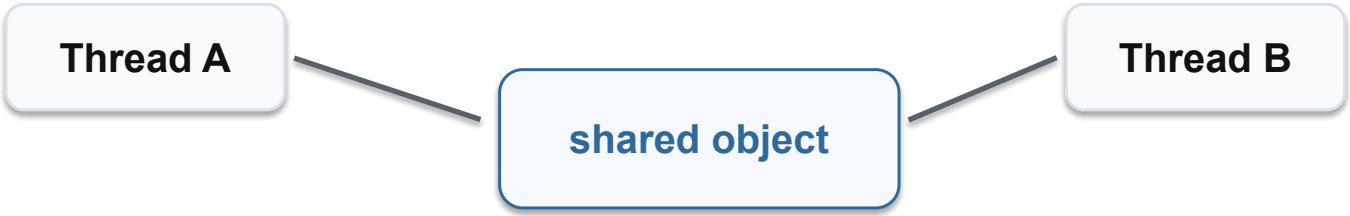
Pyrona

Shared background

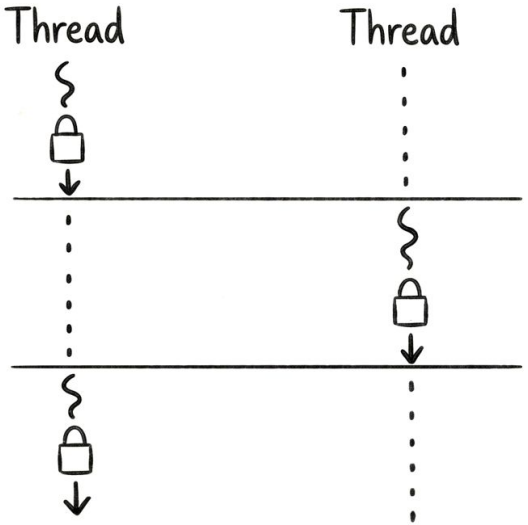
Python and Threads

Python makes threads easy to create, but shared mutable data still needs protection.



Problem: If both threads want to mutate the same object, the runtime must know when that is safe.

Global Interpreter Lock



Interpreters and Sub-Interpreters

Moving toward parallelism inside CPython

- An interpreter is one Python execution environment
- A sub-interpreter is another environment in the same process
- Each sub-interpreter can have its own GIL
- Parallel bytecode execution becomes possible

Parallelism yes, but safe sharing still needs rules

Pyrona

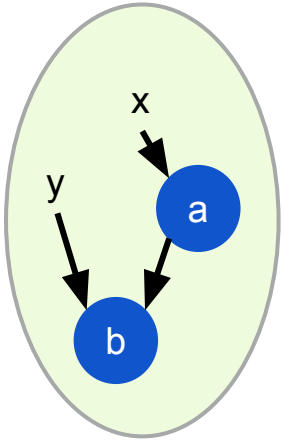
Shared Memory with a Thread-Safe Runtime

- Programmer follows ownership rules for thread safety
- Valid ownership use ⇒ shared memory with parallelism
- Invalid ownership use ⇒ runtime exception

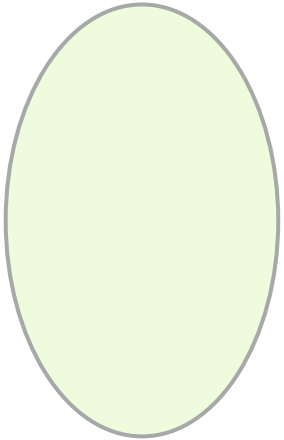
Achieves: Parallelism without data races

Immutability

Sharing Between Sub-interpreters

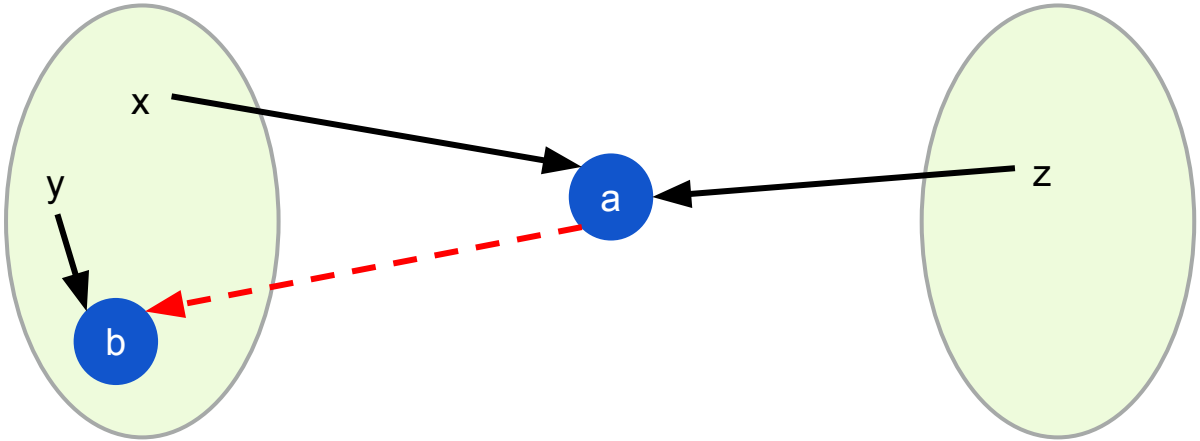


Sub-interpreter 1

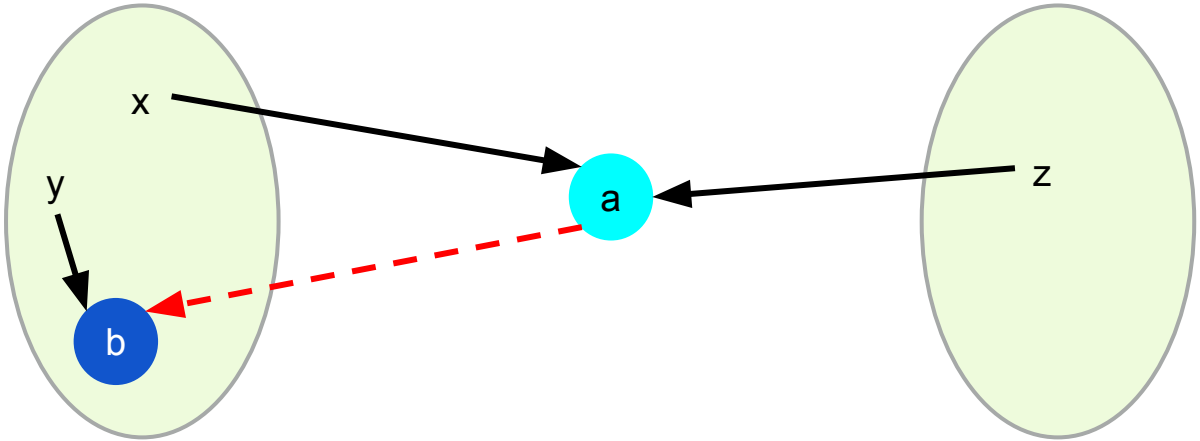


Sub-interpreter 2

Sharing Between Sub-interpreters

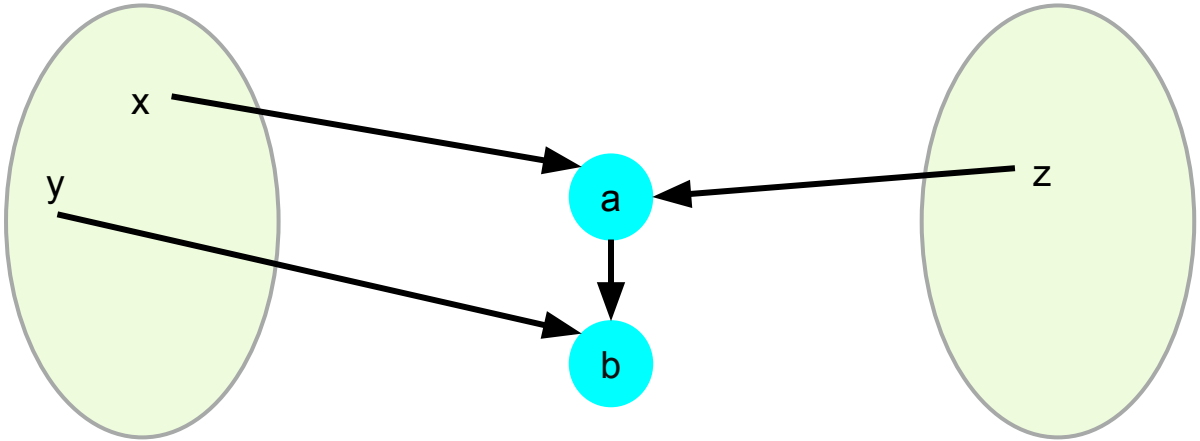


Sharing Between Sub-interpreters: Shallow Immutability



`freeze(a)`

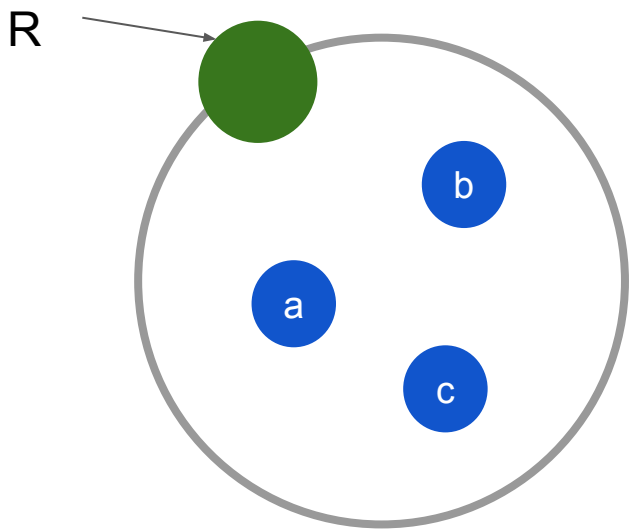
Sharing Between Sub-interpreters: Deep Immutability



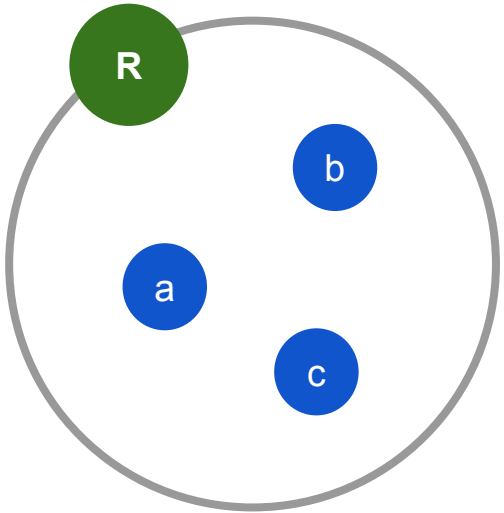

```
freeze(a)
```

Regions

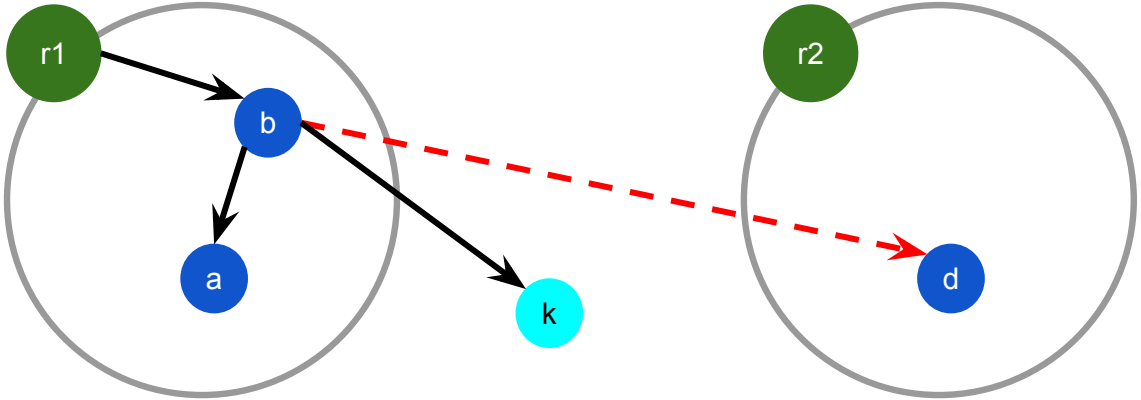
Region



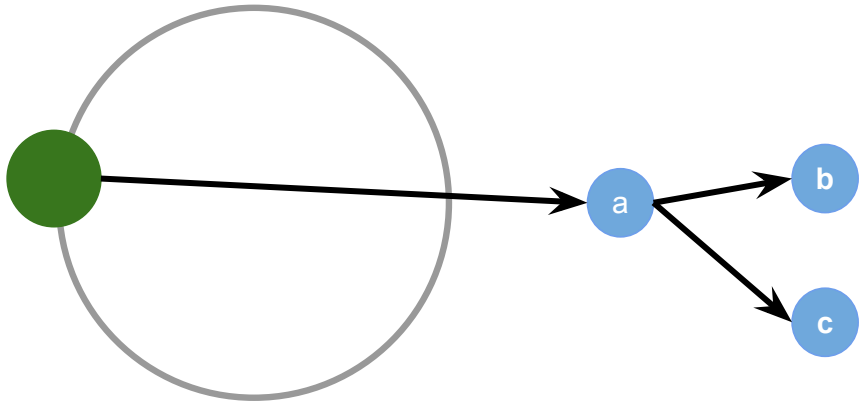
Region



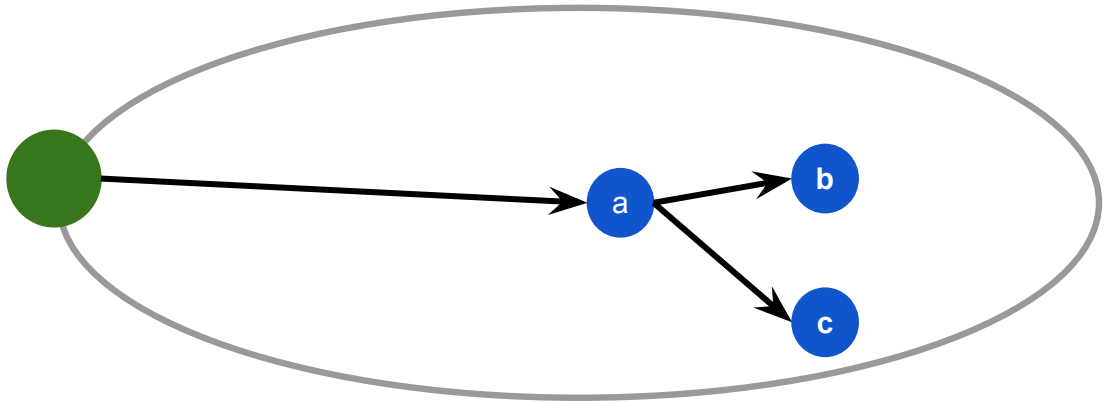
Region’s Rules (Constraints)



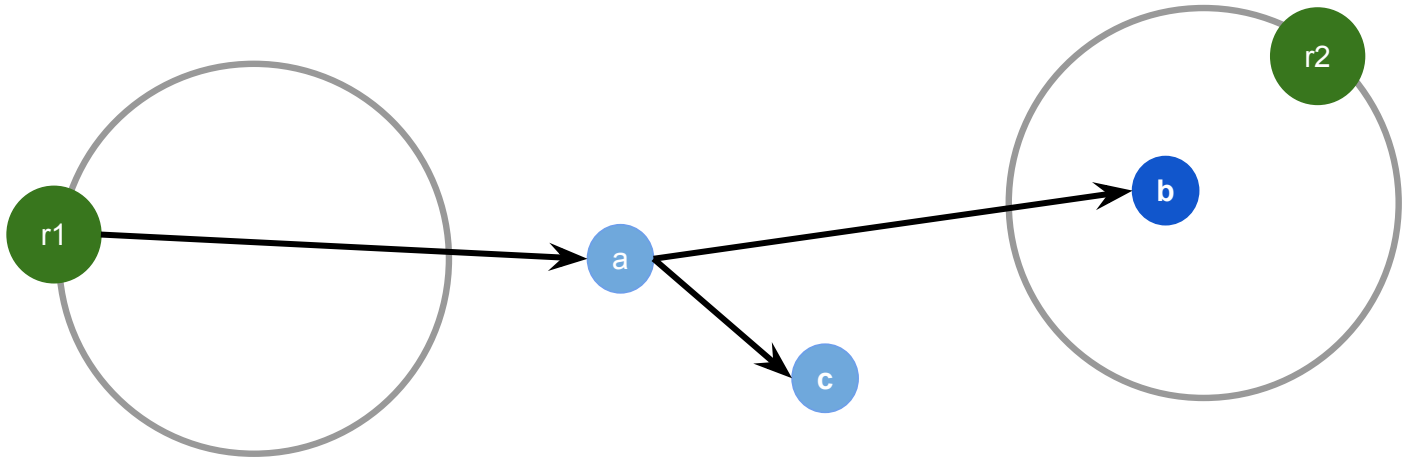
Region’s Rules (Constraints) - Ephemeral



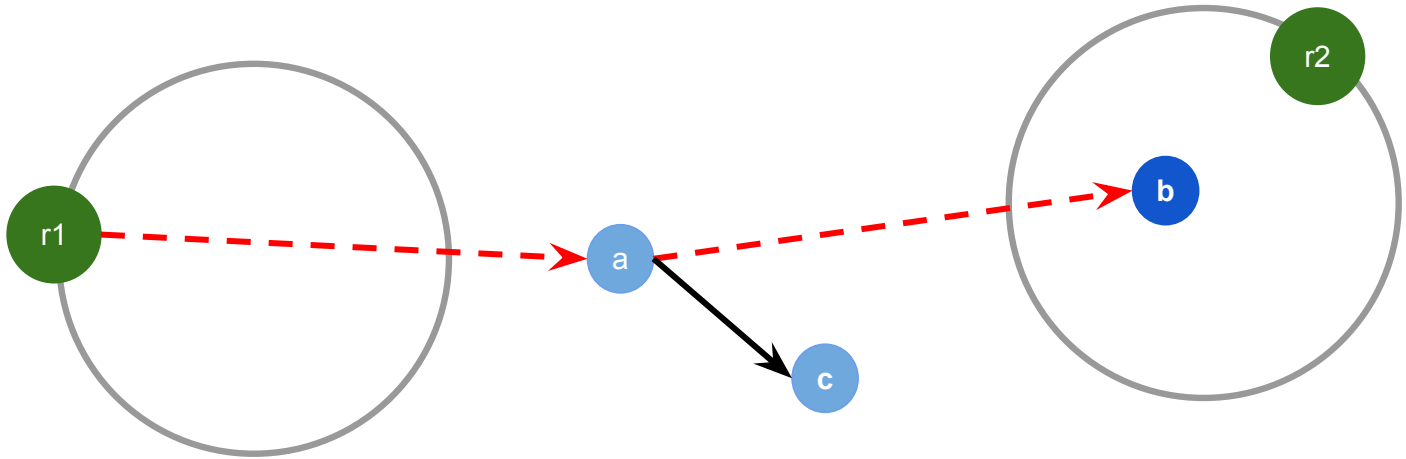
Region’s Rules (Constraints) - Ephemeral



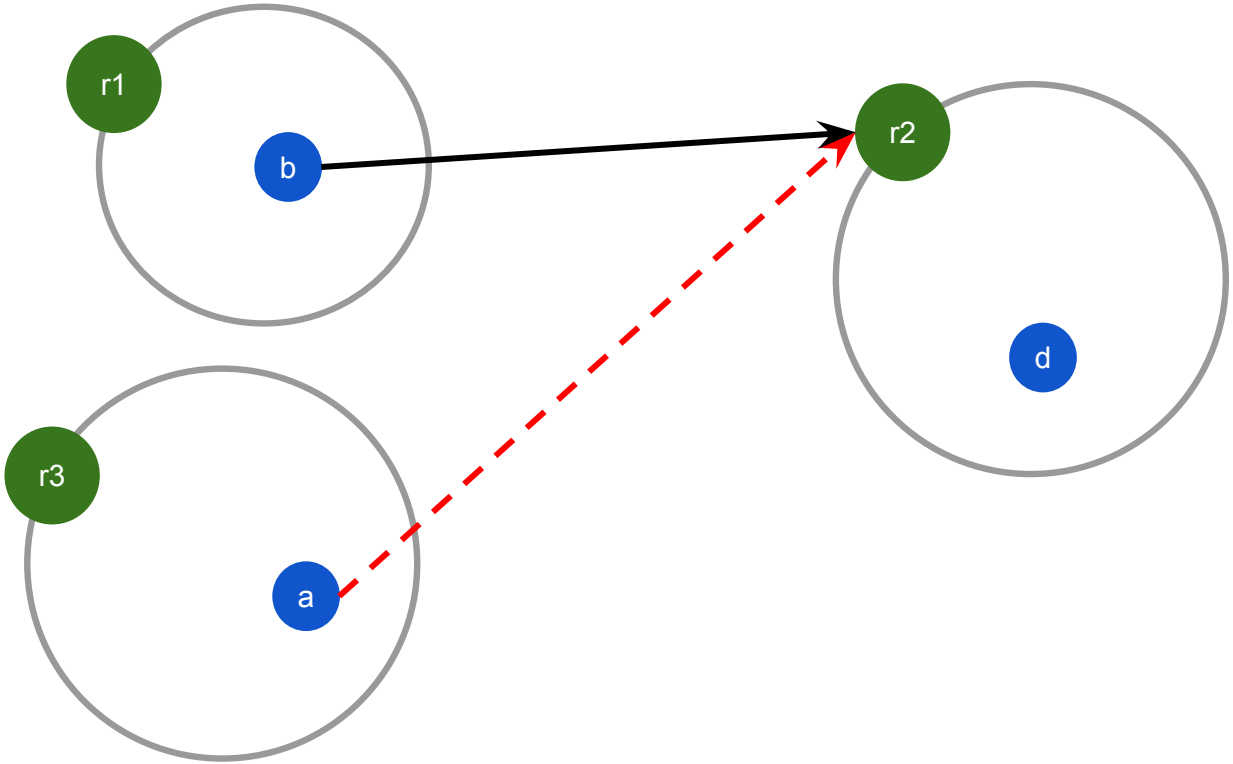
Region’s Rules (Constraints) - Ephemeral



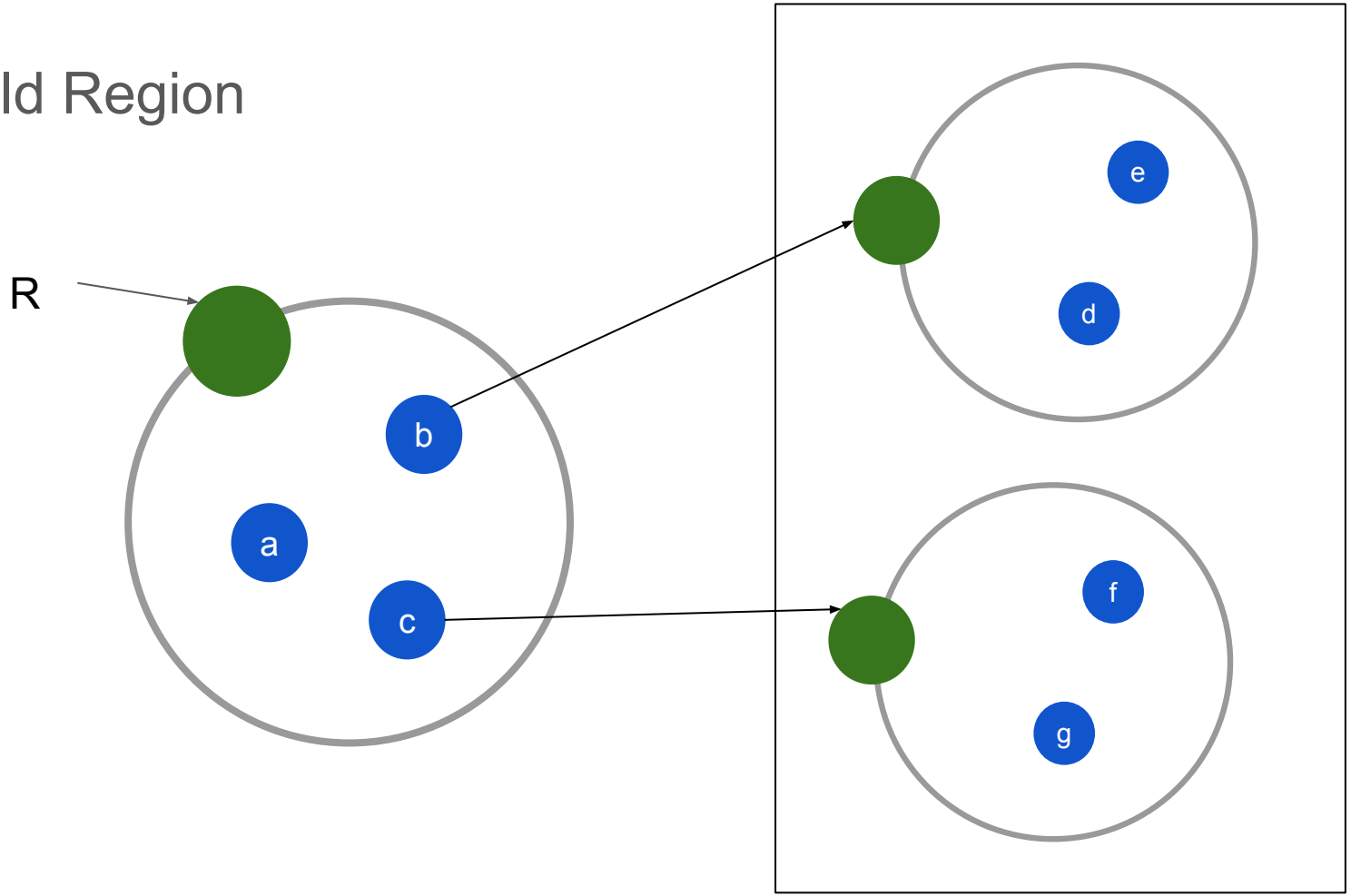
Region’s Rules (Constraints) - Ephemeral



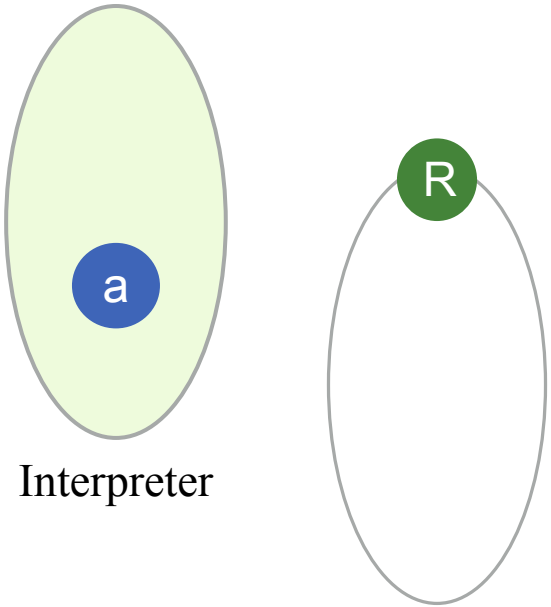
Region’s Rules (Constraints) - External Uniqueness



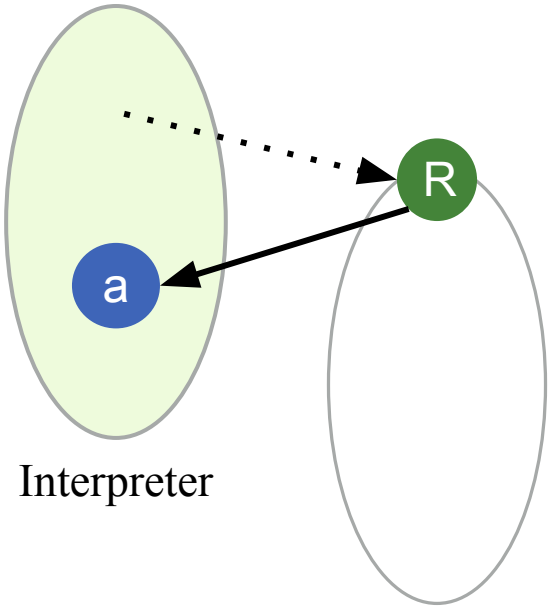
Child Region



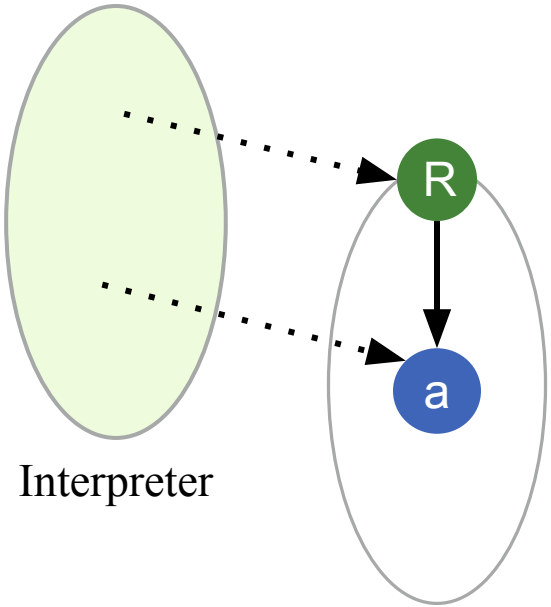
Local Region



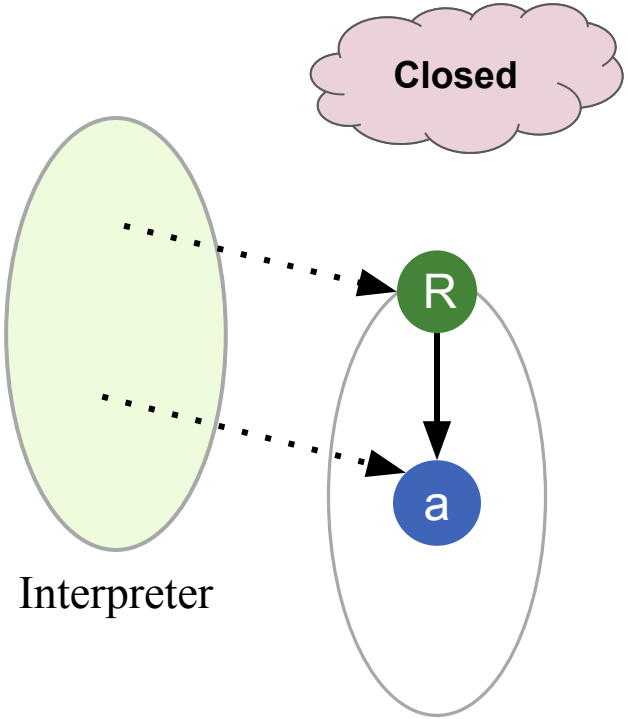
Borrowed references



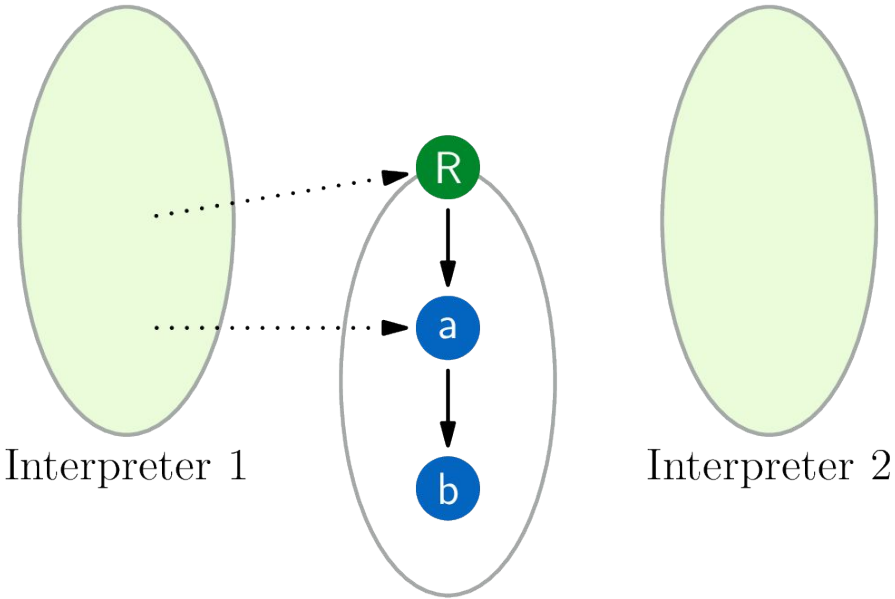
Borrowed references



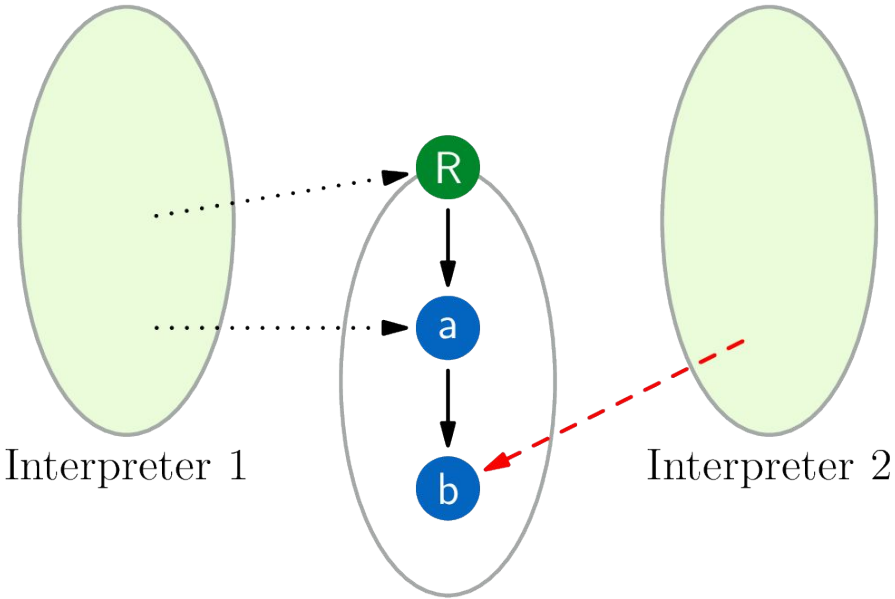
Open and closed regions



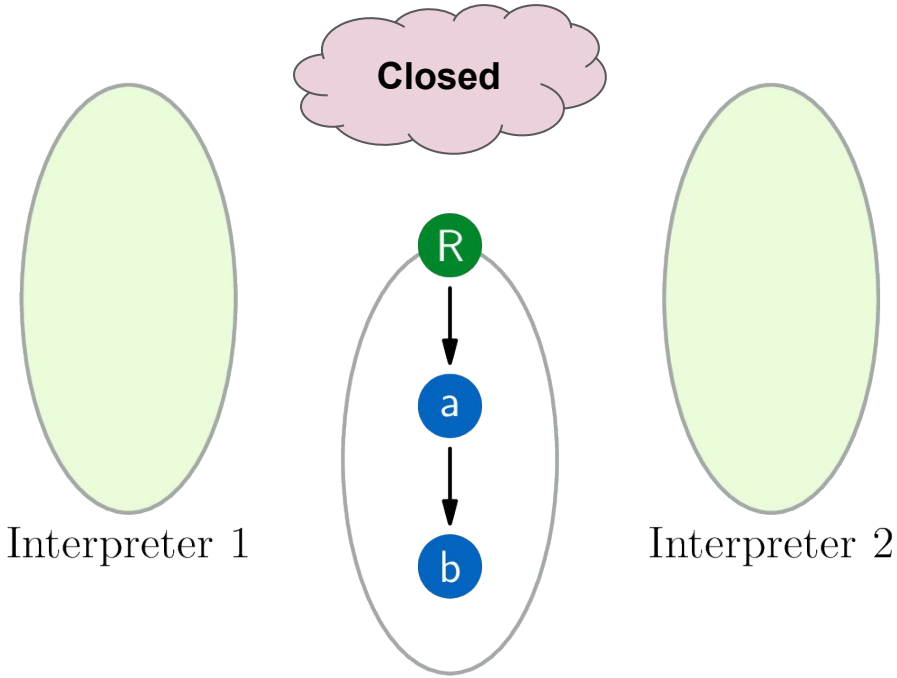
Region transfer



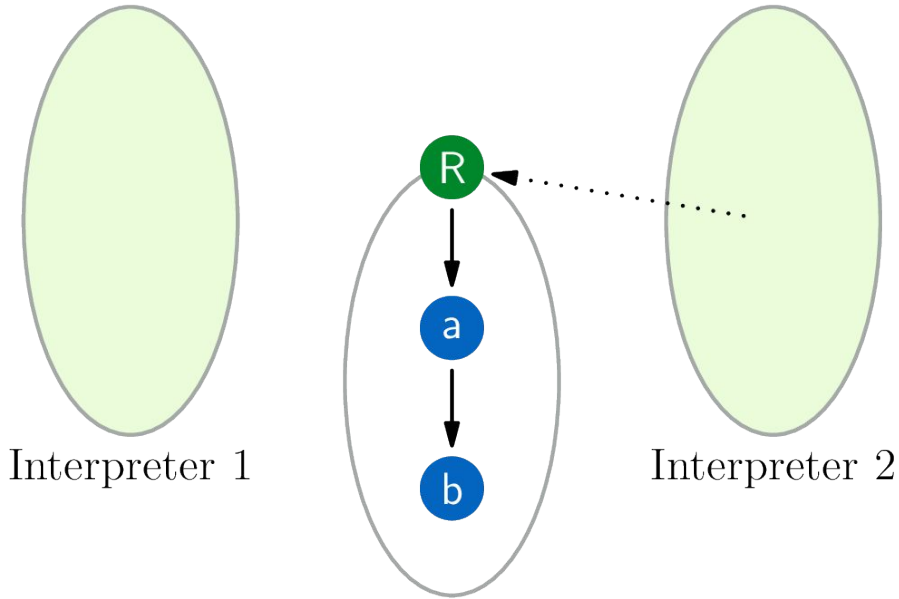
Region transfer



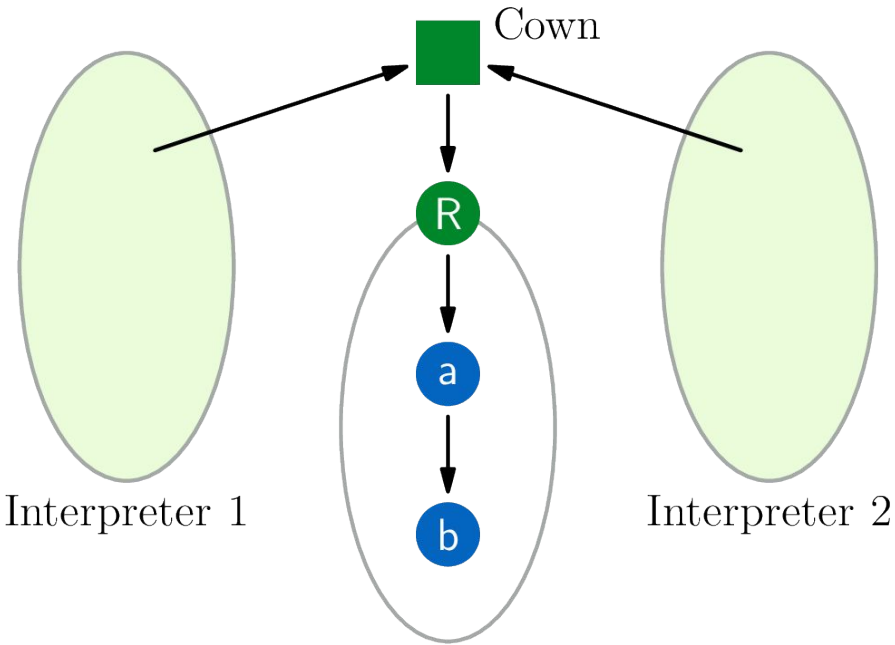
Region transfer



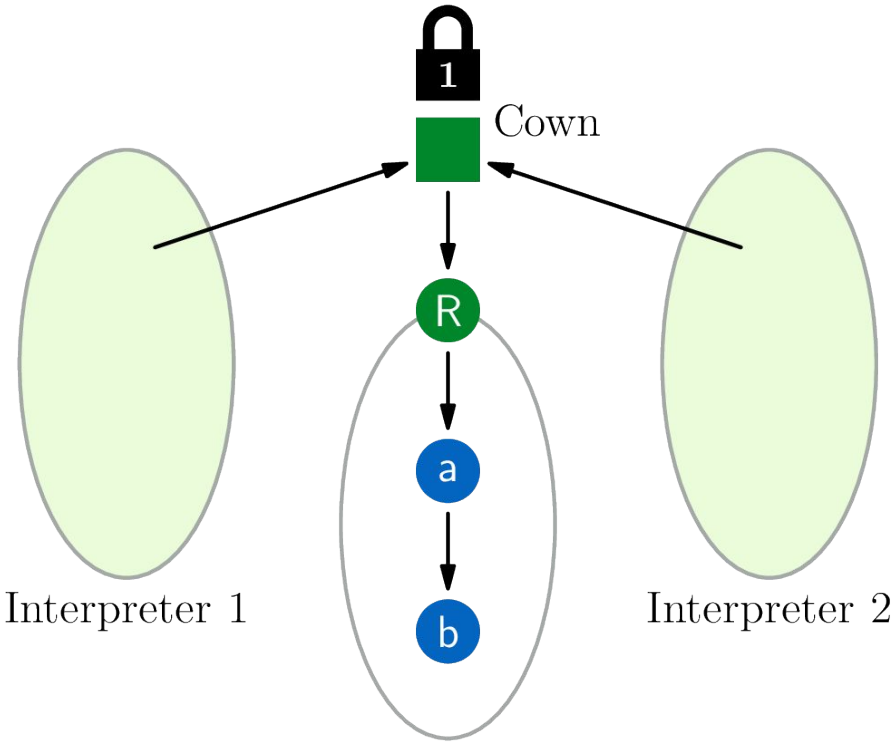
Region transfer



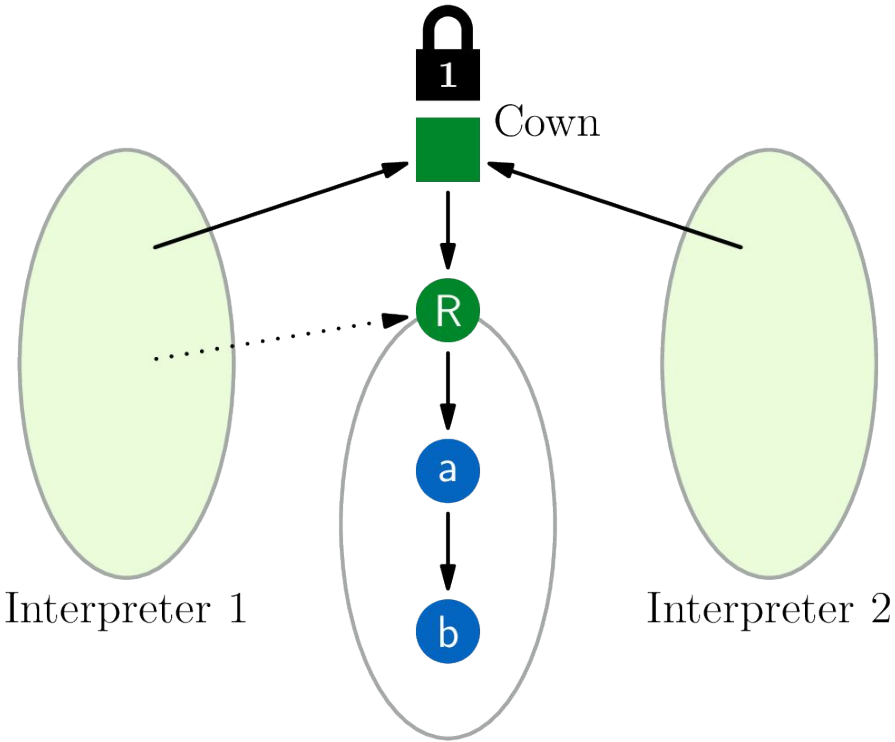
Cown



Cown



Cown



Summary

- Immutable objects
  - Freely shareable
- Regions for mutable objects
  - Shareable via cowns
- Familiar parallelism
- No data races