Herd, MemSAT, Nemos, ...
Which litmus tests can be run to check whether a machine conforms to a given MCM?	semi-automatic generation with DIY [Alglave+ CAV'10];
Is one MCM more permissive than another?	manual proof; manual examples [Batty+ POPL'16]; semi-automatic checking with DIY+Herd;
Does my MCM allow a given compiler optimisation?	manual c'examples [Vafeiaidis+ POPL'15]; manual proof [Sevcik PLDI'11];
Does my MCM allow a given compiler mapping?	manual c'examples [Wickerson+ OOPSLA'15]; manual proof [Batty+ POPL'11, Batty+ POPL'12];

Our contributions

Question	Existing work
Can a given litmus test pass under a given MCM?	CppMem, Herd, MemSAT, Nemos, ...
Which litmus tests can be run to check whether a machine conforms to a given MCM?	semi-automatic generation with DIY [Alglave+ CAV'10];
Is one MCM more permissive than another?	manual proof; manual examples [Batty+ POPL'16]; semi-automatic checking with DIY+Herd;
Does my MCM allow a given compiler optimisation?	manual c'examples [Vafeiaidis+ POPL'15]; manual proof [Sevcik PLDI'11];
Does my MCM allow a given compiler mapping?	manual c'examples [Wickerson+ OOPSLA'15]; manual proof [Batty+ POPL'11, Batty+ POPL'12];

Our contributions

Question	Existing work
Can a given litmus test pass under a given MCM?	CppMem, Herd, MemSAT, Nemos, ...
Which litmus tests can be run to check whether a machine conforms to a given MCM?	semi-automatic generation with DIY [Alglave+ CAV'10]; **automatic generation**
Is one MCM more permissive than another?	manual proof; manual examples [Batty+ POPL'16]; semi-automatic checking with DIY+Herd;
Does my MCM allow a given compiler optimisation?	manual c'examples [Vafeiaidis+ POPL'15]; manual proof [Sevcik PLDI'11];
Does my MCM allow a given compiler mapping?	manual c'examples [Wickerson+ OOPSLA'15]; manual proof [Batty+ POPL'11, Batty+ POPL'12];

Our contributions

Question	Existing work
Can a given litmus test pass under a given MCM?	CppMem, Herd, MemSAT, Nemos, ...
Which litmus tests can be run to check whether a machine conforms to a given MCM?	semi-automatic generation with DIY [Alglave+ CAV'10]; **automatic generation**
Is one MCM more permissive than another?	manual proof; manual examples [Batty+ POPL'16]; semi-automatic checking with DIY+Herd; **automatic checking**
Does my MCM allow a given compiler optimisation?	manual c'examples [Vafeiaidis+ POPL'15]; manual proof [Sevcik PLDI'11];
Does my MCM allow a given compiler mapping?	manual c'examples [Wickerson+ OOPSLA'15]; manual proof [Batty+ POPL'11, Batty+ POPL'12];

Our contributions

Question	Existing work
Can a given litmus test pass under a given MCM?	CppMem, Herd, MemSAT, Nemos, ...
Which litmus tests can be run to check whether a machine conforms to a given MCM?	semi-automatic generation with DIY [Alglave+ CAV'10]; **automatic generation**
Is one MCM more permissive than another?	manual proof; manual examples [Batty+ POPL'16]; semi-automatic checking with DIY+Herd; **automatic checking**
Does my MCM allow a given compiler optimisation?	manual c'examples [Vafeiaidis+ POPL'15]; manual proof [Sevcik PLDI'11]; **automatic checking**
Does my MCM allow a given compiler mapping?	manual c'examples [Wickerson+ OOPSLA'15]; manual proof [Batty+ POPL'11, Batty+ POPL'12];

Our contributions

Question	Existing work
Can a given litmus test pass under a given MCM?	CppMem, Herd, MemSAT, Nemos, ...
Which litmus tests can be run to check whether a machine conforms to a given MCM?	semi-automatic generation with DIY [Alglave+ CAV'10]; **automatic generation**
Is one MCM more permissive than another?	manual proof; manual examples [Batty+ POPL'16]; semi-automatic checking with DIY+Herd; **automatic checking**
Does my MCM allow a given compiler optimisation?	manual c'examples [Vafeiaidis+ POPL'15]; manual proof [Sevcik PLDI'11]; **automatic checking**
Does my MCM allow a given compiler mapping?	manual c'examples [Wickerson+ OOPSLA'15]; manual proof [Batty+ POPL'11, Batty+ POPL'12]; **automatic checking**

Contents

- Context: memory consistency models (MCMs)
- Where our work fits in
- Key Ideas
- Applications

Key Idea 1

Key Idea 1

- What are **M**'s conformance tests?
  Find (P,σ) where $\sigma \notin obs_{\mathbf{M}}(P)$ and $\sigma \in obs_{\mathbf{0}}(P)$.

Key Idea 1

- What are **M**'s conformance tests?
  Find (P,σ) where $\sigma \notin obs_{\mathbf{M}}(P)$ and $\sigma \in obs_{\mathbf{0}}(P)$.

- Is **M** stronger than **N**?
  No if $\exists$(P,σ) where $\sigma \notin obs_{\mathbf{N}}(P)$ and $\sigma \in obs_{\mathbf{M}}(P)$.

Key Idea 1

- What are **M**'s conformance tests?
  Find (P,σ) where $\sigma \notin obs_{\mathbf{M}}(P)$ and $\sigma \in obs_{\mathbf{0}}(P)$.

- Is **M** stronger than **N**?
  No if ∃(P,σ) where $\sigma \notin obs_{\mathbf{N}}(P)$ and $\sigma \in obs_{\mathbf{M}}(P)$.

- Does **M** allow my optimisation?
  No if ∃(P,Q,σ) where $\sigma \notin obs_{\mathbf{M}}(P)$, $\sigma \in obs_{\mathbf{M}}(Q)$ and P optimises to Q.

Key Idea 1

- What are **M**'s conformance tests?
  Find (P,σ) where $\sigma \notin obs_{\mathbf{M}}(P)$ and $\sigma \in obs_{\mathbf{0}}(P)$.

- Is **M** stronger than **N**?
  No if ∃(P,σ) where $\sigma \notin obs_{\mathbf{N}}(P)$ and $\sigma \in obs_{\mathbf{M}}(P)$.

- Does **M** allow my optimisation?
  No if ∃(P,Q,σ) where $\sigma \notin obs_{\mathbf{M}}(P)$, $\sigma \in obs_{\mathbf{M}}(Q)$ and P optimises to Q.

- Can **M** be implemented by my mapping to **N**?
  No if ∃(P,Q,σ) where $\sigma \notin obs_{\mathbf{M}}(P)$, $\sigma \in obs_{\mathbf{N}}(Q)$ and P compiles to Q.

Key Idea 1

- What are **M**'s conformance tests?
  Find (P,σ) where $\sigma \notin obs_{\mathbf{M}}(P)$ and $\sigma \in obs_{\mathbf{0}}(P)$.

- Is **M** stronger than **N**?
  No if ∃(P,σ) where $\sigma \notin obs_{\mathbf{N}}(P)$ and $\sigma \in obs_{\mathbf{M}}(P)$.

- Does **M** allow my optimisation?
  No if ∃(P,Q,σ) where $\sigma \notin obs_{\mathbf{M}}(P)$, $\sigma \in obs_{\mathbf{M}}(Q)$ and P optimises to Q.

- Can **M** be implemented by my mapping to **N**?
  No if ∃(P,Q,σ) where $\sigma \notin obs_{\mathbf{M}}(P)$, $\sigma \in obs_{\mathbf{N}}(Q)$ and P compiles to Q.

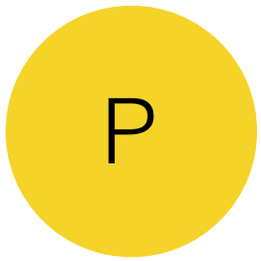
$$\{(P,Q,\sigma) \mid \sigma \notin obs_{\mathbf{M}}(P) \wedge \sigma \in obs_{\mathbf{N}}(Q) \wedge P \blacktriangleright Q\}$$

Key Idea 2

$$\{(P,Q,\sigma) \mid \sigma \notin \mathrm{obs}_{\mathbf{M}}(P) \wedge \sigma \in \mathrm{obs}_{\mathbf{N}}(Q) \wedge P \blacktriangleright Q\}$$

Key Idea 2

$$\{(P,Q,\sigma) \mid \sigma \notin obs_{M}(P) \wedge \sigma \in obs_{N}(Q) \wedge P \blacktriangleright Q\}$$

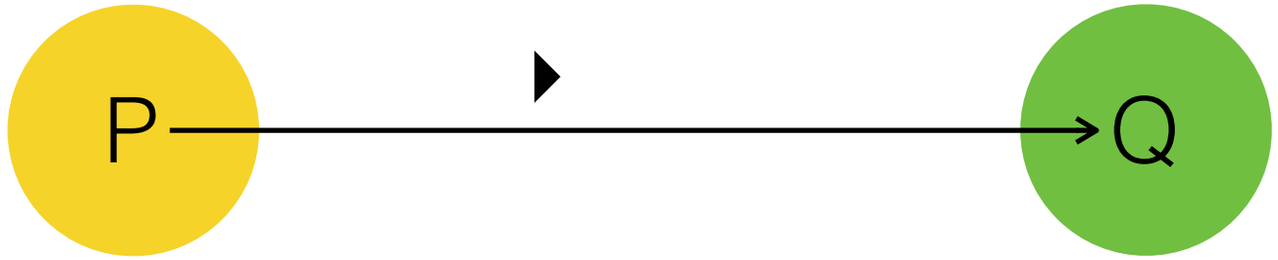


Key Idea 2

$$\{(P,Q,\sigma) \mid \sigma \notin \mathrm{obs}_{\mathbf{M}}(P) \wedge \sigma \in \mathrm{obs}_{\mathbf{N}}(Q) \wedge P \blacktriangleright Q\}$$

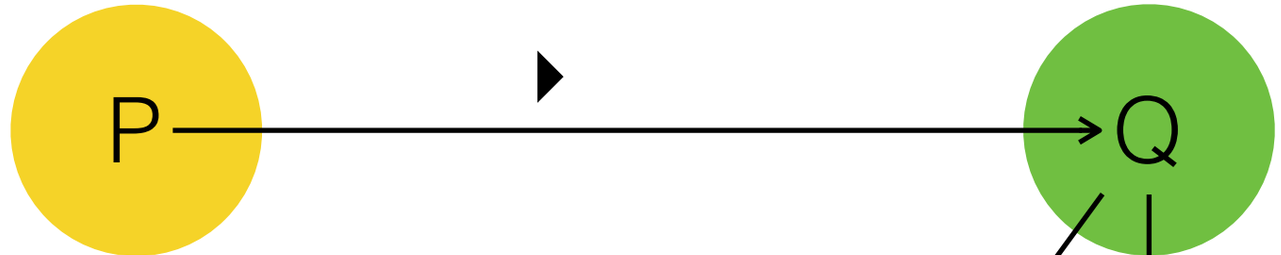
Key Idea 2

$$\{(P,Q,\sigma) \mid \sigma \notin \mathrm{obs}_{\mathbf{M}}(P) \wedge \sigma \in \mathrm{obs}_{\mathbf{N}}(Q) \wedge P \blacktriangleright Q\}$$



Key Idea 2

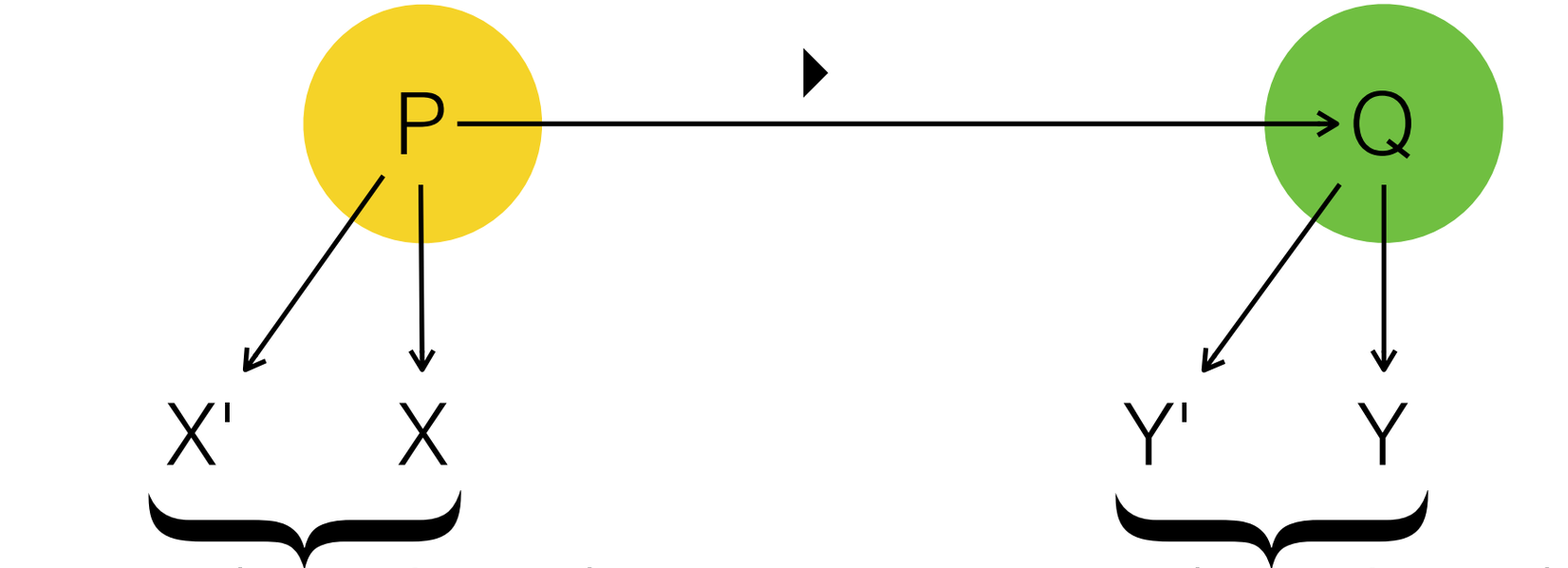
$$\{(P,Q,\sigma) \mid \sigma \notin \mathrm{obs}_{\mathbf{M}}(P) \wedge \sigma \in \mathrm{obs}_{\mathbf{N}}(Q) \wedge P \blacktriangleright Q\}$$



some (consistent) execution must reach σ

Key Idea 2

$$\{(P,Q,\sigma) \mid \sigma \notin \mathrm{obs}_{\mathbf{M}}(P) \wedge \sigma \in \mathrm{obs}_{\mathbf{N}}(Q) \wedge P \blacktriangleright Q\}$$



every (consistent) execution must not reach σ

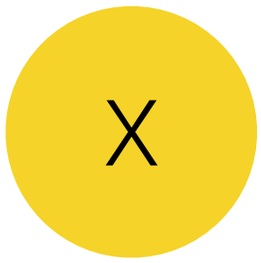
some (consistent) execution must reach σ

Key Idea 2

$$\{(P,Q,\sigma) \mid \sigma \notin obs_{M}(P) \wedge \sigma \in obs_{N}(Q) \wedge P \blacktriangleright Q\}$$

Key Idea 2

$$\{(P,Q,\sigma) \mid \sigma \notin \mathrm{obs}_{\mathbf{M}}(P) \wedge \sigma \in \mathrm{obs}_{\mathbf{N}}(Q) \wedge P \blacktriangleright Q\}$$

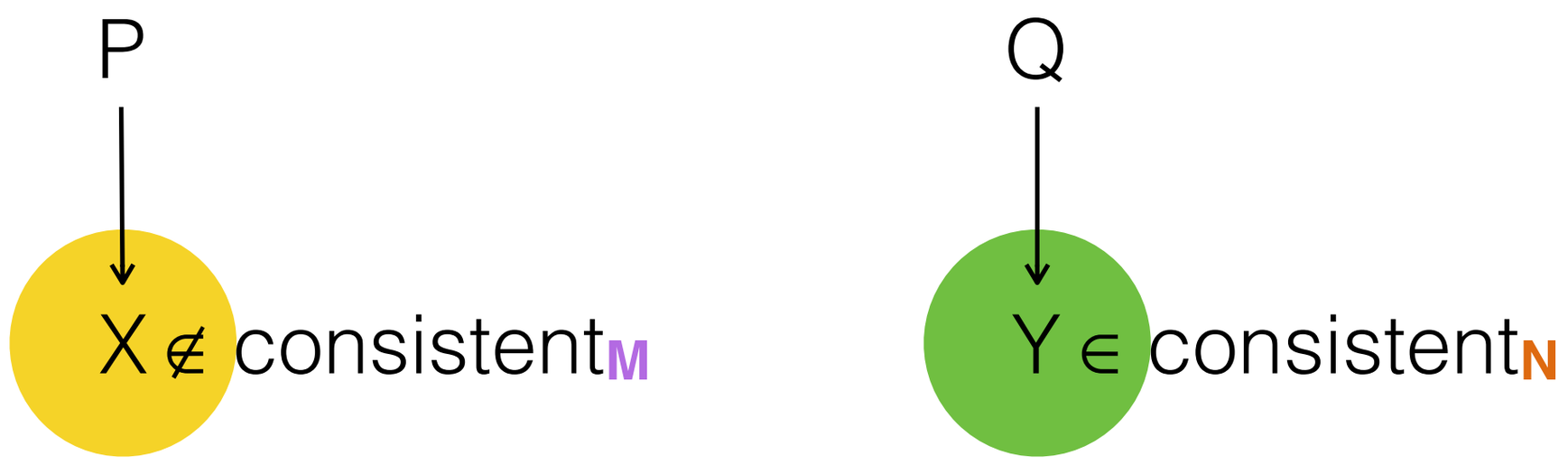


Key Idea 2

$$\{(P,Q,\sigma) \mid \sigma \notin \mathrm{obs}_{\mathbf{M}}(P) \wedge \sigma \in \mathrm{obs}_{\mathbf{N}}(Q) \wedge P \blacktriangleright Q\}$$

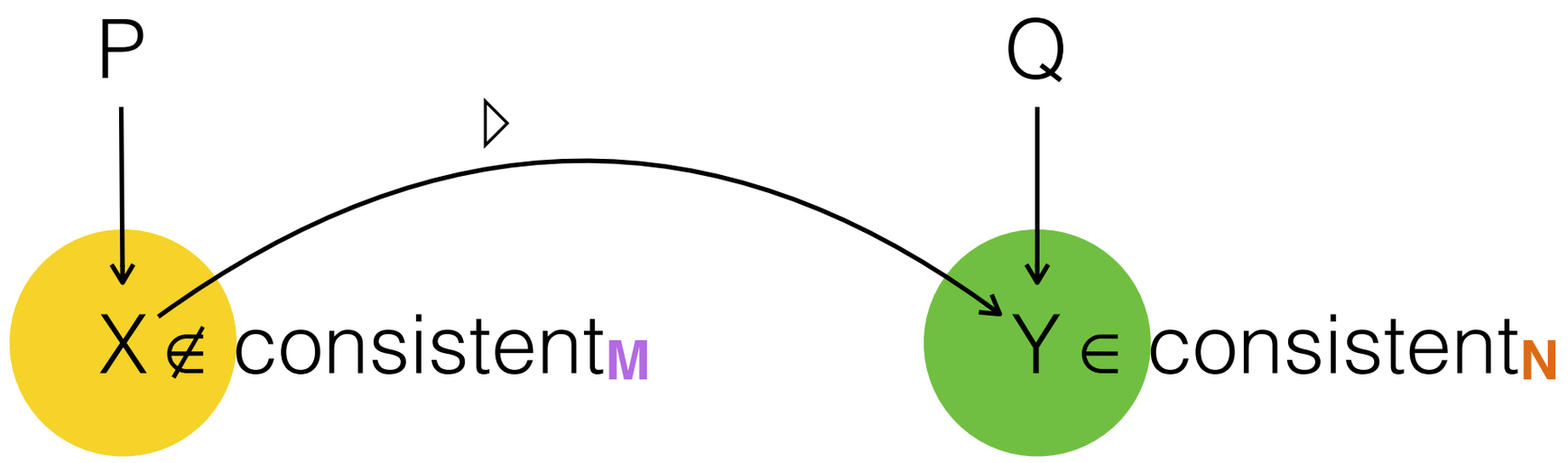
Key Idea 2

$$\{(P,Q,\sigma) \mid \sigma \notin \mathrm{obs}_{\mathbf{M}}(P) \wedge \sigma \in \mathrm{obs}_{\mathbf{N}}(Q) \wedge P \blacktriangleright Q\}$$



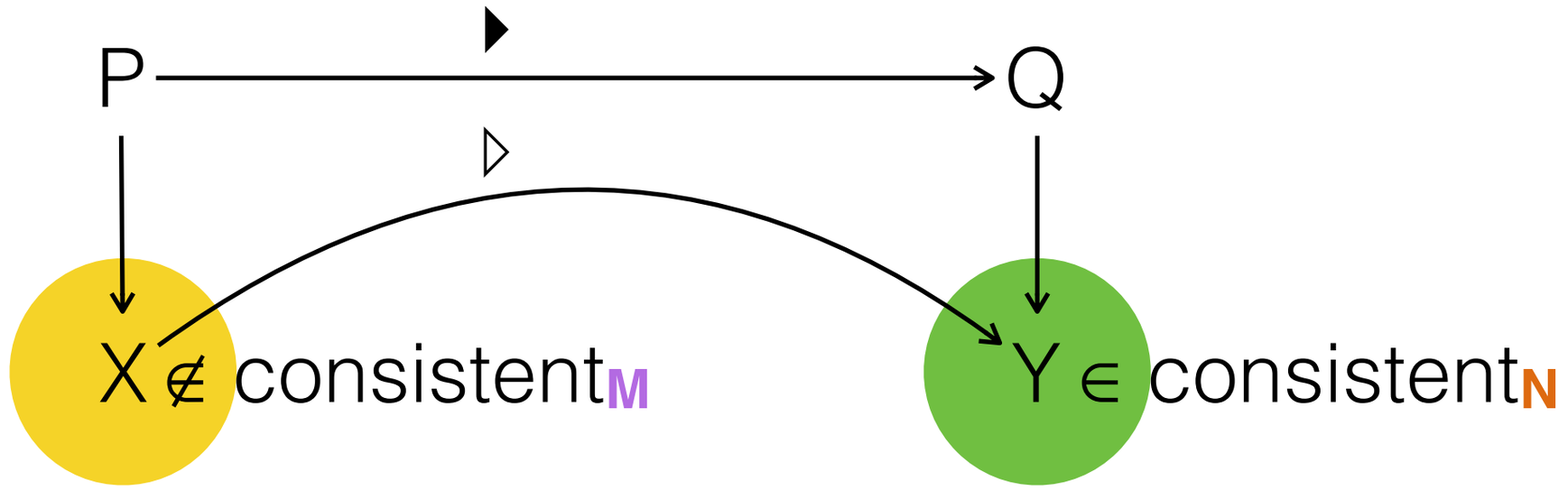
Key Idea 2

$$\{(P,Q,\sigma) \mid \sigma \notin \mathrm{obs}_{\mathbf{M}}(P) \wedge \sigma \in \mathrm{obs}_{\mathbf{N}}(Q) \wedge P \blacktriangleright Q\}$$



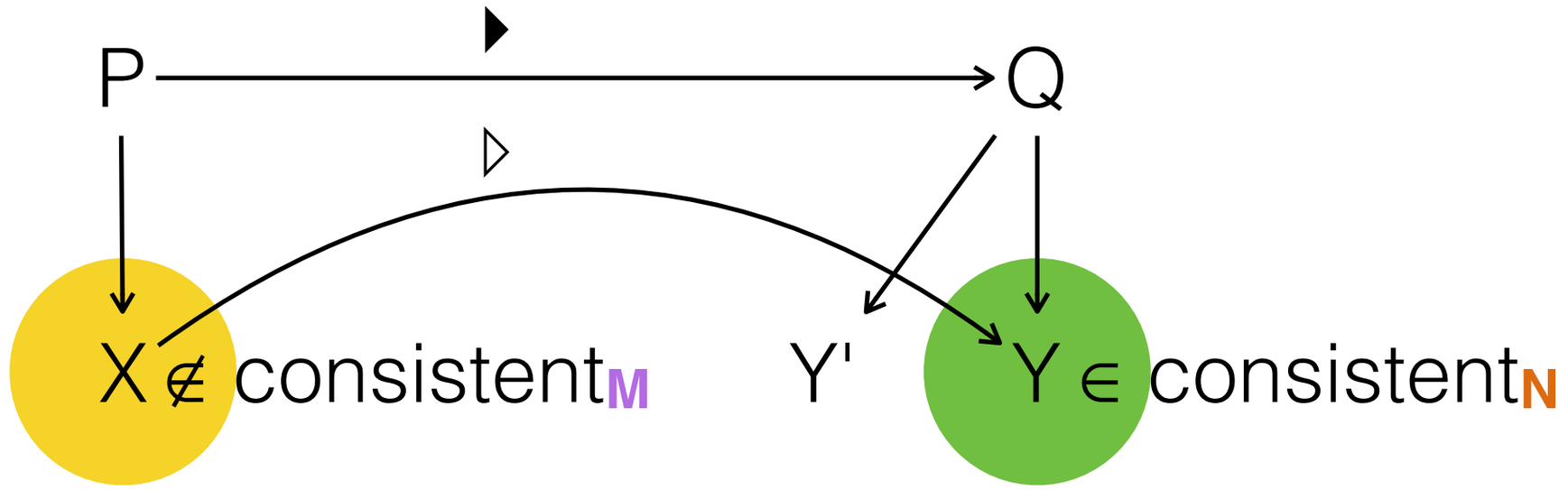
Key Idea 2

$$\{(P,Q,\sigma) \mid \sigma \notin \mathrm{obs}_{\mathbf{M}}(P) \wedge \sigma \in \mathrm{obs}_{\mathbf{N}}(Q) \wedge P \blacktriangleright Q\}$$



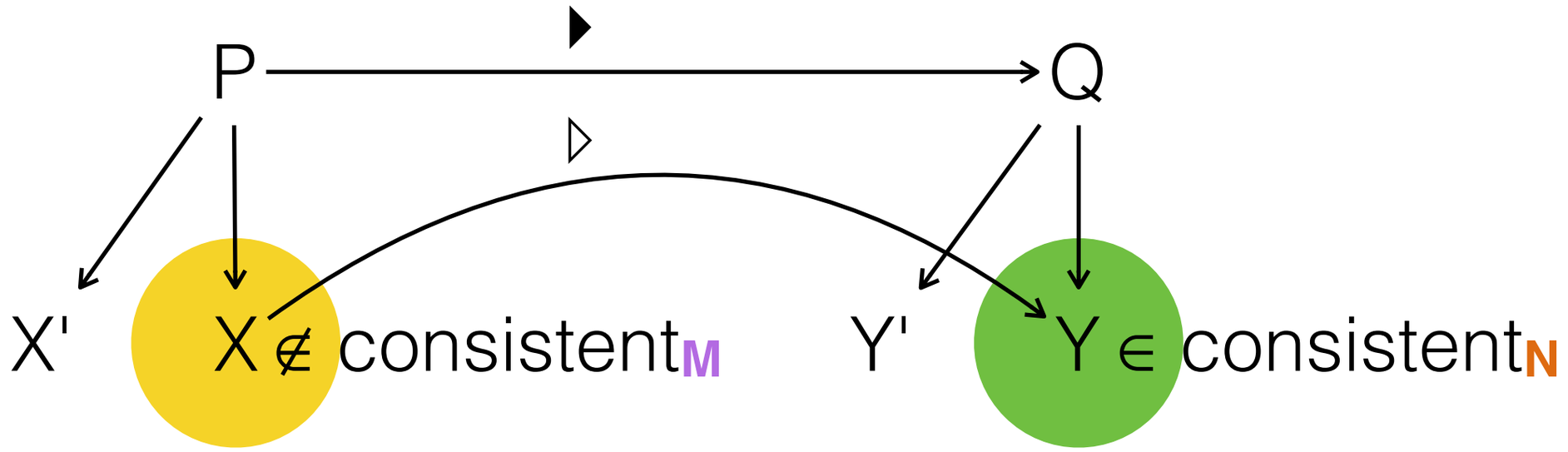
Key Idea 2

$$\{(P,Q,\sigma) \mid \sigma \notin \mathrm{obs}_{\mathbf{M}}(P) \wedge \sigma \in \mathrm{obs}_{\mathbf{N}}(Q) \wedge P \blacktriangleright Q\}$$



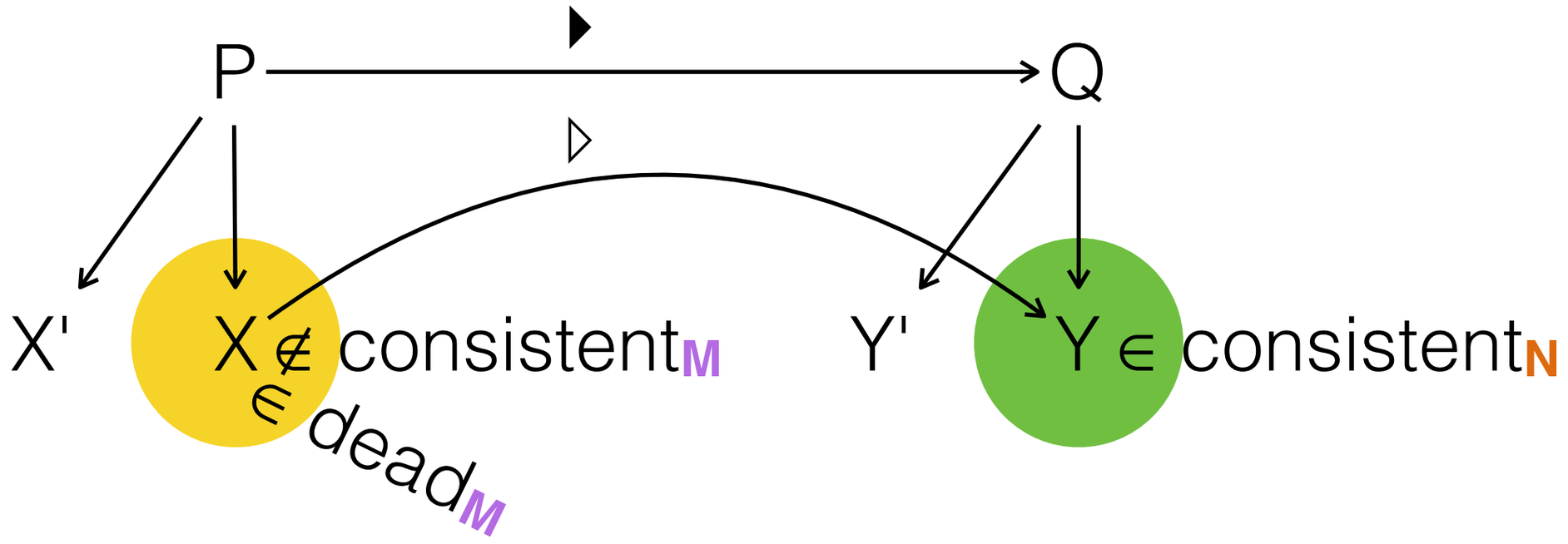
Key Idea 2

$$\{(P,Q,\sigma) \mid \sigma \notin \mathrm{obs}_{\mathbf{M}}(P) \wedge \sigma \in \mathrm{obs}_{\mathbf{N}}(Q) \wedge P \blacktriangleright Q\}$$



Key Idea 2

$$\{(P,Q,\sigma) \mid \sigma \notin \mathrm{obs}_{M}(P) \wedge \sigma \in \mathrm{obs}_{N}(Q) \wedge P \blacktriangleright Q\}$$



The Alloy Constraint Solver

/Users/jpw48/git/memalloy/models/tests/Q2_c11_sra_simp/question.als

New Open Reload Save Execute Show

```
open ../../sw/c11_nafence[E] as M1
open ../../sw/c11_simp[E] as M2

sig E {}

pred gp [X : Exec_C] {

  // Prefer solutions with total sb per thread
  total_sb[X]

  // ignore RMWs
  no_RMWs[X]

  // The execution is forbidden in M1
  not(M1/consistent[X])
  //M1/dead[X]

  // The execution is allowed (and not faulty)
  M2/consistent[X]

}

run gp for 1 Exec, 5 E, 3 Int
```

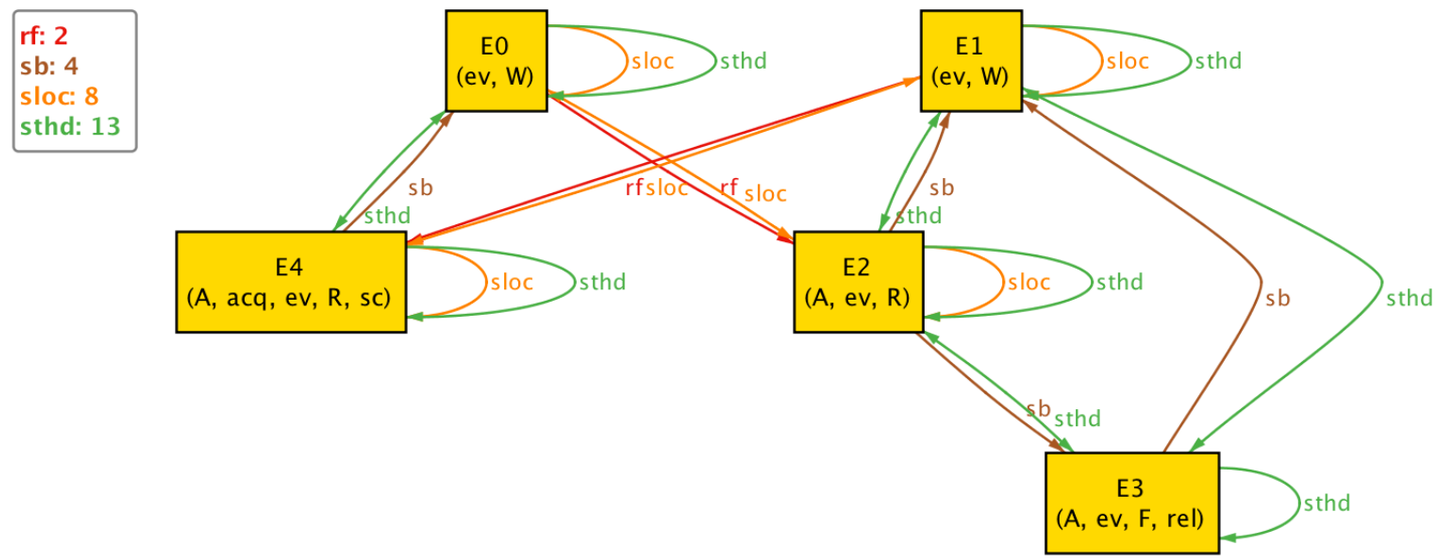
```
Alloy Analyzer 4.2.? (build date: unknown)

Executing "Run gp for 3 int, 1 Exec, 5 E"
   Solver=glucose(jni) Bitwidth=3 MaxSeq=3 Symmetry=20
   13526 vars. 258 primary vars. 43898 clauses. 1466ms.
   Instance found. Predicate is consistent. 756ms.
```

(question) Run gp for 3 int, 1 Exec, 5 E

Viz Txt Tree Theme Magic Layout Evaluator Next

Projected over M1/c11_nafence_base/exec_C/exec/Exec



<< M1/c11_nafence_base/exec_C/Exec_C$0 >>

Contents

- Context: memory consistency models (MCMs)
- Where our work fits in
- Key Ideas
- Applications

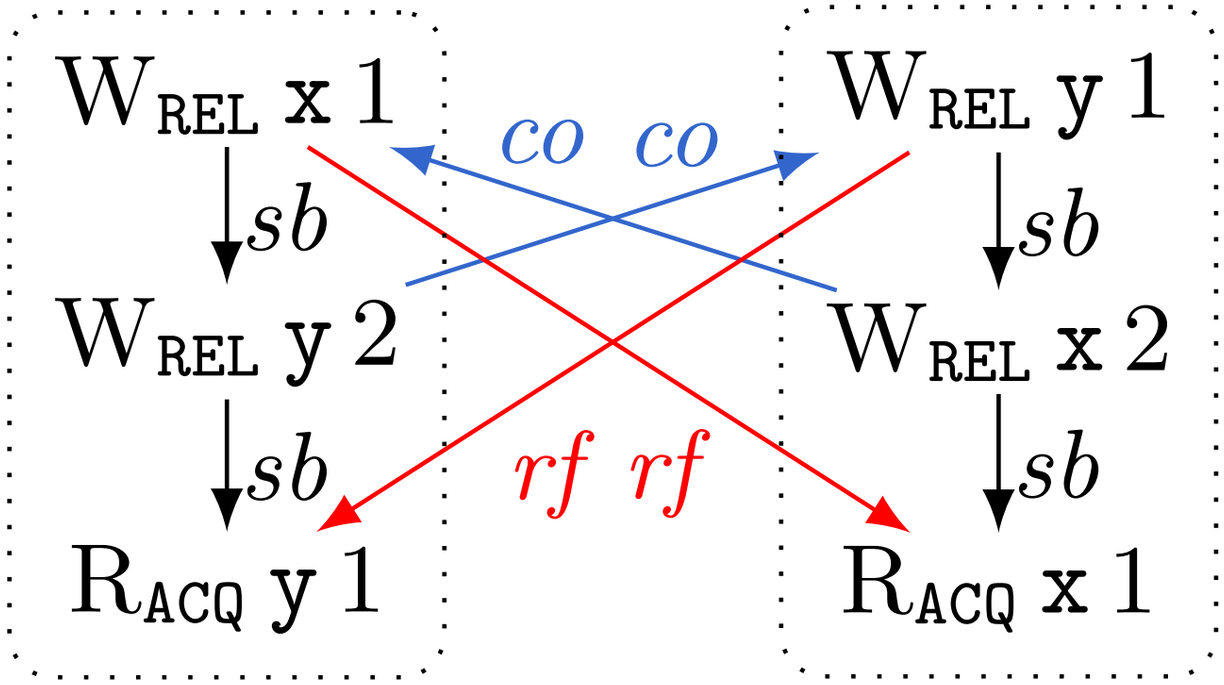
Comparing "strong release-acquire" to original release-acquire

Taming Release-Acquire Consistency

Ori Lahav Nick Giannarakis Viktor Vafeiadis

for Software Systems (MPI-SWS), Germany

Comparing "strong release-acquire" to original release-acquire




```
atomic_int x=0,y=0;
  x.store(1,REL);
  y.store(2,REL);
  r0=y.load(ACQ);
  ||
  y.store(1,REL);
  x.store(2,REL);
  r1=x.load(ACQ);
r0==1 && r1==1
```

Taming Release-Acquire Consistency

Ori Lahav Nick Giannarakis Viktor Vafeiadis

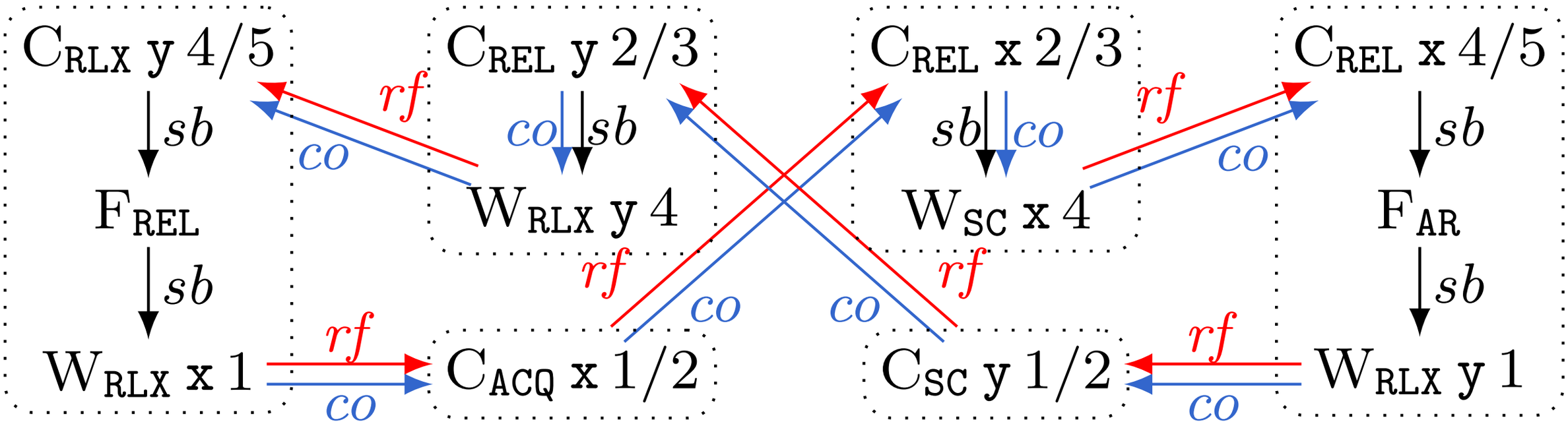
for Software Systems (MPI-SWS), Germany

Comparing Nienhuis et al.'s C++ variant to the original

- Cf. Nienhuis et al. (OOPSLA '16):

Comparing Nienhuis et al.'s C++ variant to the original

- Cf. Nienhuis et al. (OOPSLA '16):



Comparing Batty et al.'s C++ variant to the original

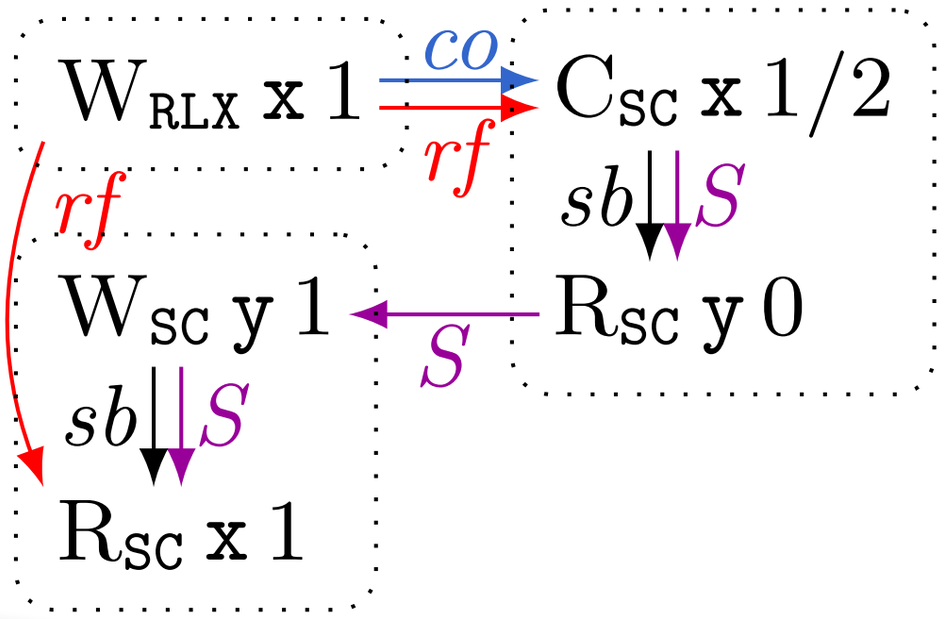
Overhauling SC Atomics in C11 and OpenCL

Mark Batty
University of Kent, UK
m.j.batty@kent.ac.uk

Alastair F. Donaldson
Imperial College London, UK
alastair.donaldson@imperial.ac.uk

John Wickerson
Imperial College London, UK
j.wickerson@imperial.ac.uk

Comparing Batty et al.'s C++ variant to the original




```
atomic_int x=0,y=0;
  x.store(1,RLX);
  ||
  r0=x.cas(1,2,SC,RLX);
  r1=y.load(SC);
  ||
  y.store(1,SC);
  r2=x.load(SC);
r0==true && r1==0 && r2==1
```

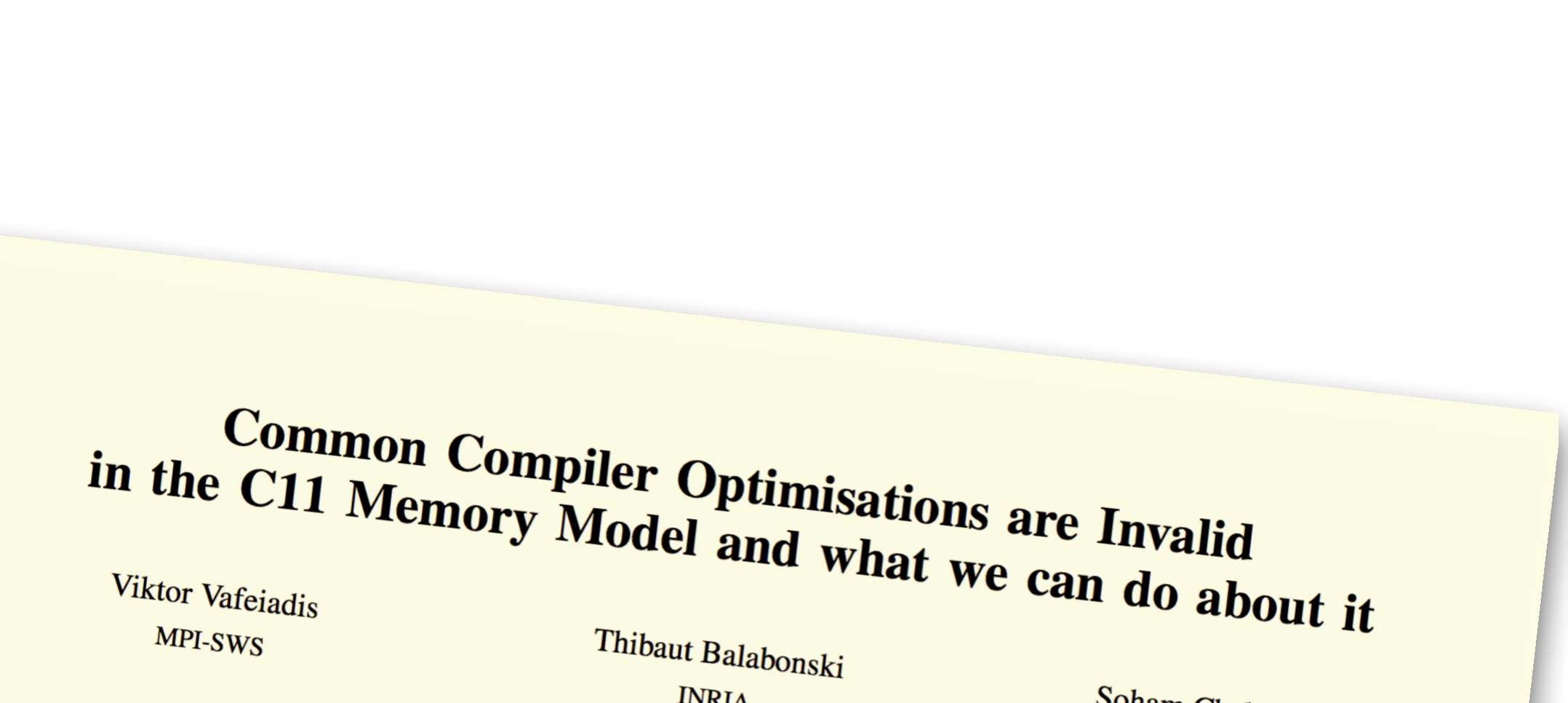
Overhauling SC Atomics in C11 and OpenCL

Mark Batty
University of Kent, UK
m.j.batty@kent.ac.uk

Alastair F. Donaldson
Imperial College London, UK
alastair.donaldson@imperial.ac.uk

John Wickerson
Imperial College London, UK
j.wickerson@imperial.ac.uk

Does C++ allow "linearisation"?

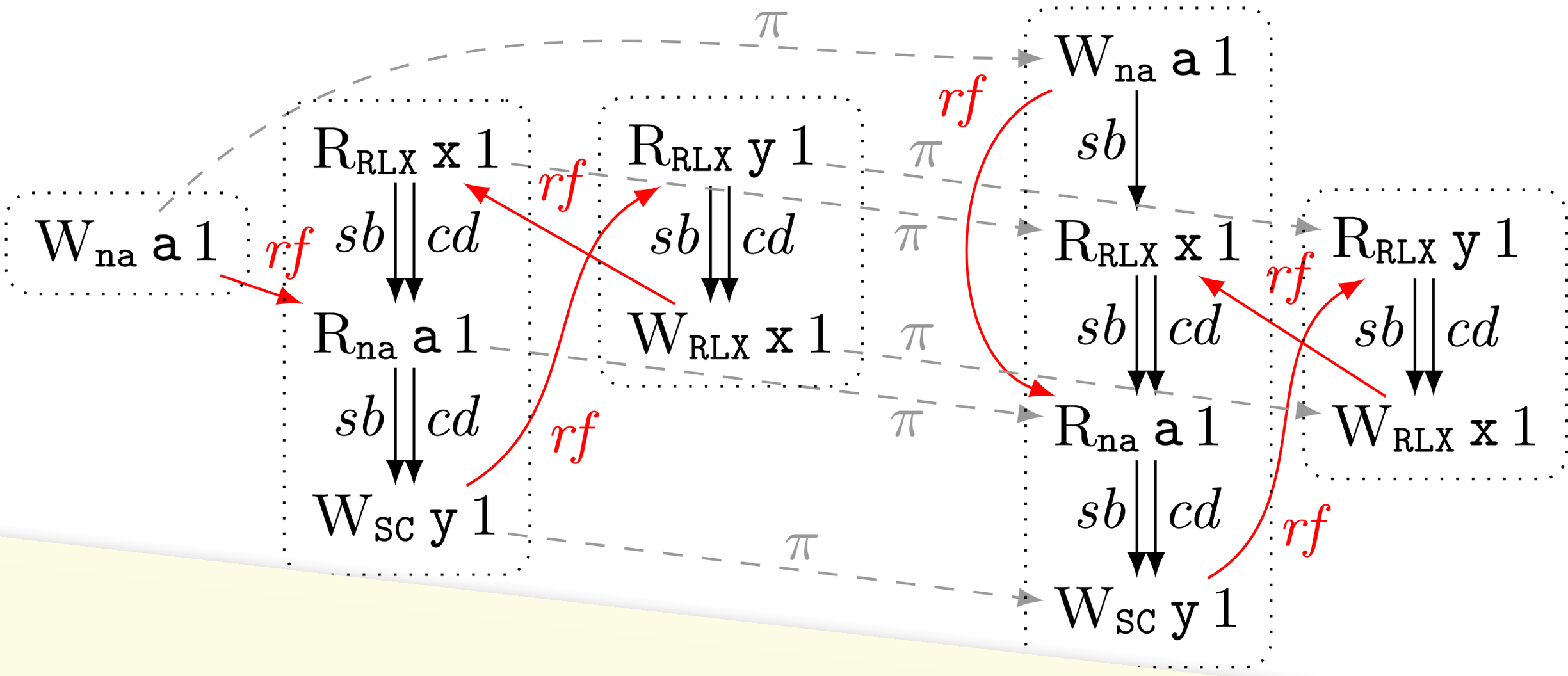


Common Compiler Optimisations are Invalid in the C11 Memory Model and what we can do about it

Viktor Vafeiadis
MPI-SWS

Thibaut Balabonski
INRIA

Does C++ allow "linearisation"?



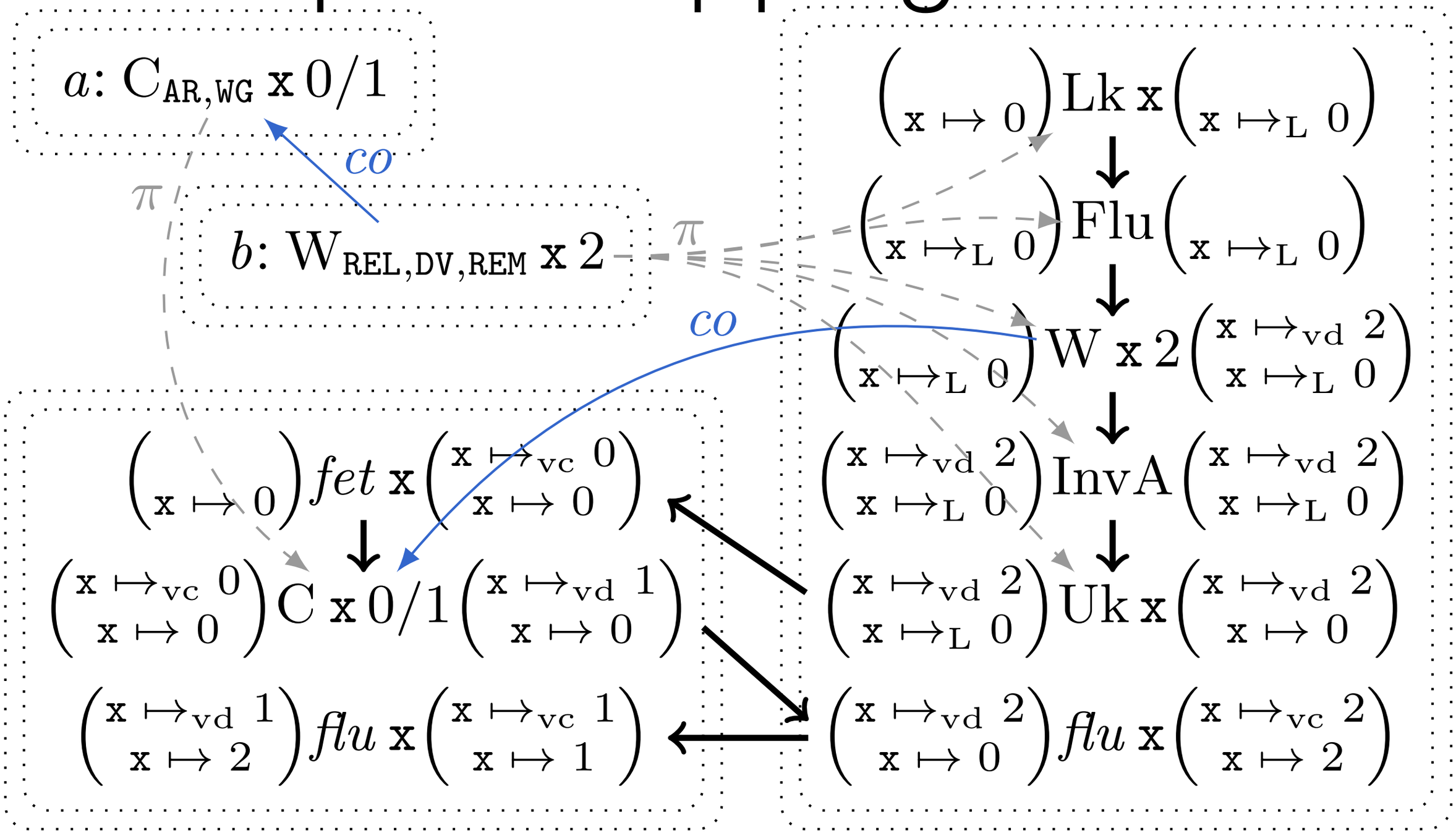
Common Compiler Optimisations are Invalid in the C11 Memory Model and what we can do about it

Viktor Vafeiadis
MPI-SWS

Thibaut Balabonski
INRIA

Is AMD's OpenCL compiler mapping sound?

Is AMD's OpenCL compiler mapping sound?

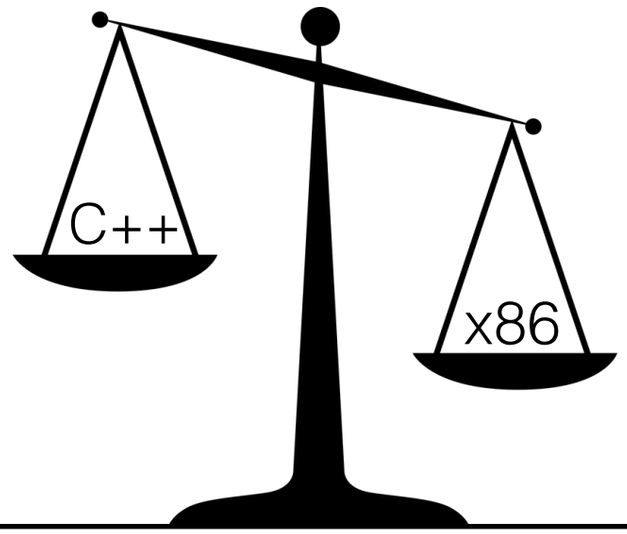


Checking and fixing an OpenCL/PTX compiler mapping

- PTX MCM proposed by Alglave et al. (ASPLOS '15)
- "Obvious" OpenCL/PTX mapping is invalid
- Manually revise PTX MCM (to obtain "PTX2")
- Now mapping is valid
- Run litmus tests that distinguish PTX/PTX2 against GPU hardware to validate PTX2

Contents

- Context: memory consistency models (MCMs)
- Where our work fits in
- Key Ideas
- Applications



Automatically Comparing Memory Consistency Models

John Wickerson (Imperial)
Mark Batty (U Kent)
Tyler Sorensen (Imperial)
George A. Constantinides (Imperial)

Newcastle University
Wednesday 31 August 2016