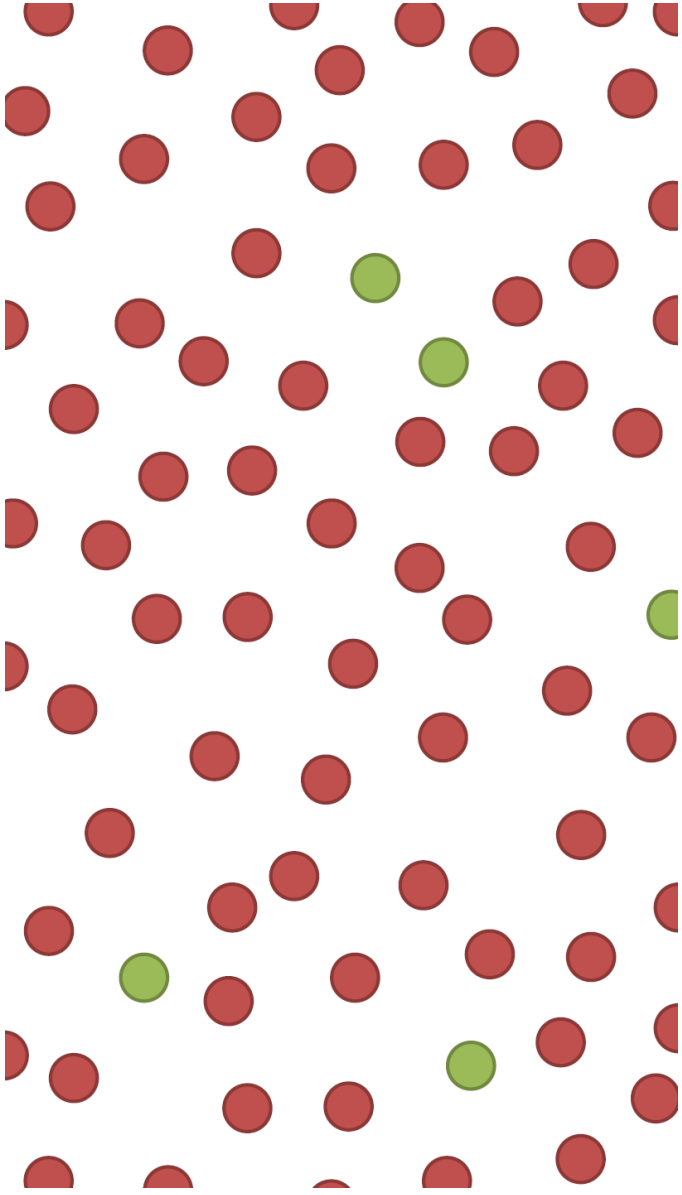
An operational semantics of C11 concurrency

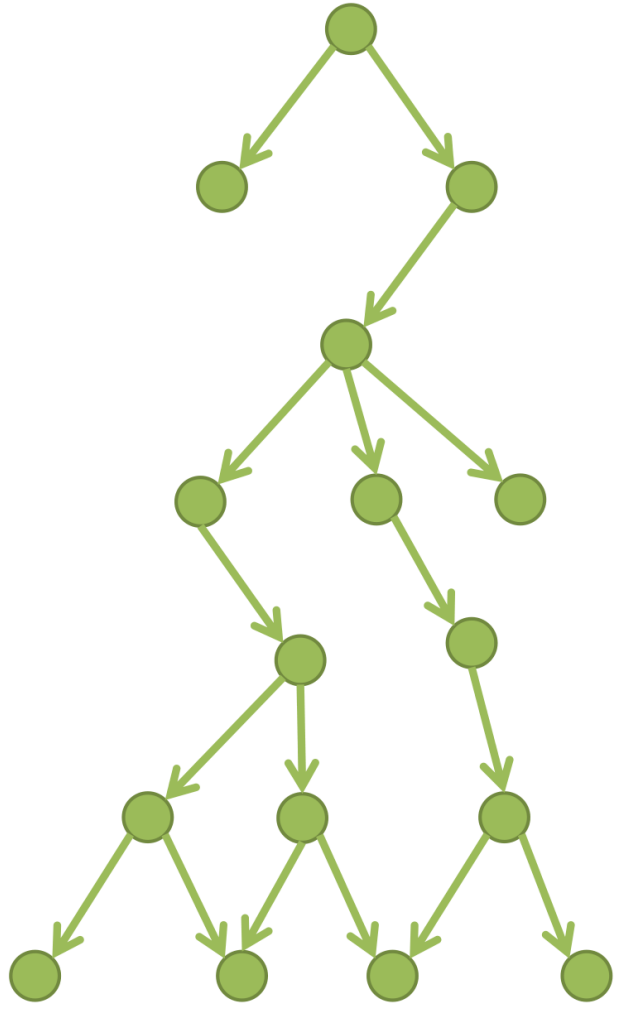
Kyndylan Nienhuis

September 24, 2014

Axiomatic



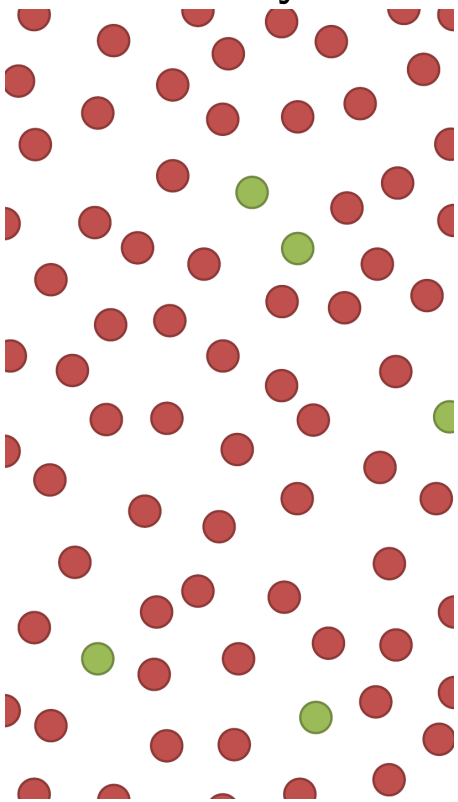
Operational



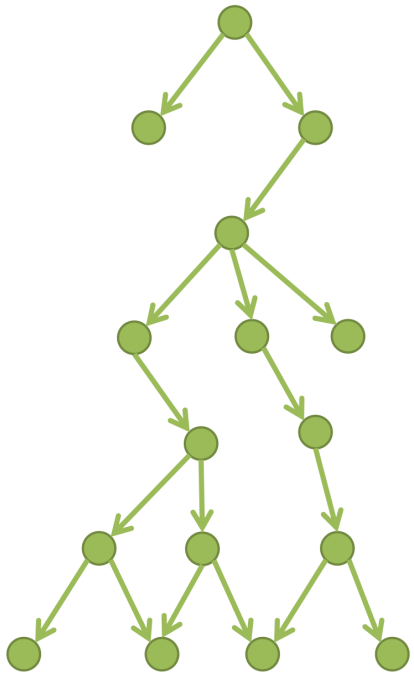
Why do we want an operational semantics of concurrency?

The C11 standard uses both styles of semantics:

An axiomatic style for concurrency

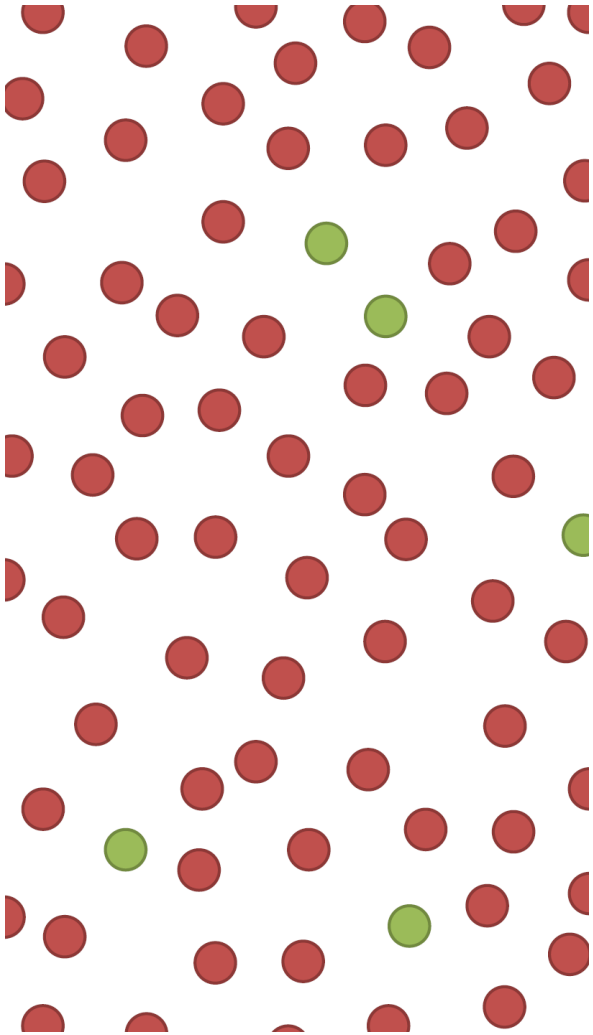


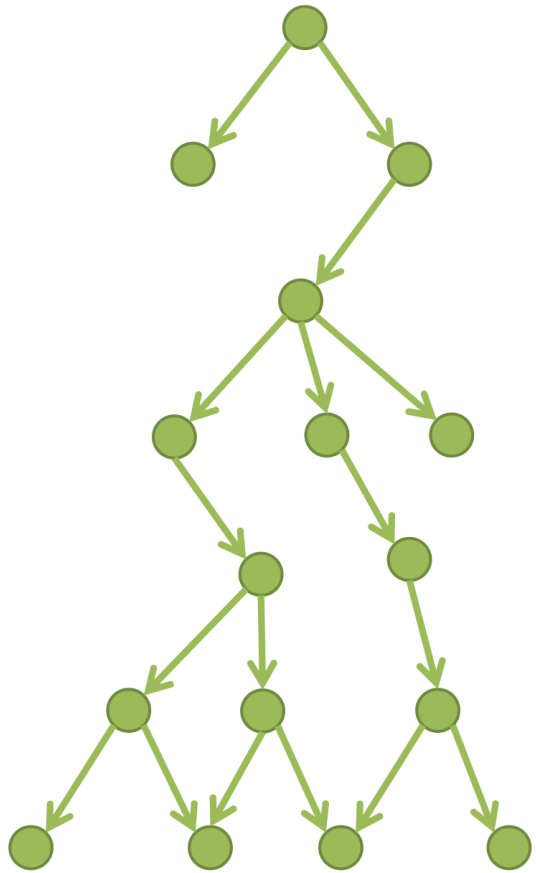
and an operational style for the rest



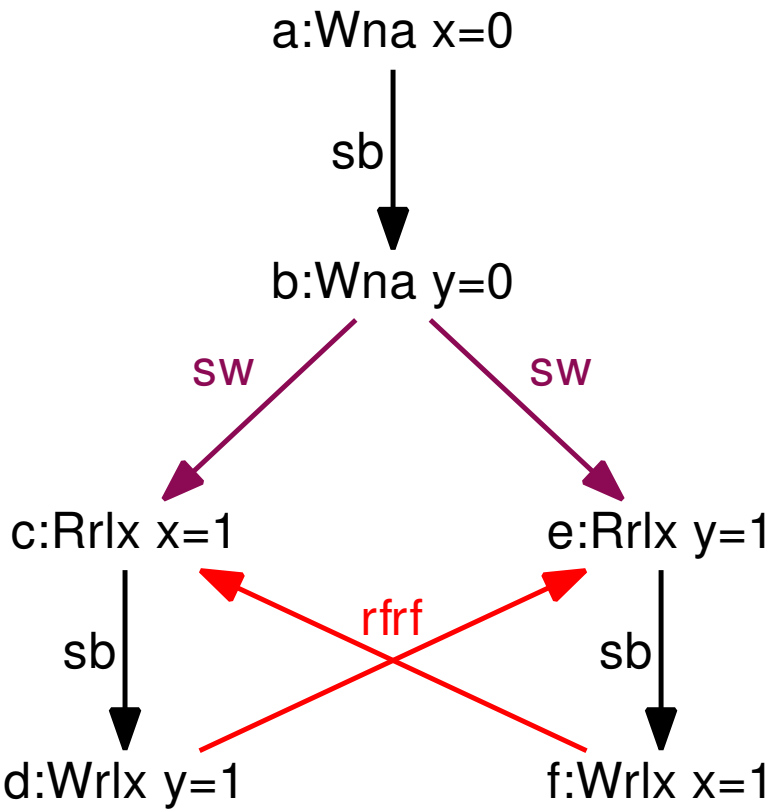
Why do we want an operational semantics of concurrency?

So we can compute (some) behaviour of large or infinite programs.



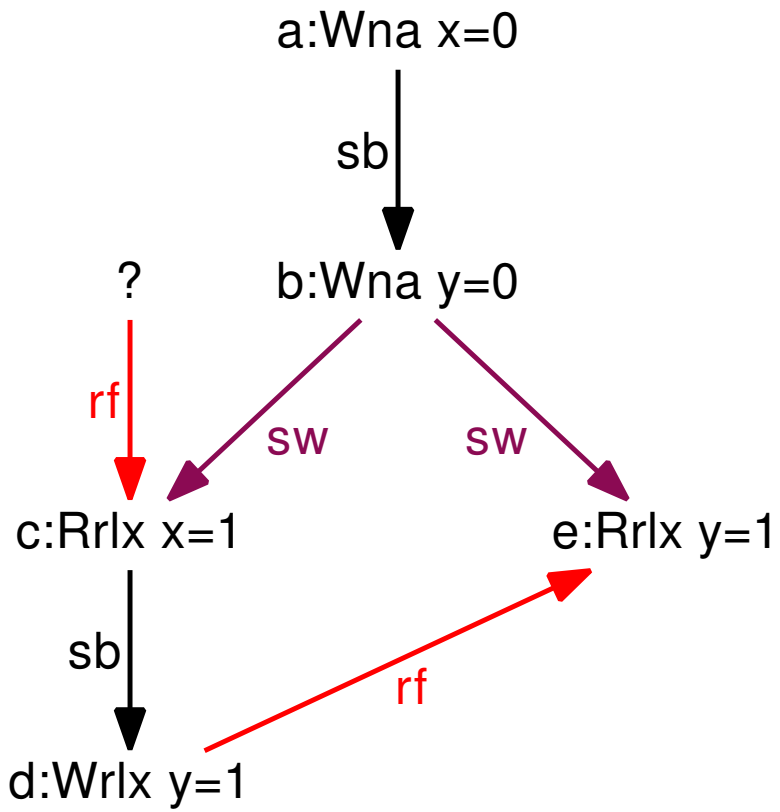


Generating relaxed behaviour



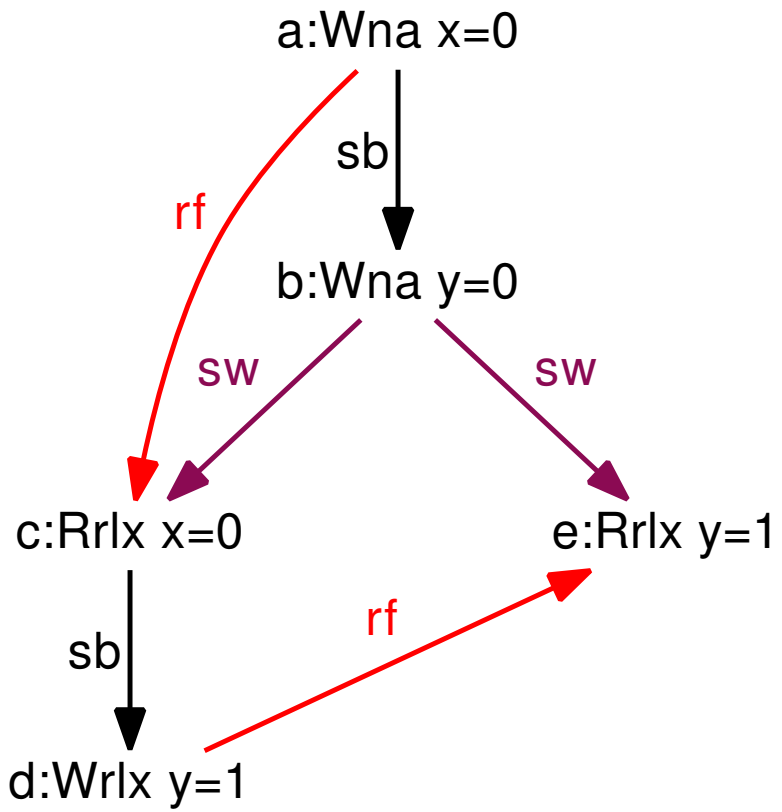

```
atomic_int x=0;
atomic_int y=0;
{{{
r1 = x.load(mo_relaxed);
y.store(1, mo_relaxed);
|||
r2 = y.load(mo_relaxed);
x.store(1, mo_relaxed);
}}}
```

Generating relaxed behaviour




```
atomic_int x=0;
atomic_int y=0;
{{{
r1 = x.load(mo_relaxed);
y.store(1, mo_relaxed);
|||
r2 = y.load(mo_relaxed);
x.store(1, mo_relaxed);
}}}
```

Generating relaxed behaviour




```
atomic_int x=0;
atomic_int y=0;
{{{
r1 = x.load(mo_relaxed);
y.store(1, mo_relaxed);
|||
r2 = y.load(mo_relaxed);
x.store(1, mo_relaxed);
}}}
```

Generating relaxed behaviour

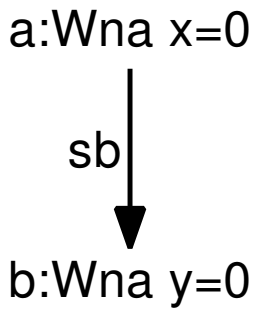
a:Wna x=0

The model has two parts:

- A threadlocal semantics that determines the the nodes, sb and asw
- A concurrency model that determines rf

They can take steps independently of each other. To allow this, the threadlocal semantics is symbolic.

Generating relaxed behaviour

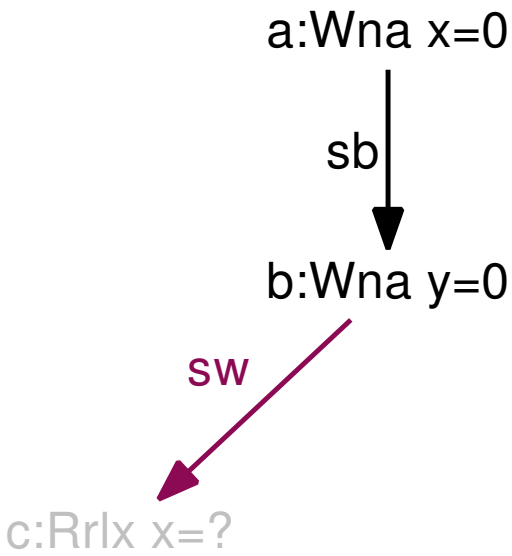


The model has two parts:

- A threadlocal semantics that determines the the nodes, sb and asw
- A concurrency model that determines rf

They can take steps independently of each other. To allow this, the threadlocal semantics is symbolic.

Generating relaxed behaviour

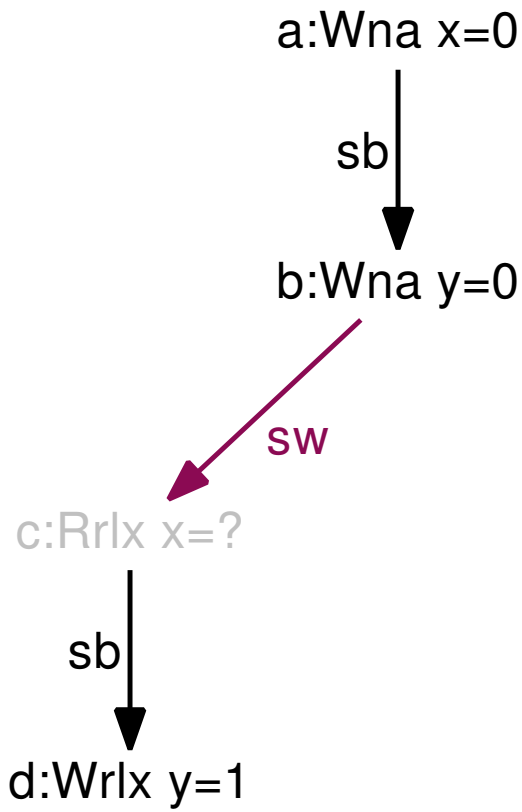


The model has two parts:

- A threadlocal semantics that determines the the nodes, sb and asw
- A concurrency model that determines rf

They can take steps independently of each other. To allow this, the threadlocal semantics is symbolic.

Generating relaxed behaviour

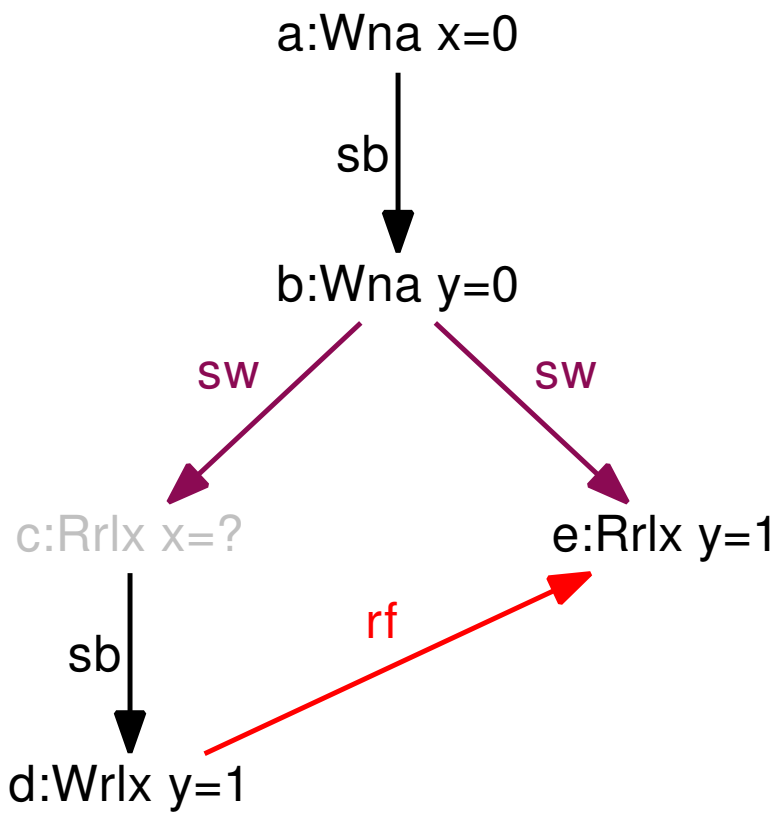


The model has two parts:

- A threadlocal semantics that determines the the nodes, sb and asw
- A concurrency model that determines rf

They can take steps independently of each other. To allow this, the threadlocal semantics is symbolic.

Generating relaxed behaviour

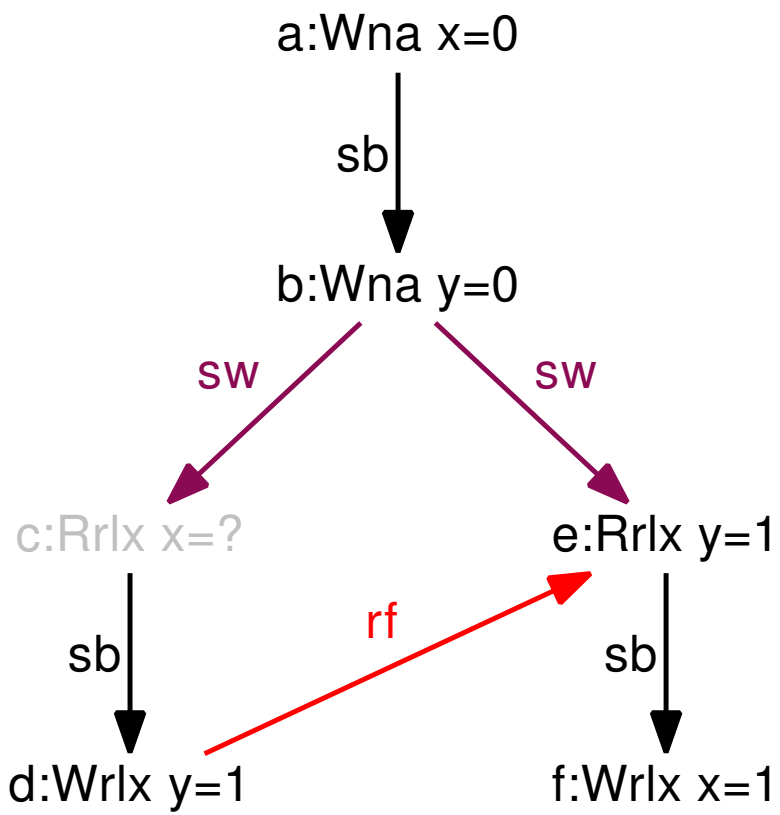


The model has two parts:

- A threadlocal semantics that determines the the nodes, sb and asw
- A concurrency model that determines rf

They can take steps independently of each other. To allow this, the threadlocal semantics is symbolic.

Generating relaxed behaviour

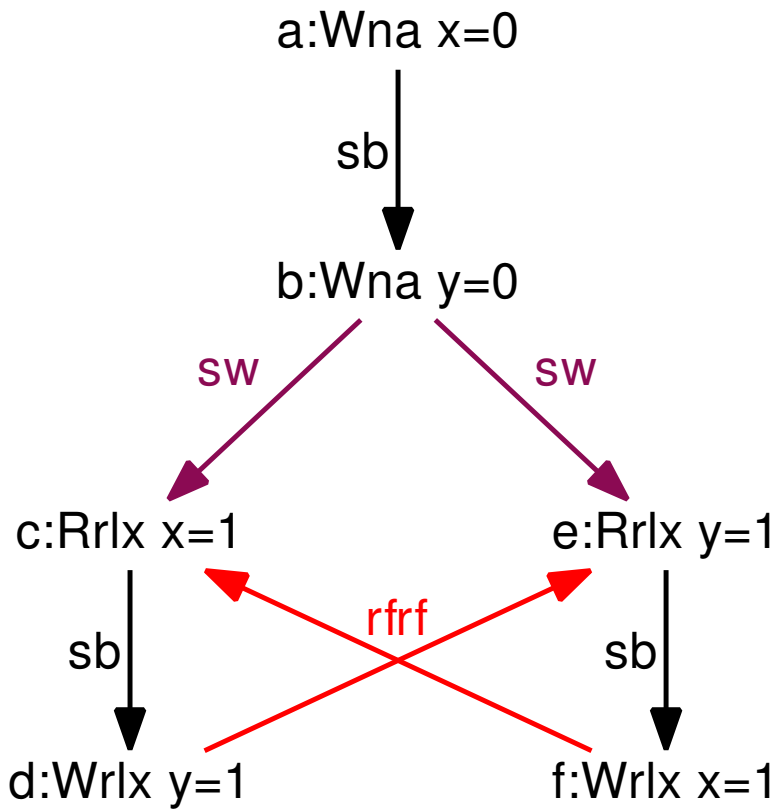


The model has two parts:

- A threadlocal semantics that determines the the nodes, sb and asw
- A concurrency model that determines rf

They can take steps independently of each other. To allow this, the threadlocal semantics is symbolic.

Generating relaxed behaviour

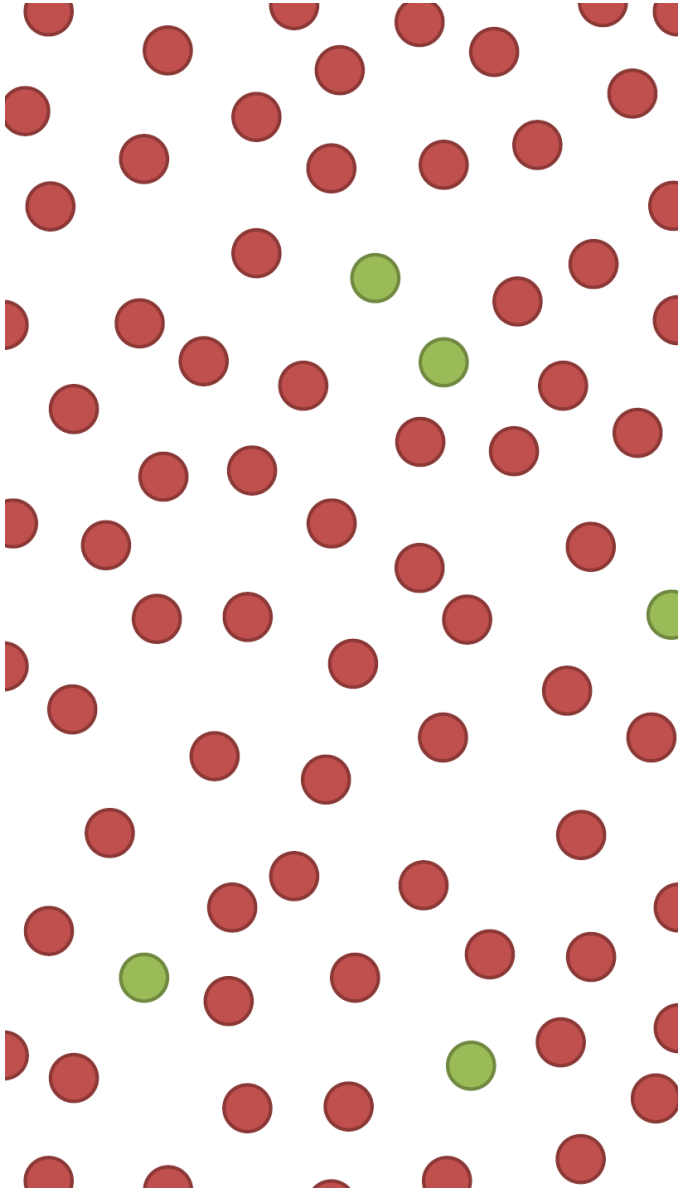


The model has two parts:

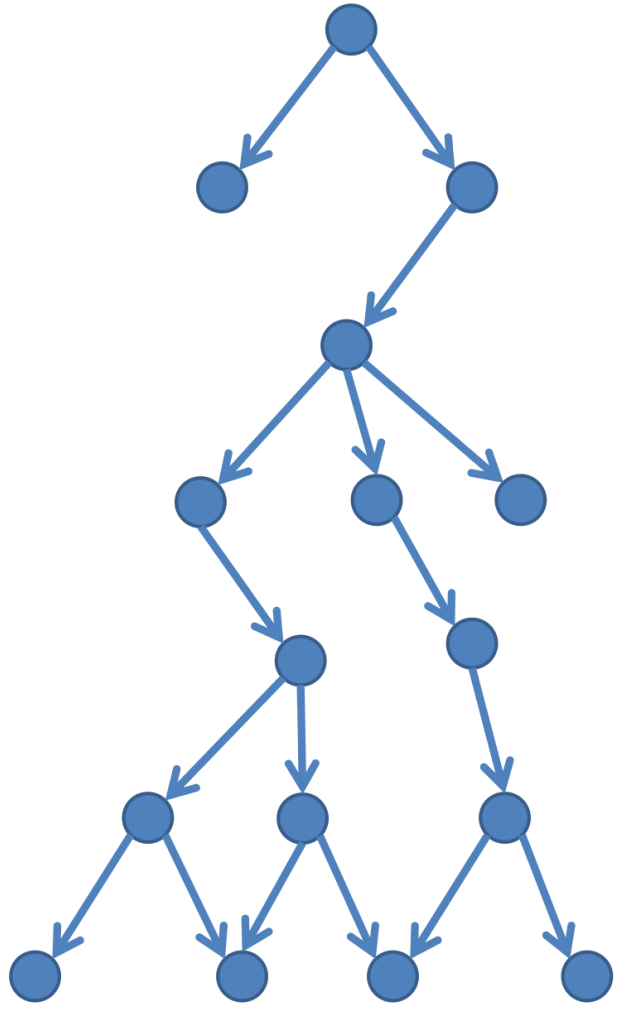
- A threadlocal semantics that determines the the nodes, sb and asw
- A concurrency model that determines rf

They can take steps independently of each other. To allow this, the threadlocal semantics is symbolic.

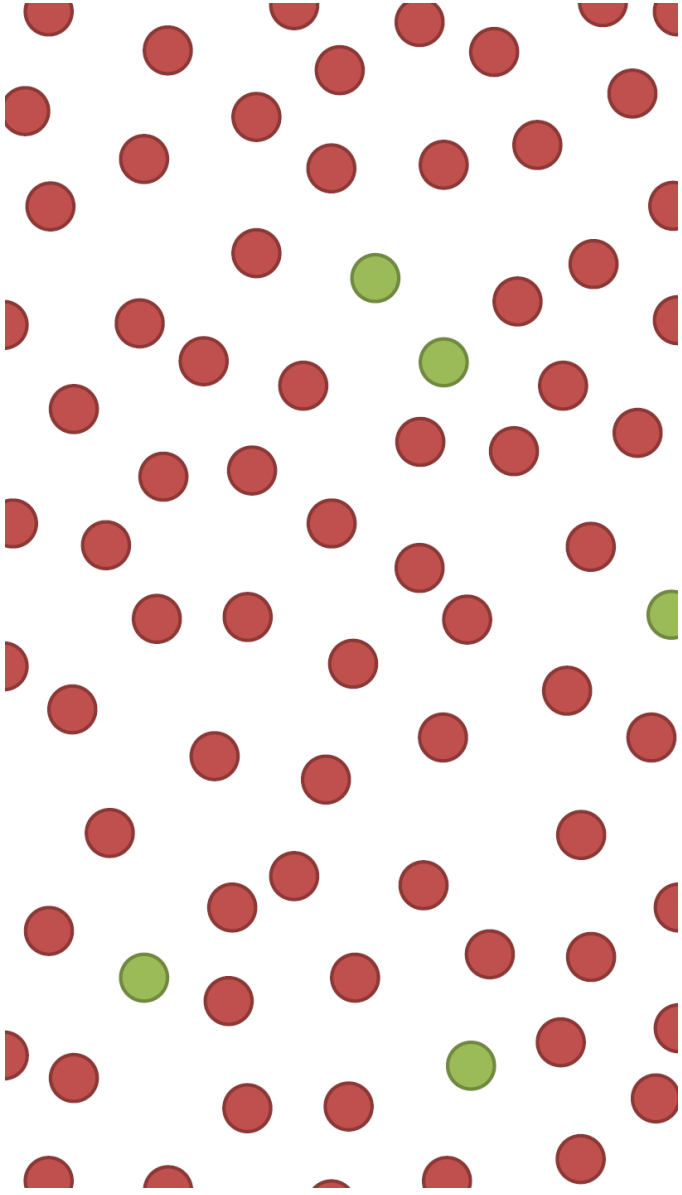
Axiomatic



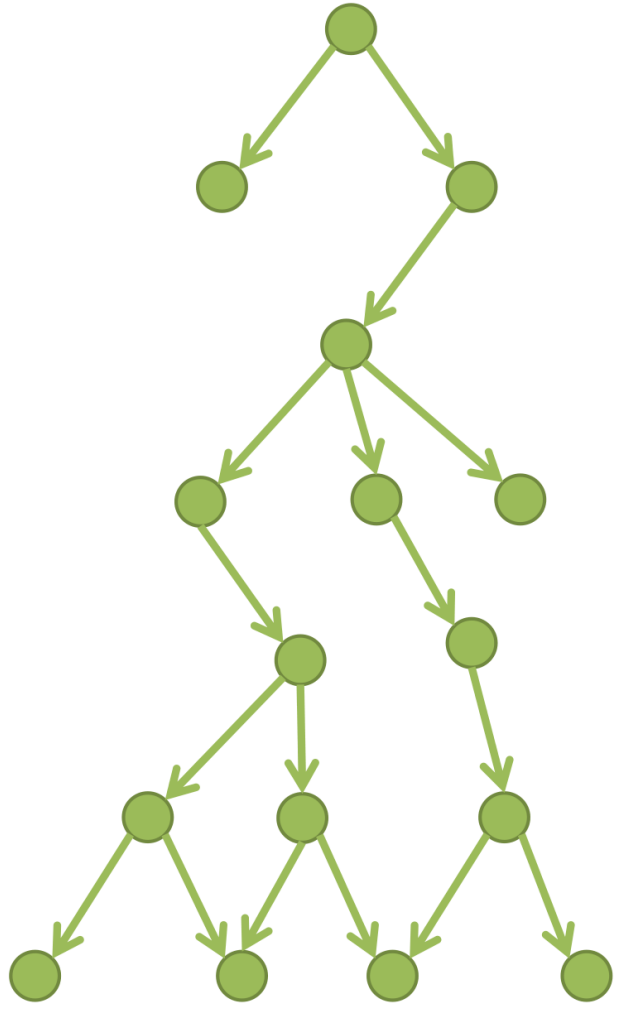
Operational



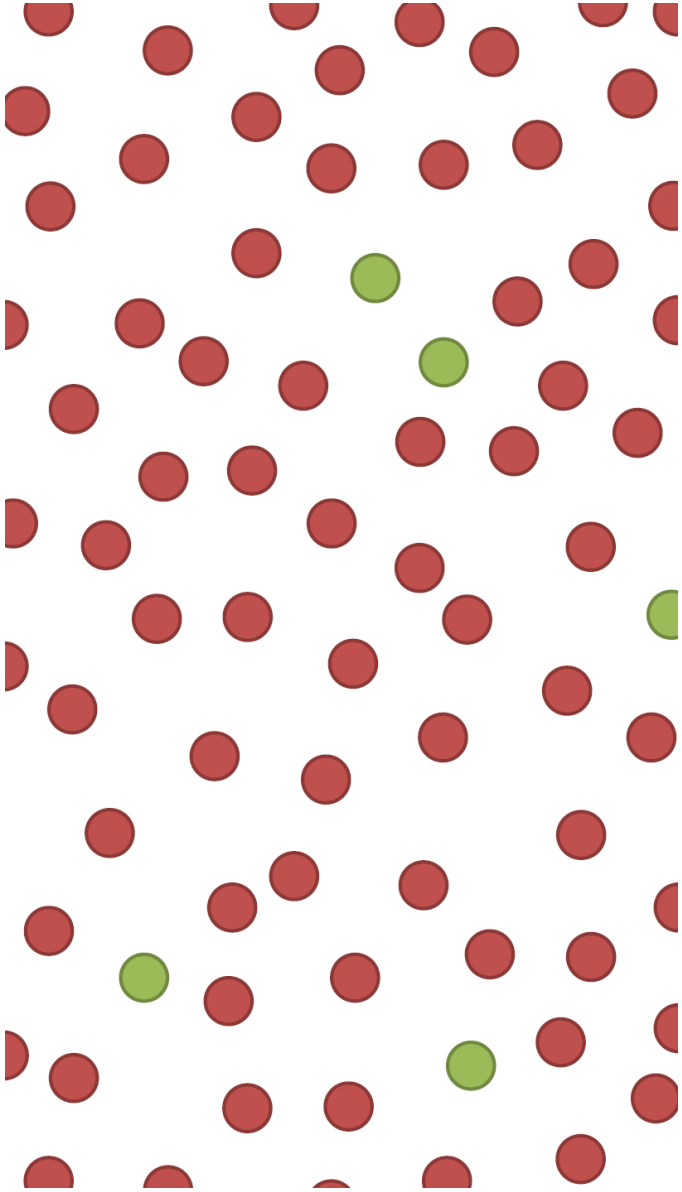
Axiomatic



Operational



Axiomatic



Operational