Modal Crash Types for Intermittent Computing

Farzaneh Derakhshan
Postdoctoral fellow, CSD, CMU

Joint work with Myra Dotzel, Milijana Surbatovich, and Limin Jia

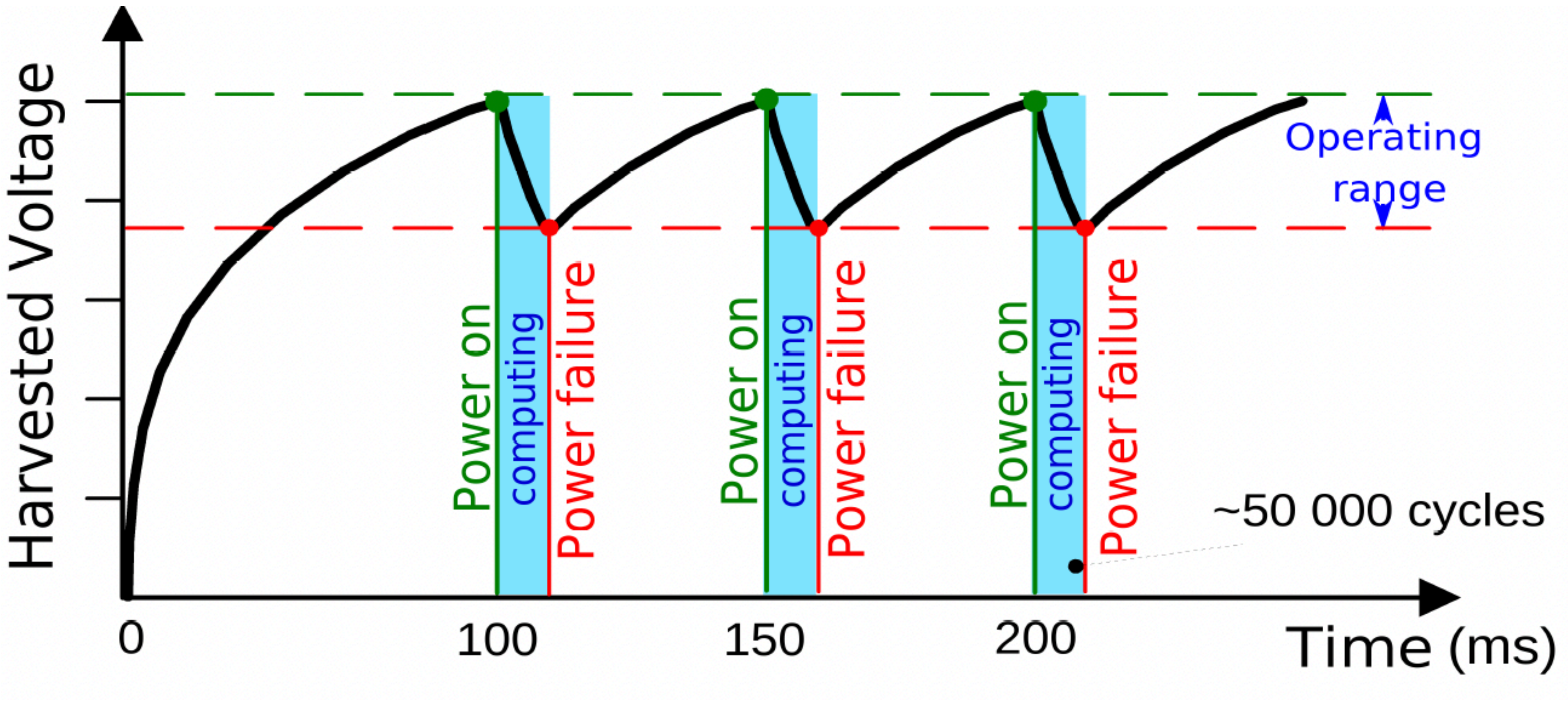
ESOP 2023

Intermittent execution in energy harvesting devices

Devices powered with energy from the environment (e.g., solar)

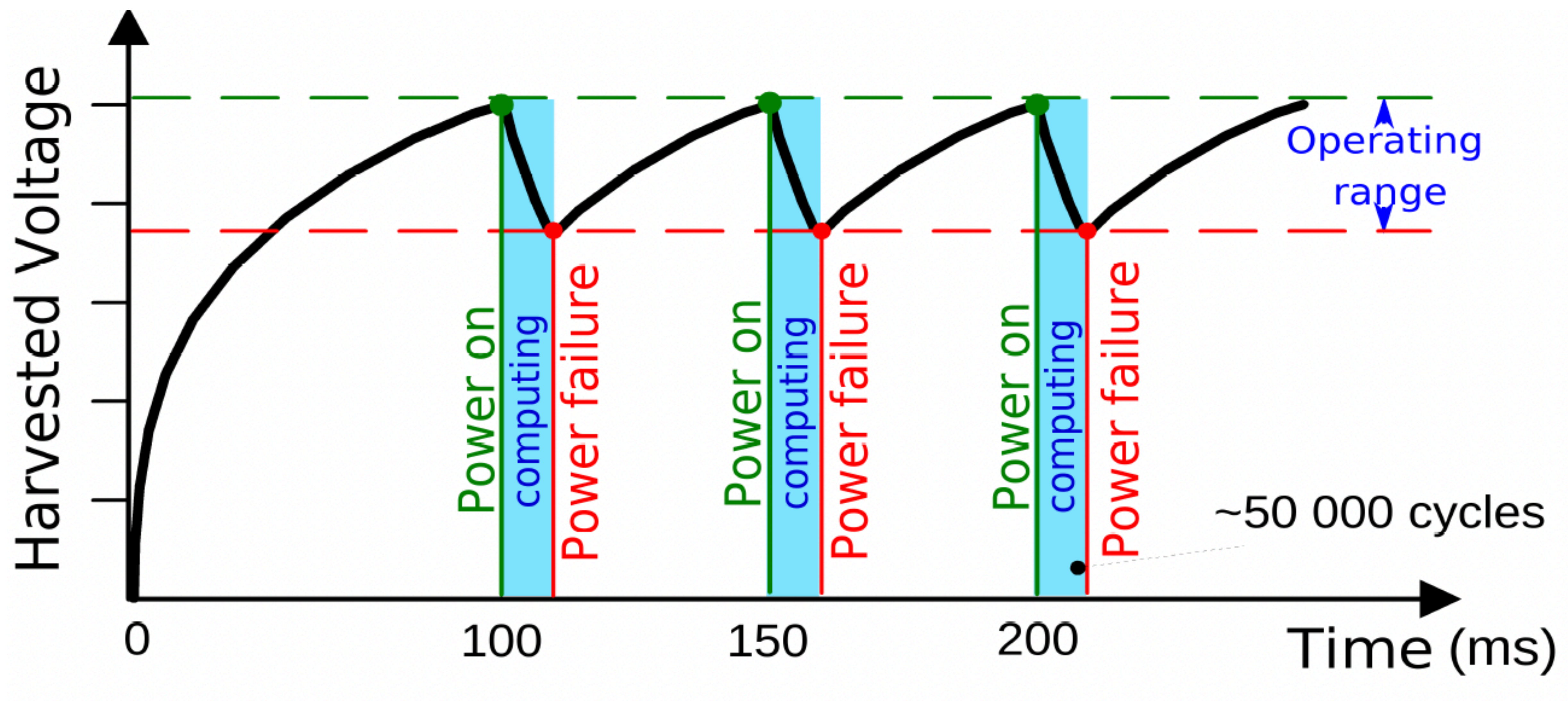
Intermittent execution in energy harvesting devices

Devices powered with energy from the environment (e.g., solar)



Intermittent execution in energy harvesting devices

Devices powered with energy from the environment (e.g., solar)

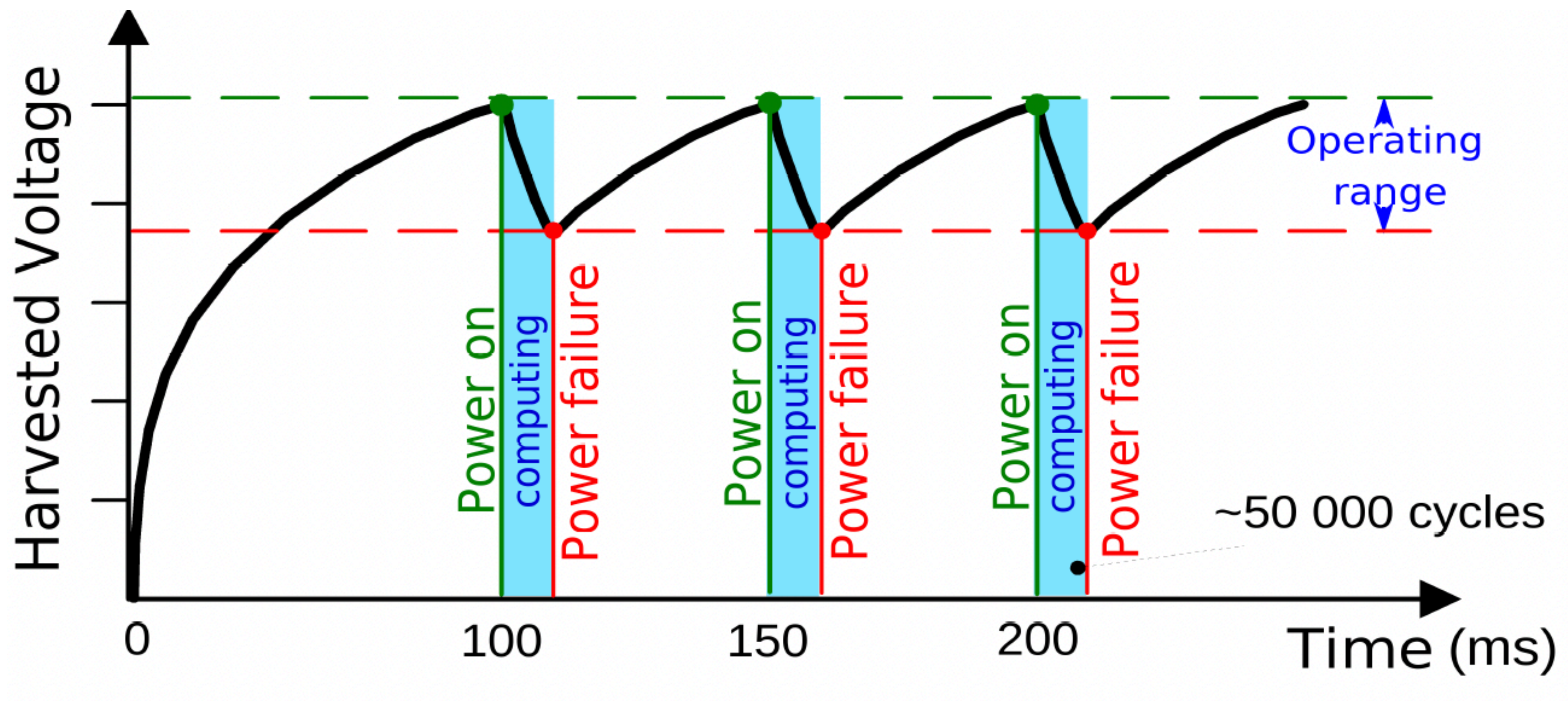


Examples

- Harsh environments
  - Space, medical devices
- Tiny devices w/o battery

Intermittent execution in energy harvesting devices

Devices powered with energy from the environment (e.g., solar)



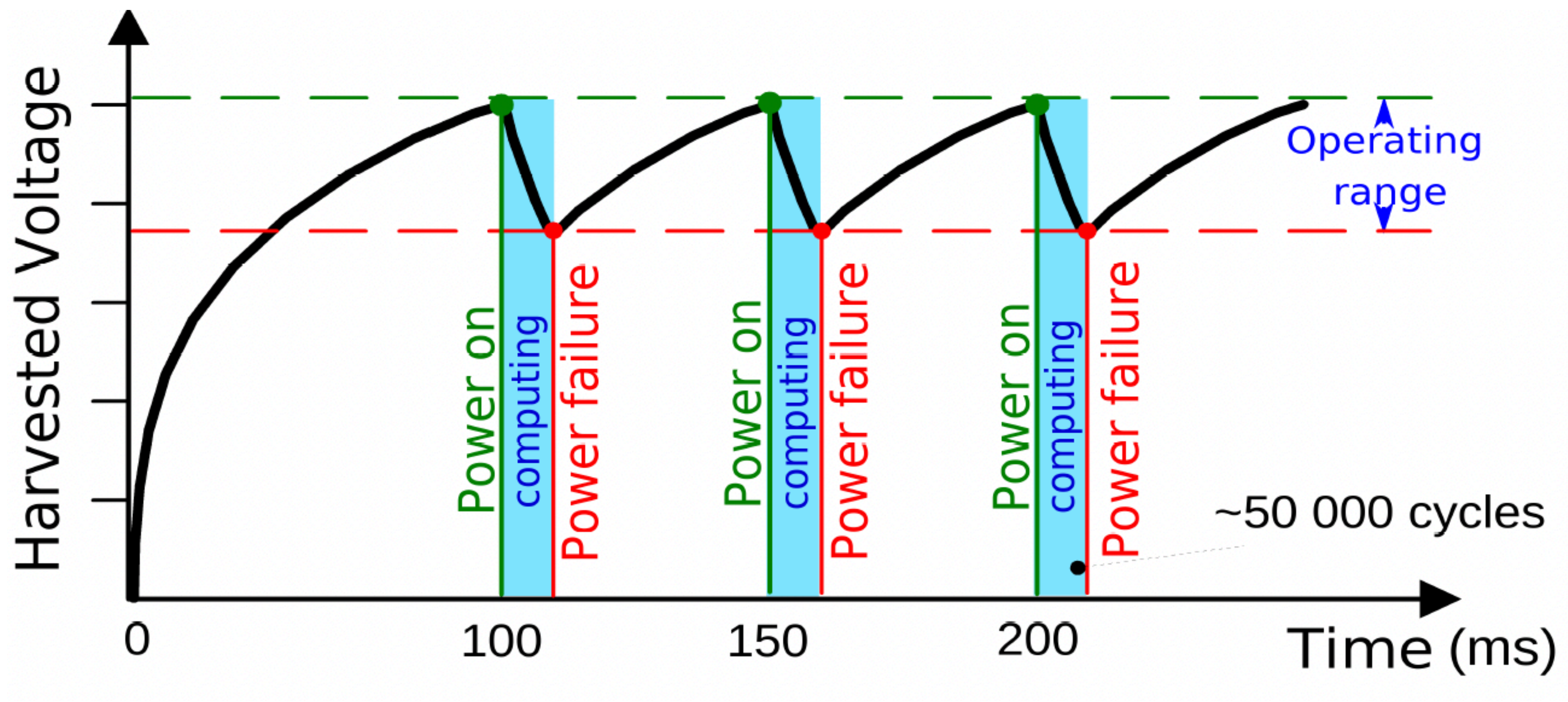
Examples

- Harsh environments
  - Space, medical devices
- Tiny devices w/o battery

Energy is available intermittently → Executions are intermittent

Intermittent execution in energy harvesting devices

Devices powered with energy from the environment (e.g., solar)



Examples

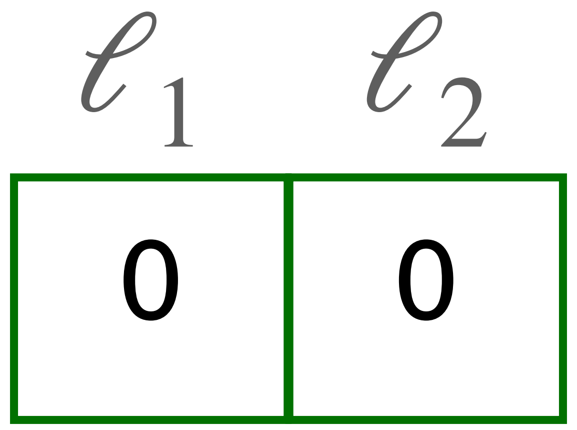
- Harsh environments
  - Space, medical devices
- Tiny devices w/o battery

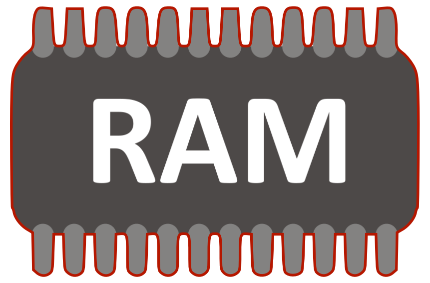
Energy is available intermittently → Executions are intermittent

We need to handle frequent power failures!

Volatile and nonvolatile memories with stable and unstable types

Nonvolatile memory

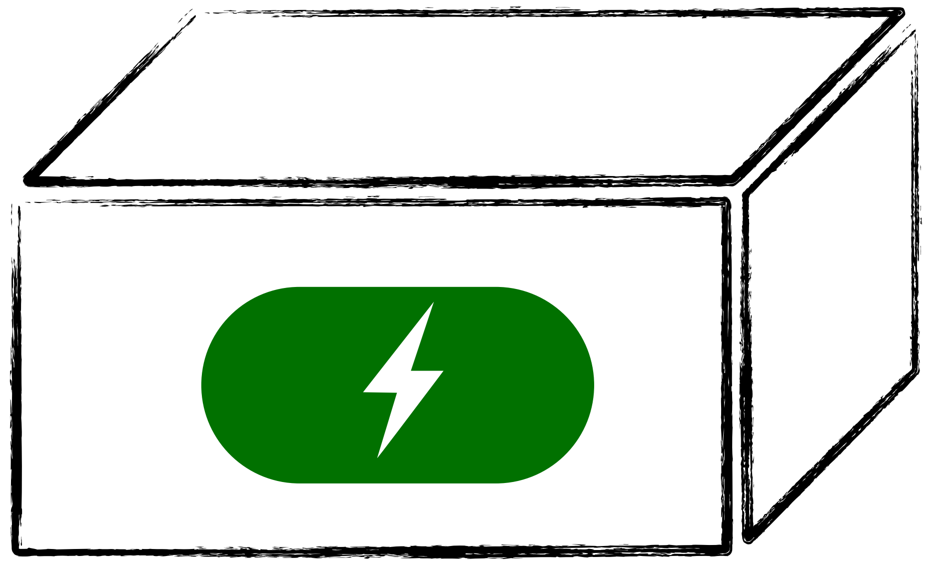




Volatile memory

pc →

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```



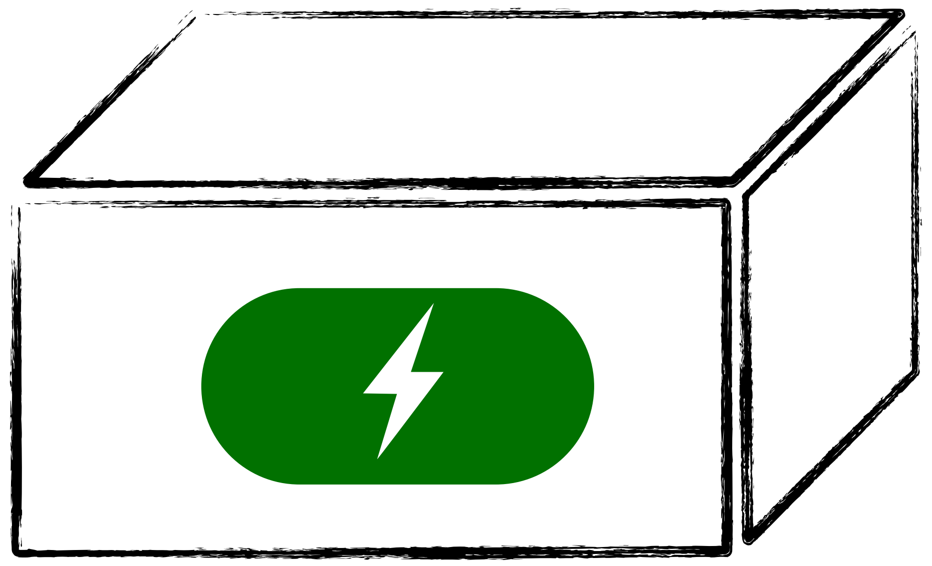
Volatile and nonvolatile memories with stable and unstable types

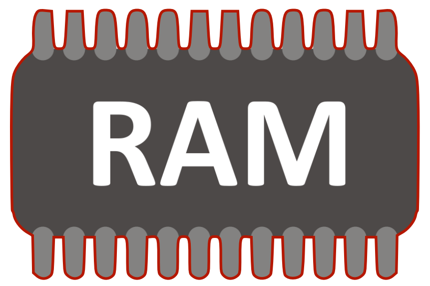
pc →

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```

Nonvolatile memory

ℓ_1	ℓ_2
0	0





Volatile memory

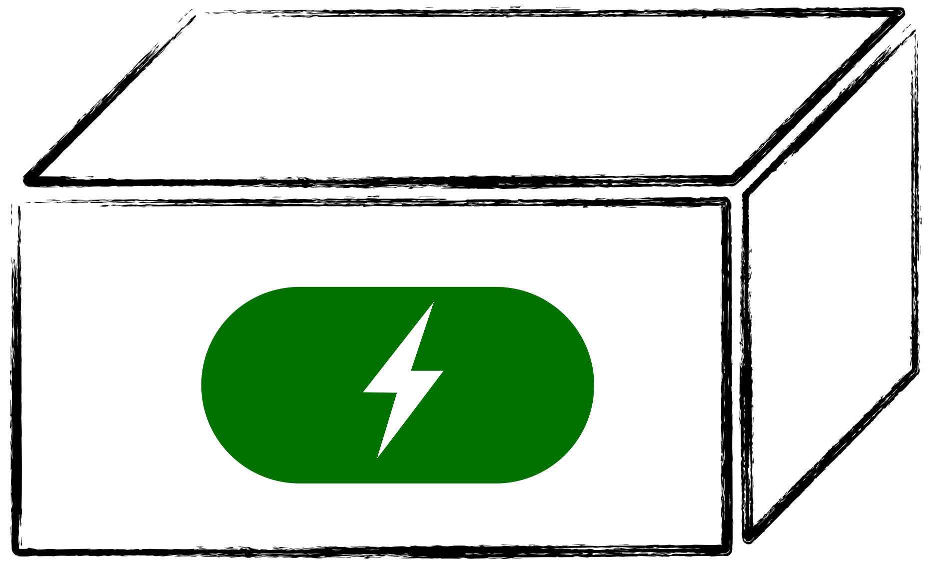
Final memory state we expect:

ℓ_1	ℓ_2
2	2

Volatile and nonvolatile memories with stable and unstable types

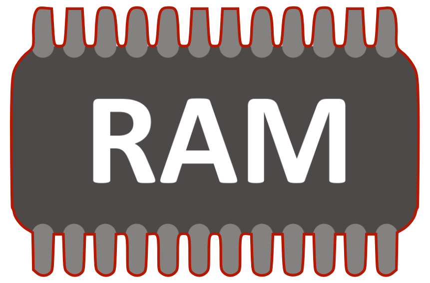
pc →

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```



Nonvolatile memory

ℓ_1	ℓ_2
0	0



Volatile memory

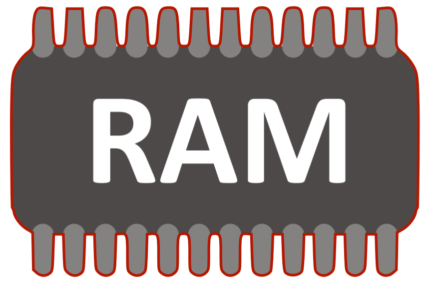
Final memory state we expect:

ℓ_1	ℓ_2
2	2

Volatile and nonvolatile memories with stable and unstable types

Nonvolatile memory

ℓ_1	ℓ_2
0	0

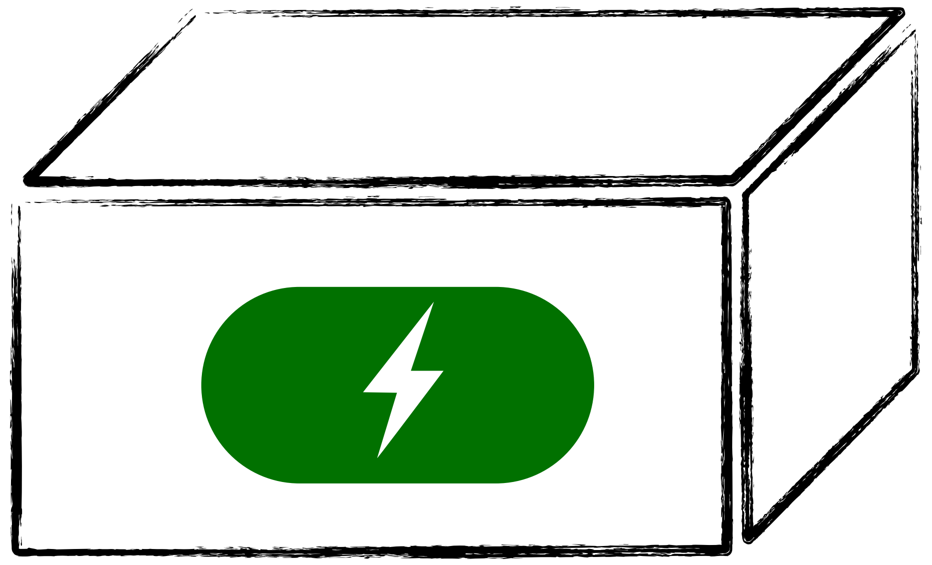


Volatile memory

w
2

pc →

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```



Final memory state we expect:

ℓ_1	ℓ_2
2	2

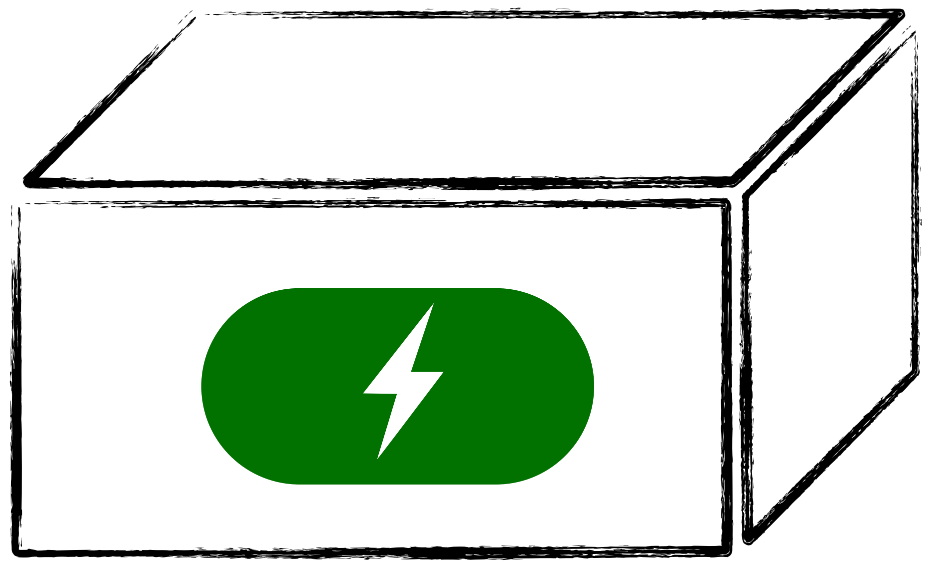
Volatile and nonvolatile memories with stable and unstable types

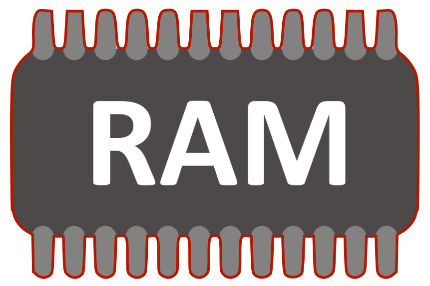
Nonvolatile memory

ℓ_1	ℓ_2
0	0

pc →

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```





Volatile memory

w
2

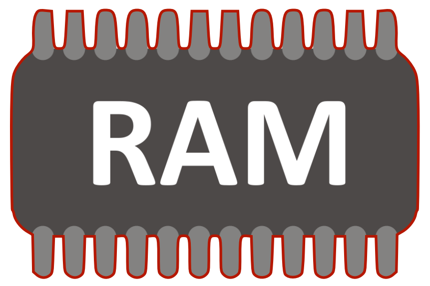
Final memory state we expect:

ℓ_1	ℓ_2
2	2

Volatile and nonvolatile memories with stable and unstable types

Nonvolatile memory

ℓ_1	ℓ_2
2	0

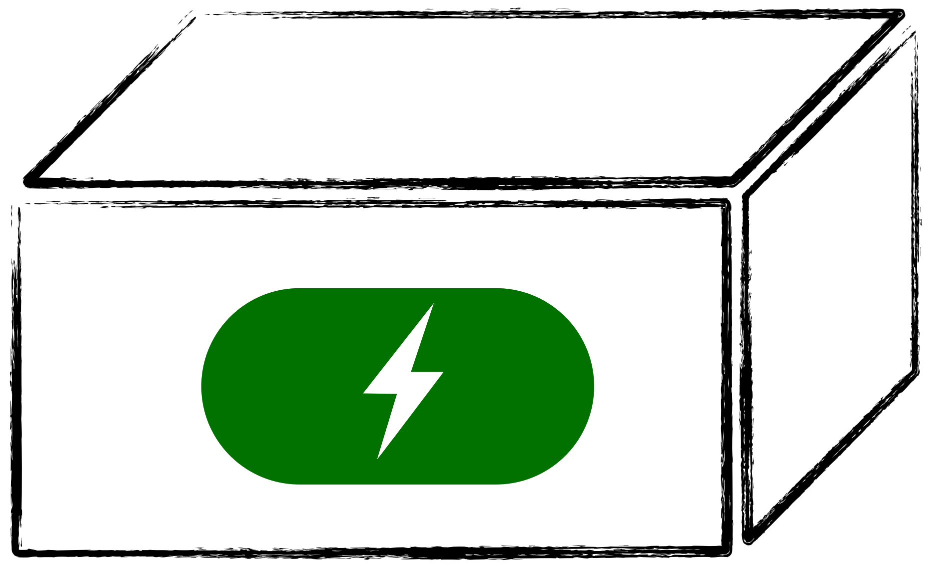


Volatile memory

w
2

pc →

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```



Final memory state we expect:

ℓ_1	ℓ_2
2	2

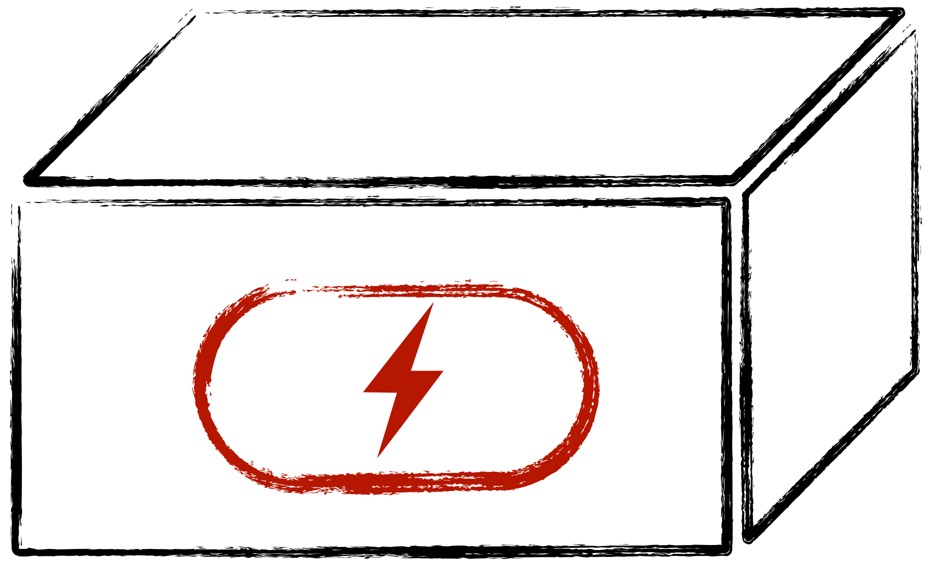
Volatile and nonvolatile memories with stable and unstable types

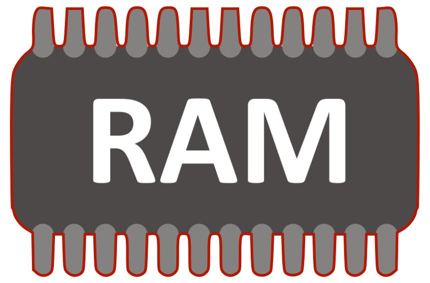
Nonvolatile memory

ℓ_1	ℓ_2
2	0

pc →

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```





Volatile memory

w
2

Final memory state we expect:

ℓ_1	ℓ_2
2	2

Volatile and nonvolatile memories with stable and unstable types

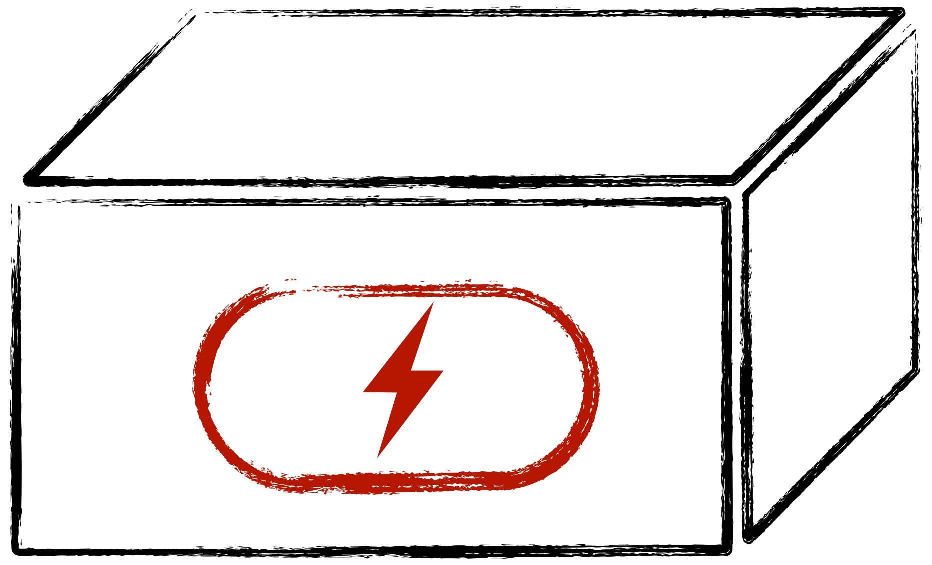
Nonvolatile memory

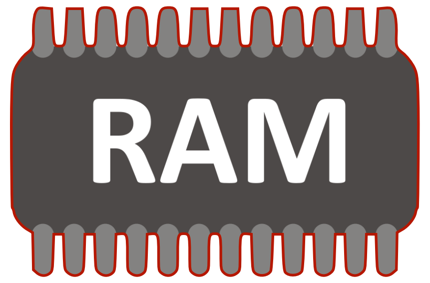
Stable values

ℓ_1	ℓ_2
2	0

pc →

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```





Volatile memory

w
2

Final memory state we expect:

ℓ_1	ℓ_2
2	2

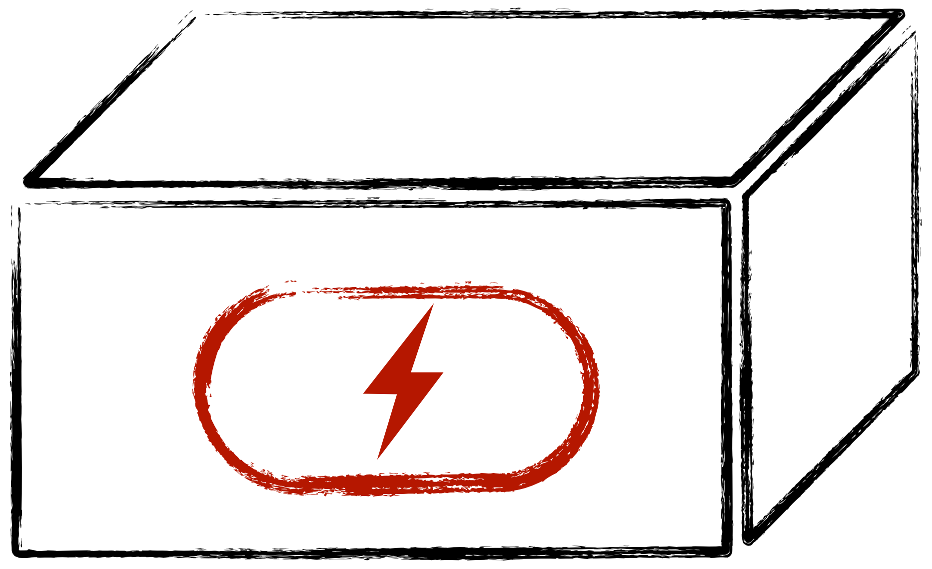
Volatile and nonvolatile memories with stable and unstable types

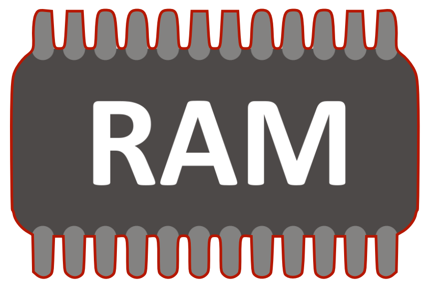
Nonvolatile memory
Stable values

ℓ_1	ℓ_2
2	0

pc →

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```





Volatile memory
Unstable values

Final memory state we expect:

ℓ_1	ℓ_2
2	2

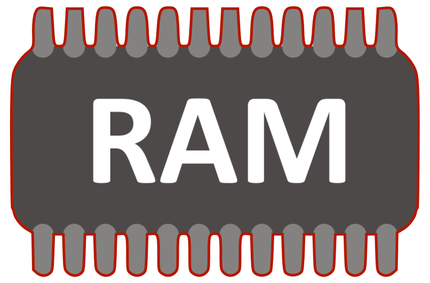
Volatile and nonvolatile memories with stable and unstable types

pc →

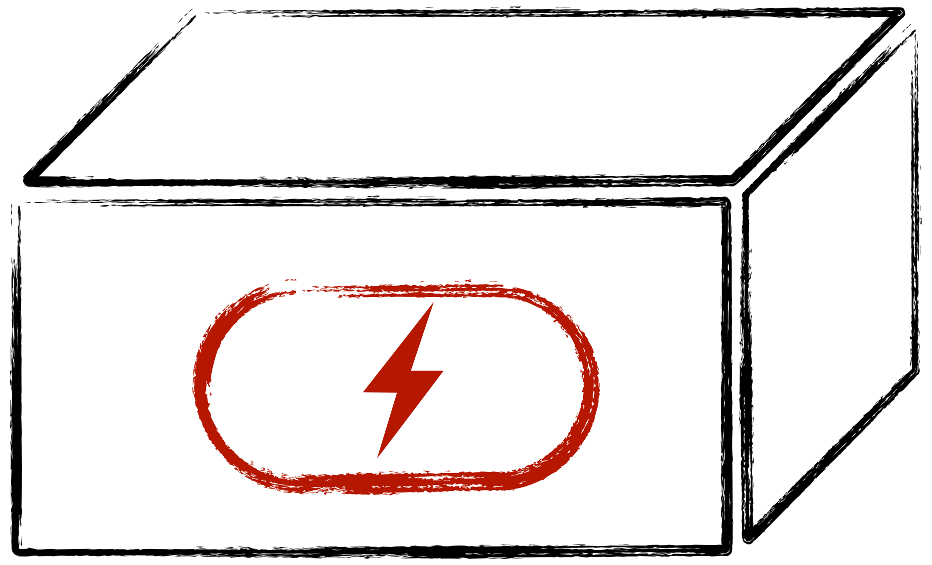
```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```

ℓ_1	ℓ_2
2	0

Nonvolatile memory
Stable values



Volatile memory
Unstable values



Final memory state we expect:

ℓ_1	ℓ_2
2	2

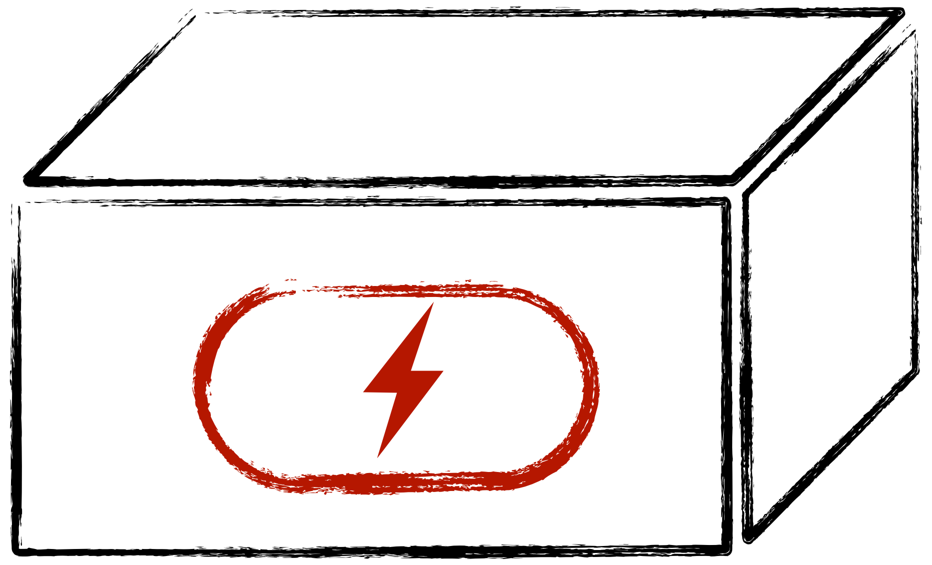
Volatile and nonvolatile memories with stable and unstable types

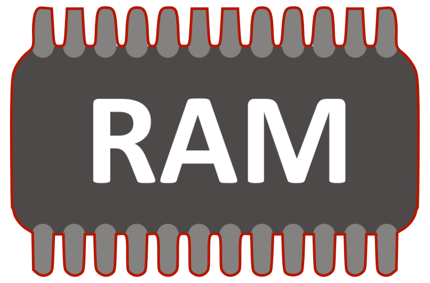
pc →

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```

ℓ_1	ℓ_2
2	0

Nonvolatile memory
Stable values





Volatile memory
Unstable values

Final memory state we expect:

ℓ_1	ℓ_2
2	2

Final memory state we get:

ℓ_1	ℓ_2
4	4

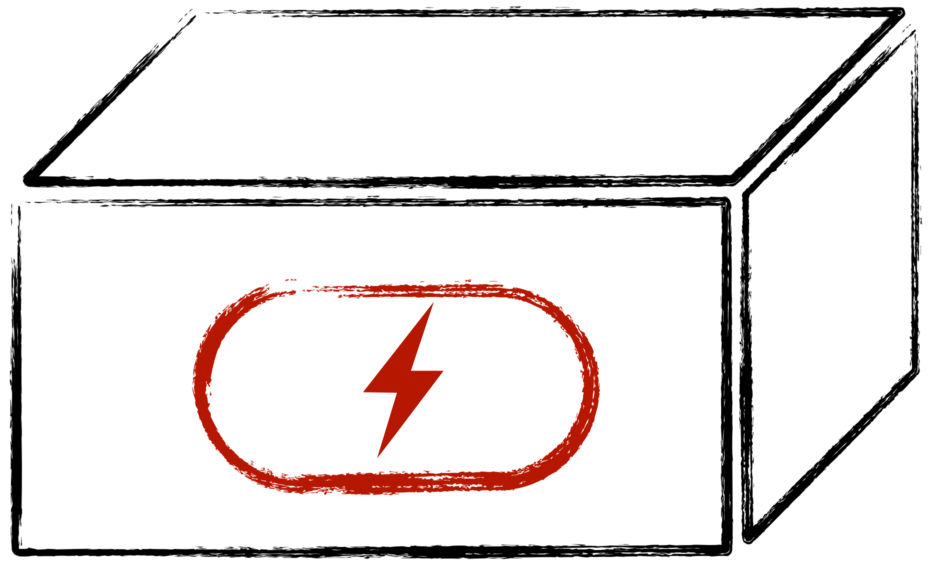
Volatile and nonvolatile memories with stable and unstable types

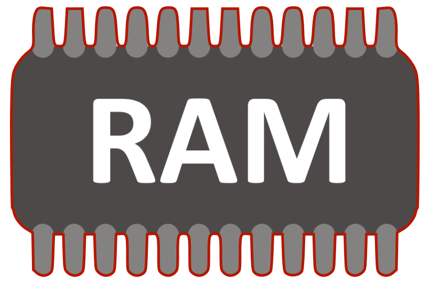
Nonvolatile memory
Stable values

ℓ_1	ℓ_2
2	0

pc →

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```





Volatile memory
Unstable values

Final memory state we expect:

ℓ_1	ℓ_2
2	2

Incorrect

Final memory state we get:

ℓ_1	ℓ_2
4	4

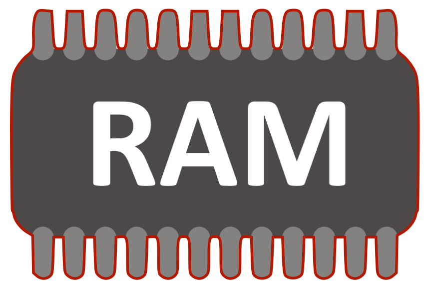
Solution: Checkpointing blocks (checkpoint-restore-finalize)

pc → Checkpoint

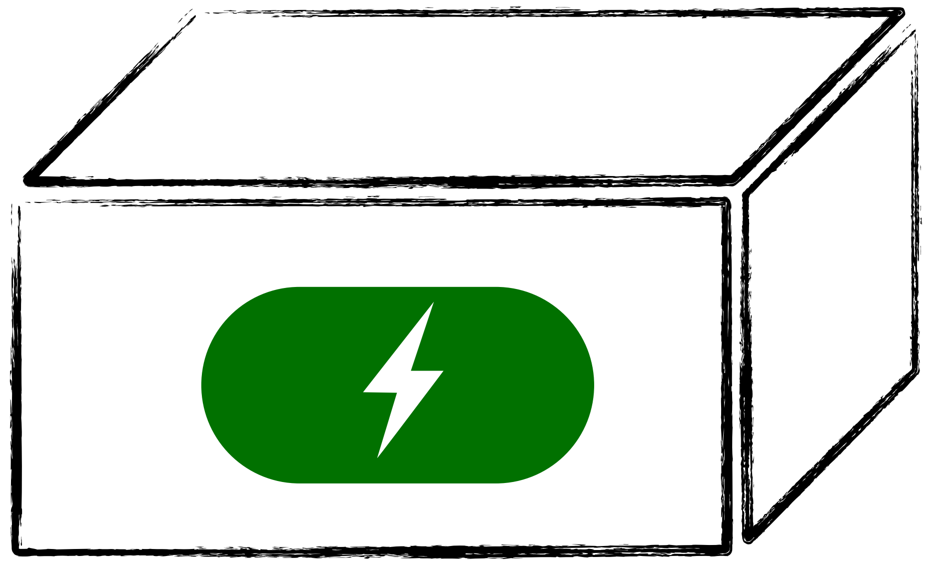
```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```

Nonvolatile memory
Stable values

ℓ_1	ℓ_2
0	0



Volatile memory
Unstable values



Final memory state we expect:

ℓ_1	ℓ_2
2	2

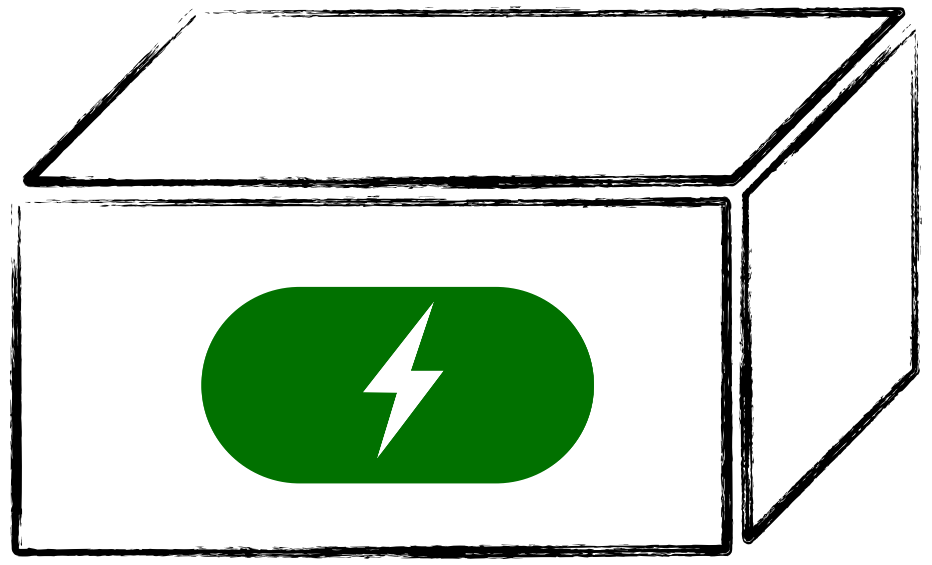
Solution: Checkpointing blocks (checkpoint-restore-finalize)

Checkpoint

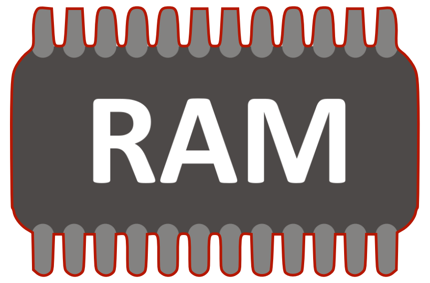
ℓ_1	ℓ_2
0	0

pc → Checkpoint

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```



Nonvolatile memory
Stable values



Volatile memory
Unstable values

Final memory state we expect:

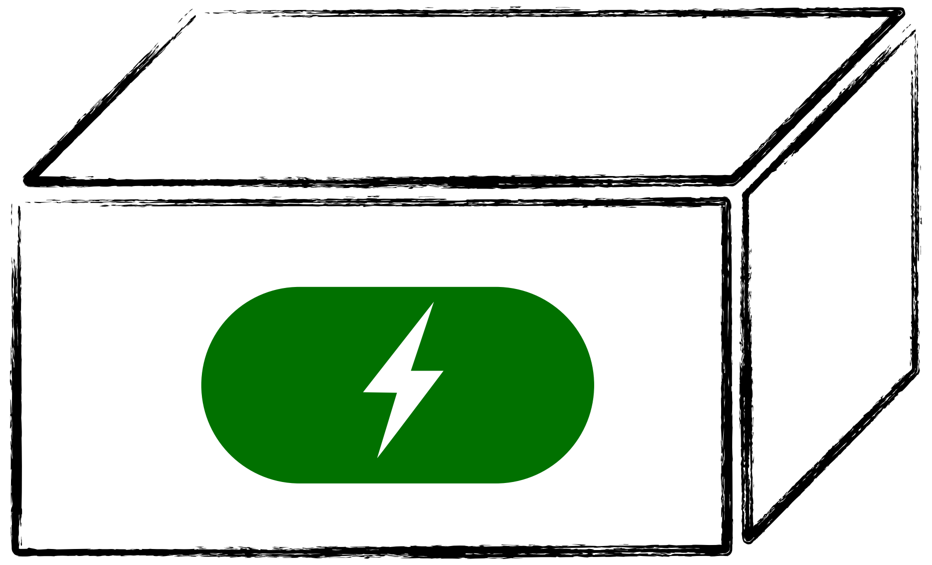
ℓ_1	ℓ_2
2	2

Solution: Checkpointing blocks (checkpoint-restore-finalize)

Checkpoint

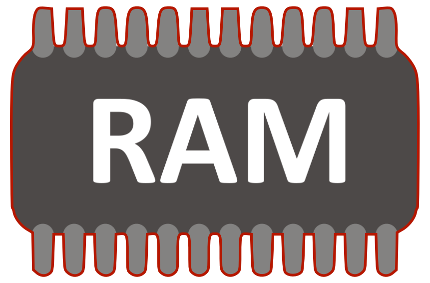
pc → Checkpoint

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```



Nonvolatile memory
Stable values

ℓ_1	ℓ_2
0	0



Volatile memory
Unstable values

ℓ_1^s
0

Final memory state we expect:

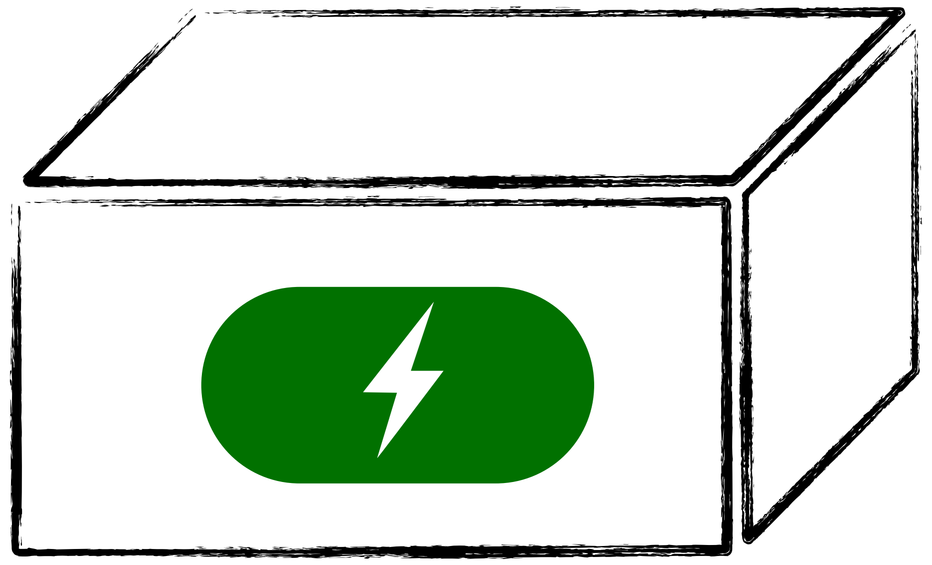
ℓ_1	ℓ_2
2	2

Solution: Checkpointing blocks (checkpoint-restore-finalize)

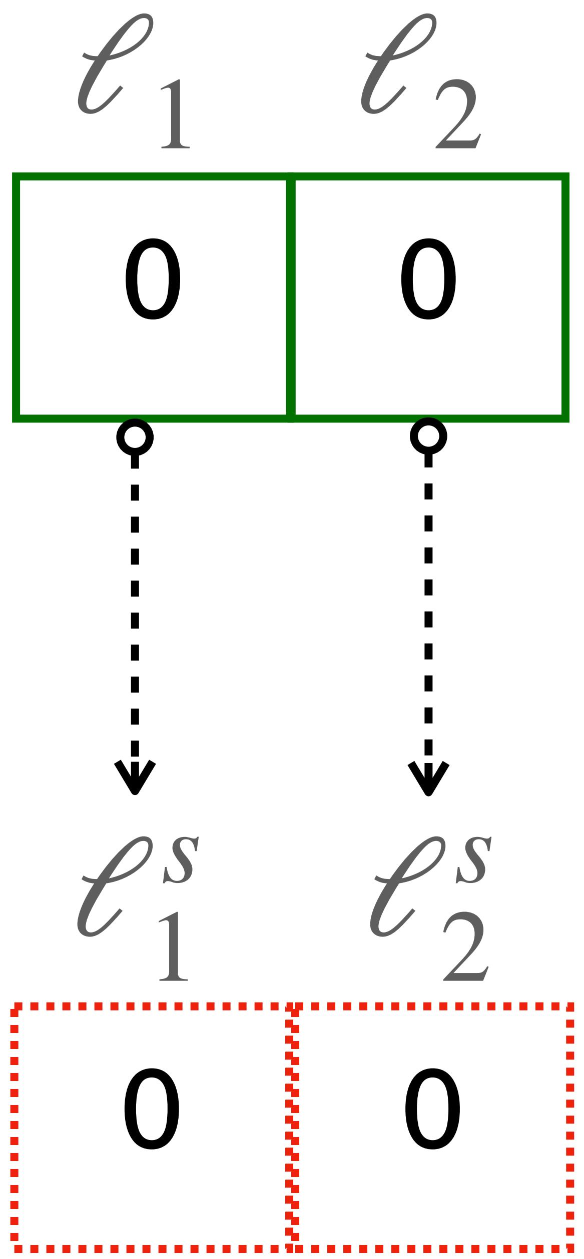
Checkpoint

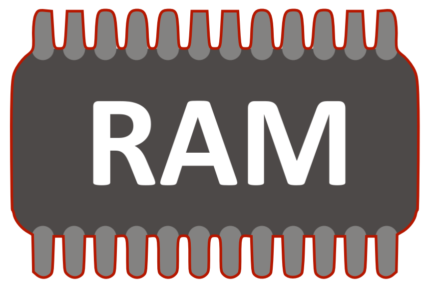
pc ⟶ Checkpoint

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```



Nonvolatile memory
Stable values





Volatile memory
Unstable values

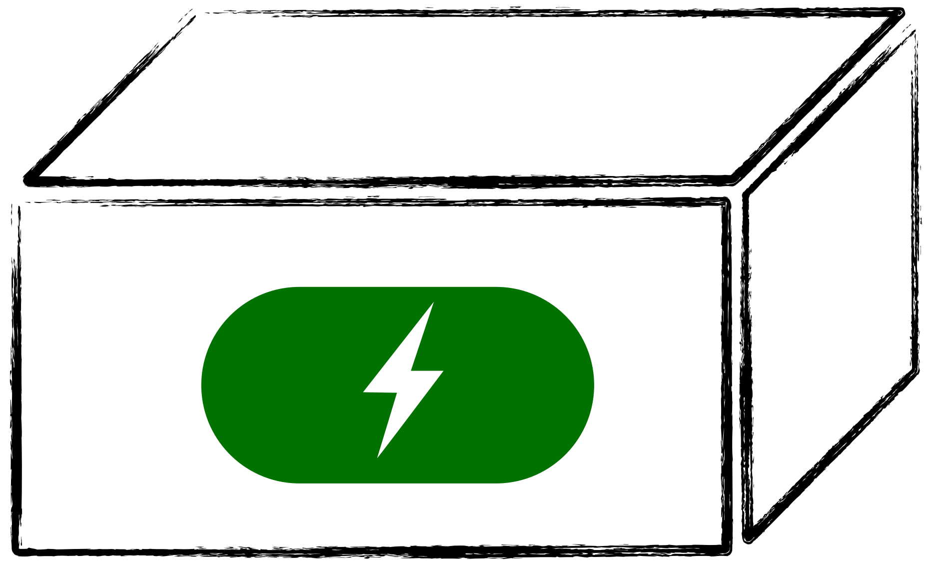
Final memory state we expect:

ℓ_1	ℓ_2
2	2

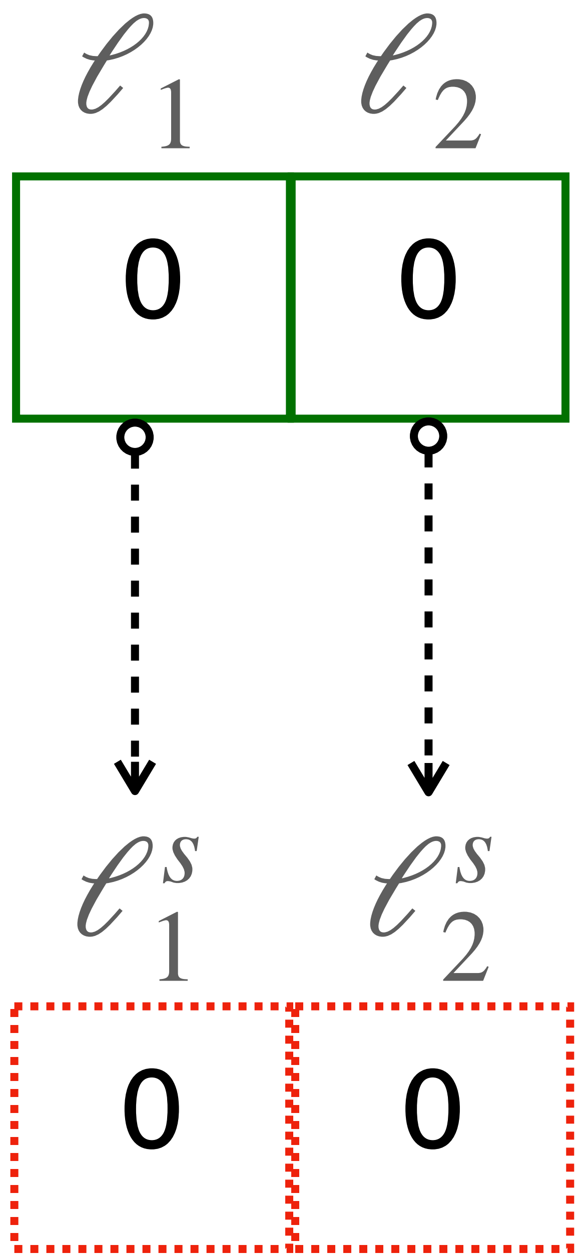
Solution: Checkpointing blocks (checkpoint-restore-finalize)

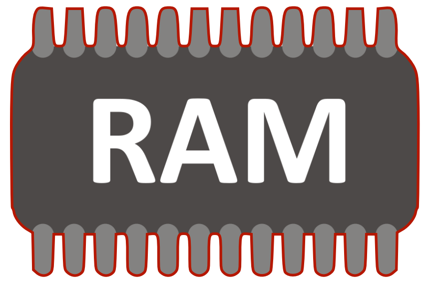
pc ⟶ Checkpoint

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```



Nonvolatile memory
Stable values





Volatile memory
Unstable values

Final memory state we expect:

ℓ_1	ℓ_2
2	2

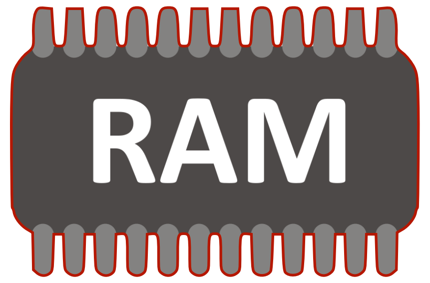
Solution: Checkpointing blocks (checkpoint-restore-finalize)

Checkpoint

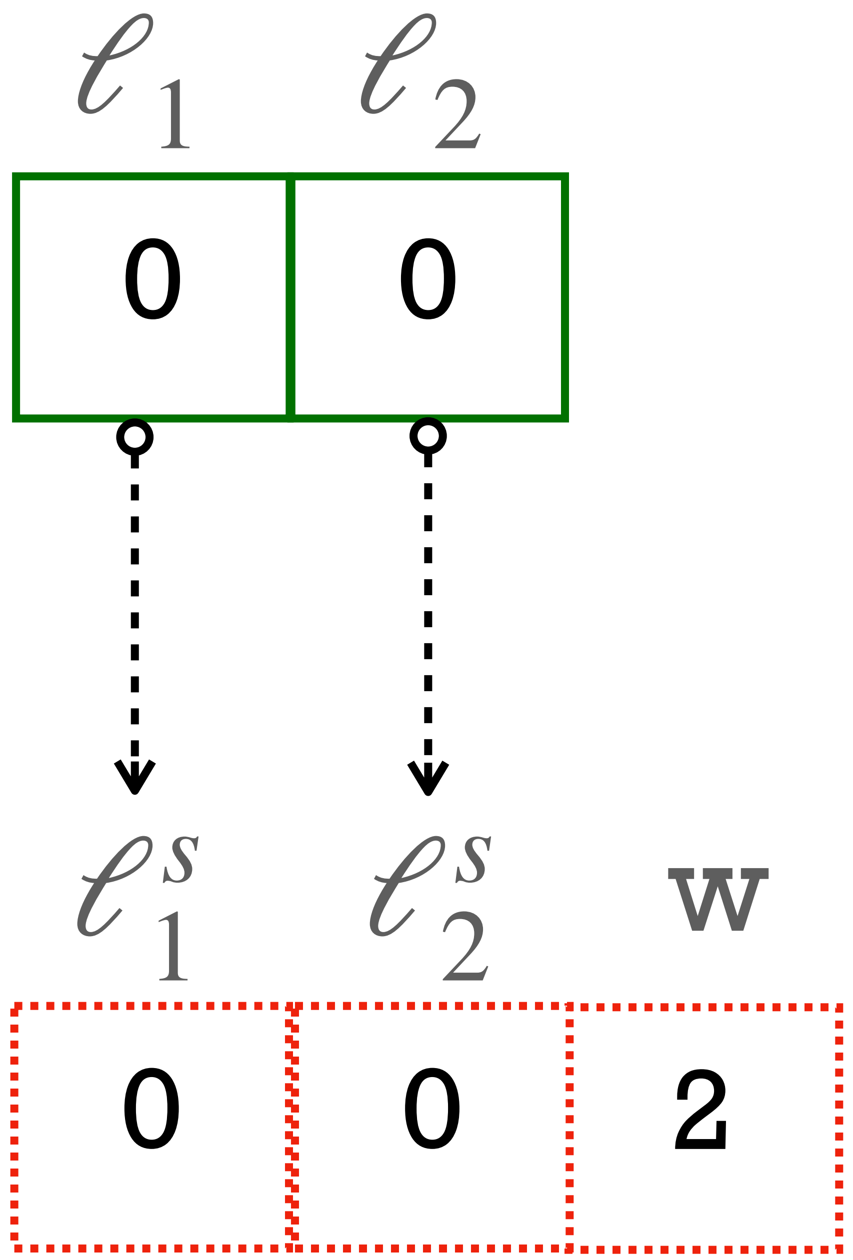
pc →

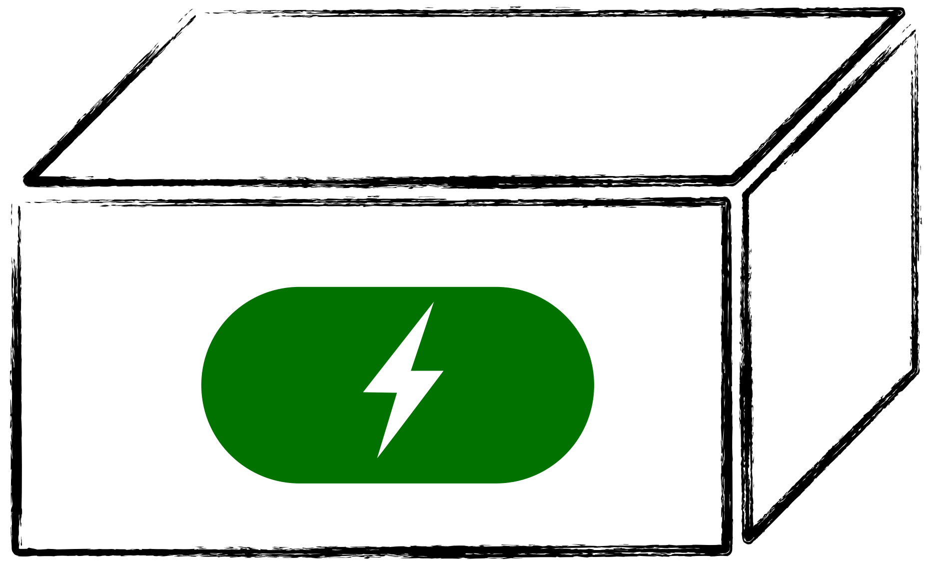
```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```

Nonvolatile memory
Stable values

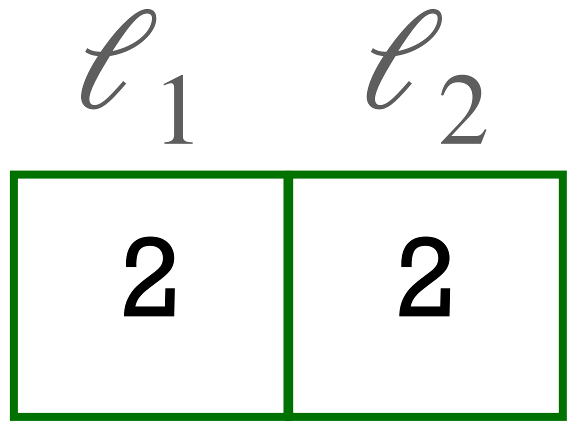


Volatile memory
Unstable values



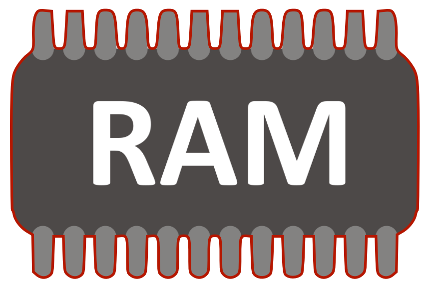


Final memory state we expect:

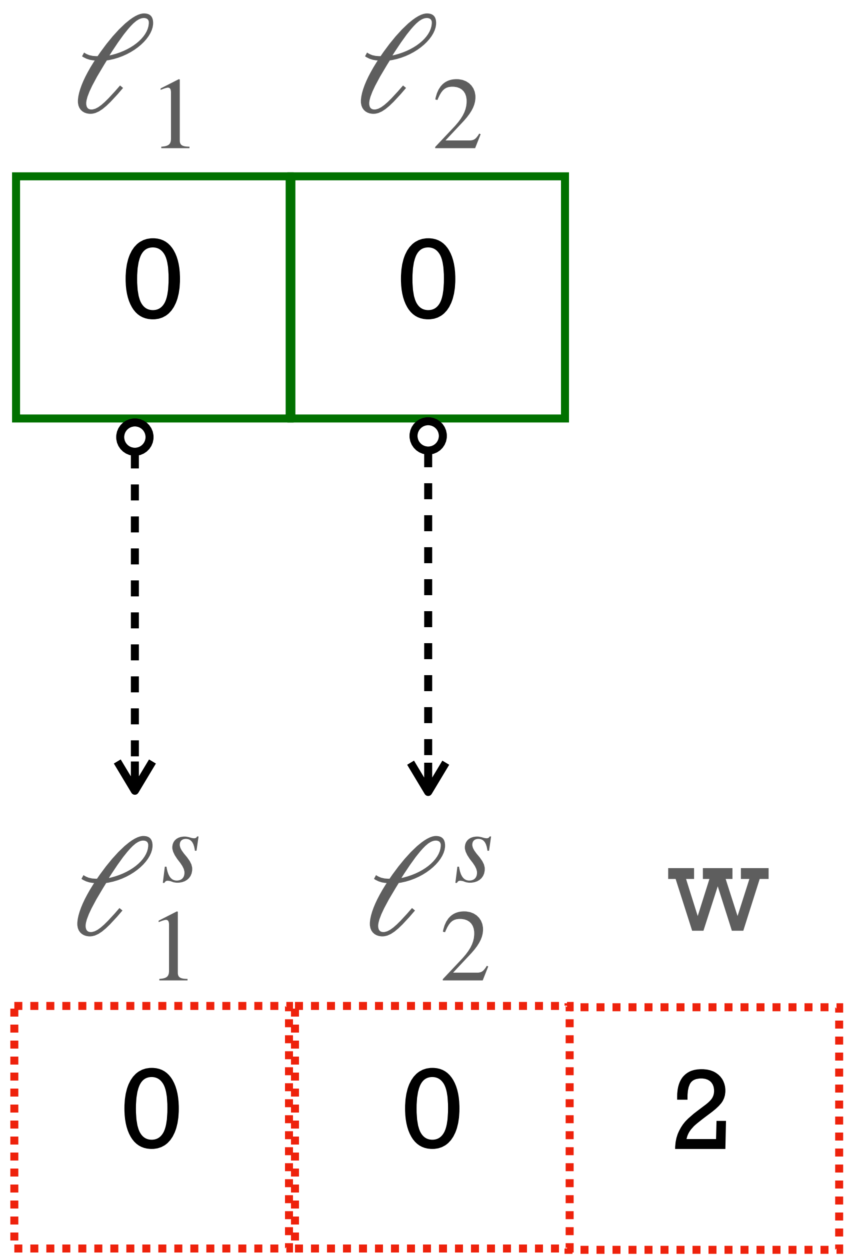


Solution: Checkpointing blocks (checkpoint-restore-finalize)

Nonvolatile memory
Stable values



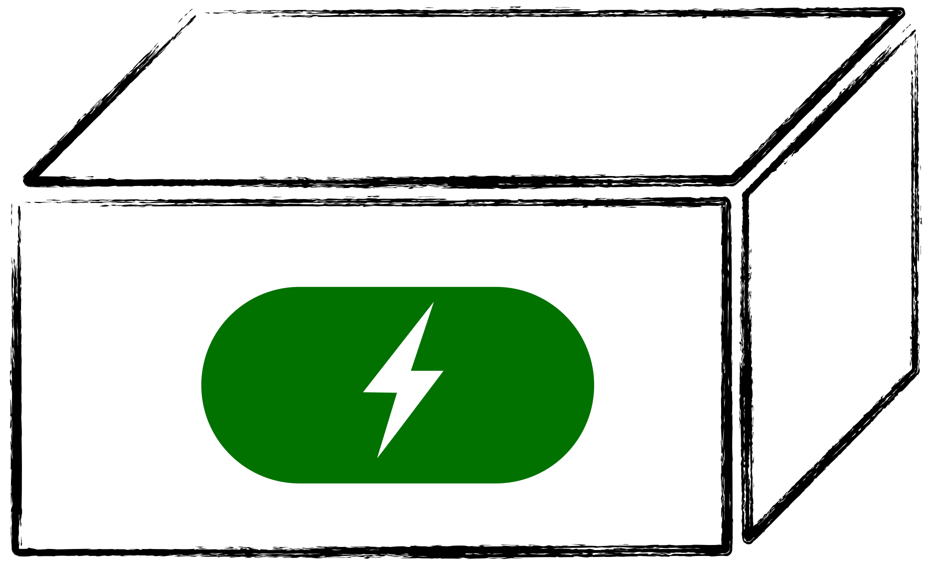
Volatile memory
Unstable values



pc →

Checkpoint

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```



Final memory state we expect:

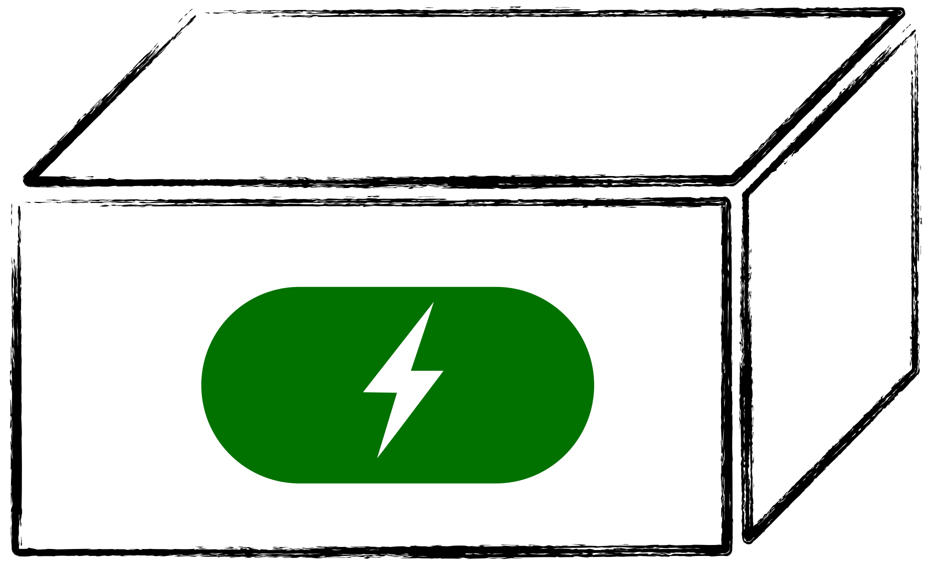
ℓ_1	ℓ_2
2	2

Solution: Checkpointing blocks (checkpoint-restore-finalize)

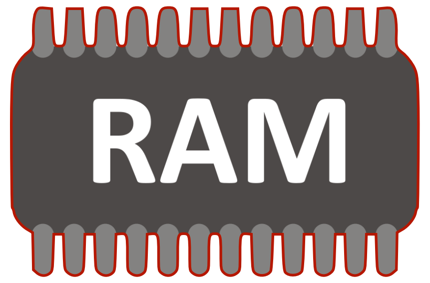
Checkpoint

pc →

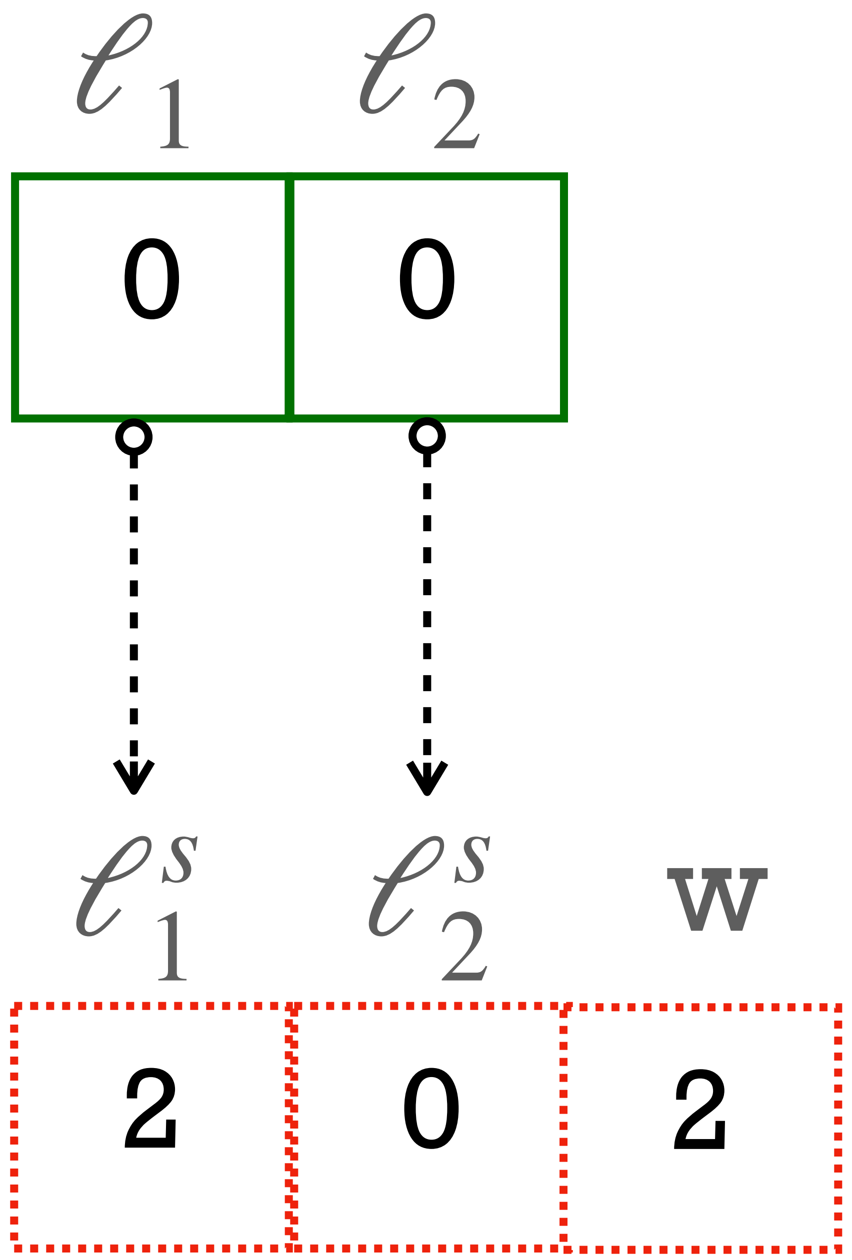
```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```



Nonvolatile memory
Stable values



Volatile memory
Unstable values

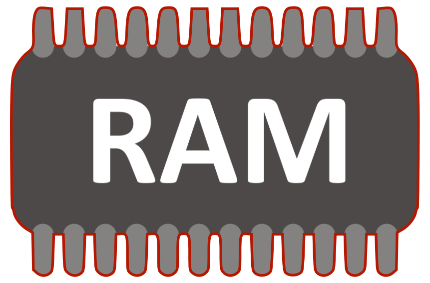


Final memory state we expect:

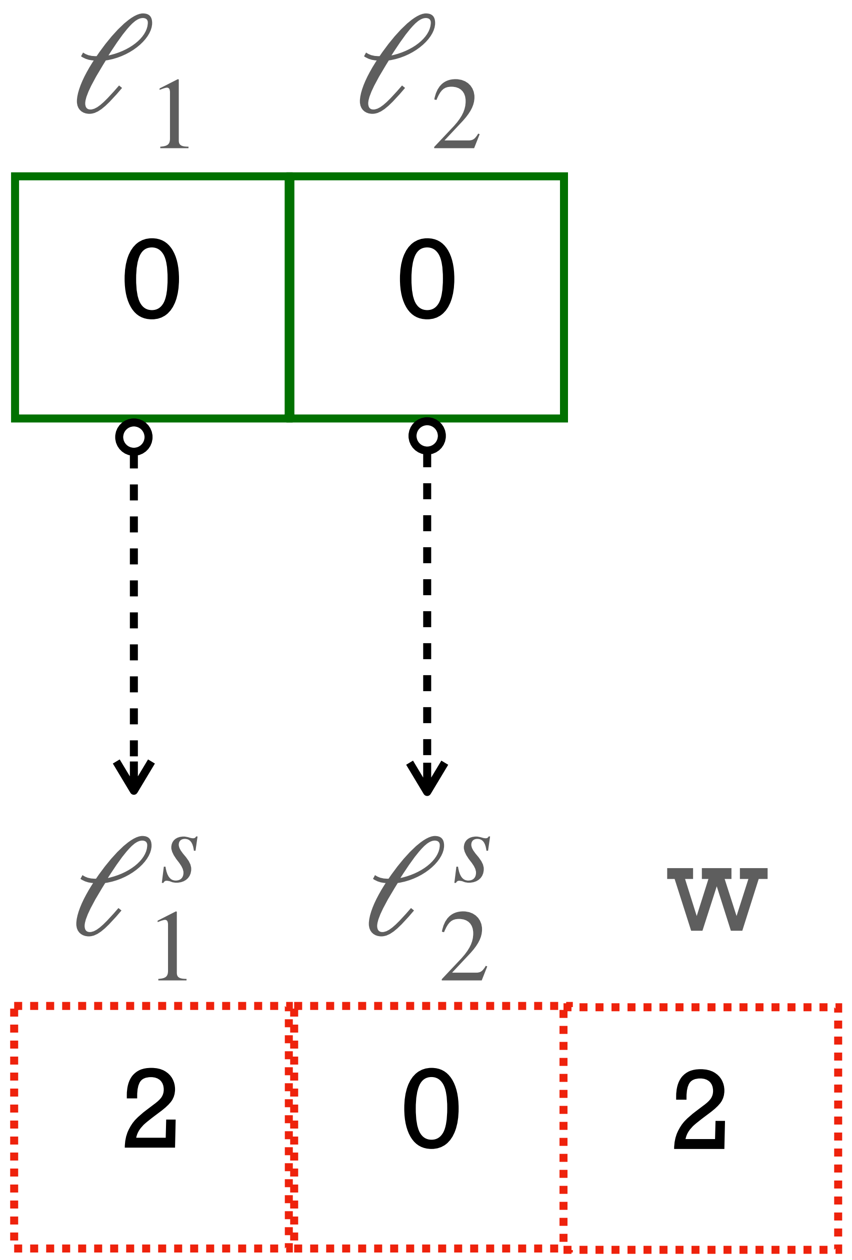
ℓ_1	ℓ_2
2	2

Solution: Checkpointing blocks (checkpoint-restore-finalize)

Nonvolatile memory
Stable values



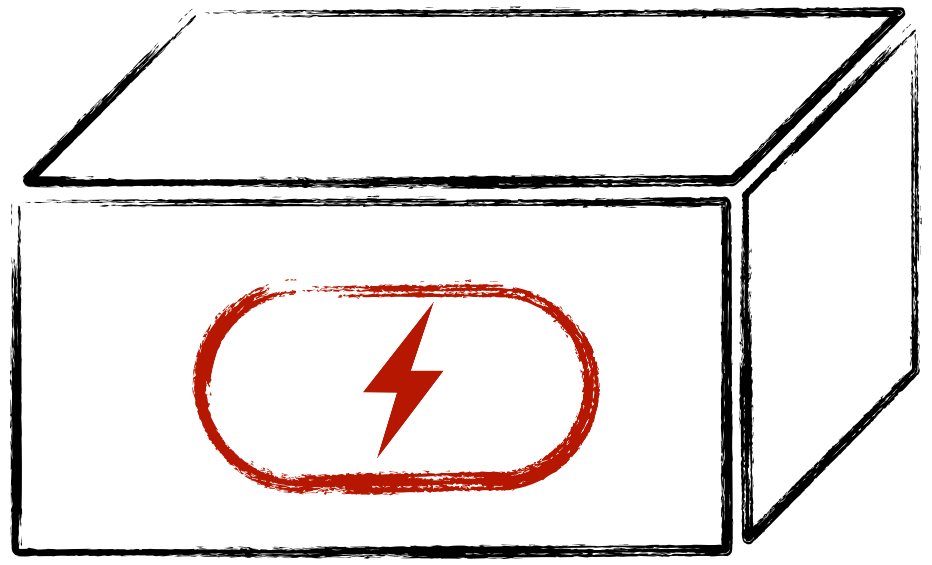
Volatile memory
Unstable values



Checkpoint

pc →

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```

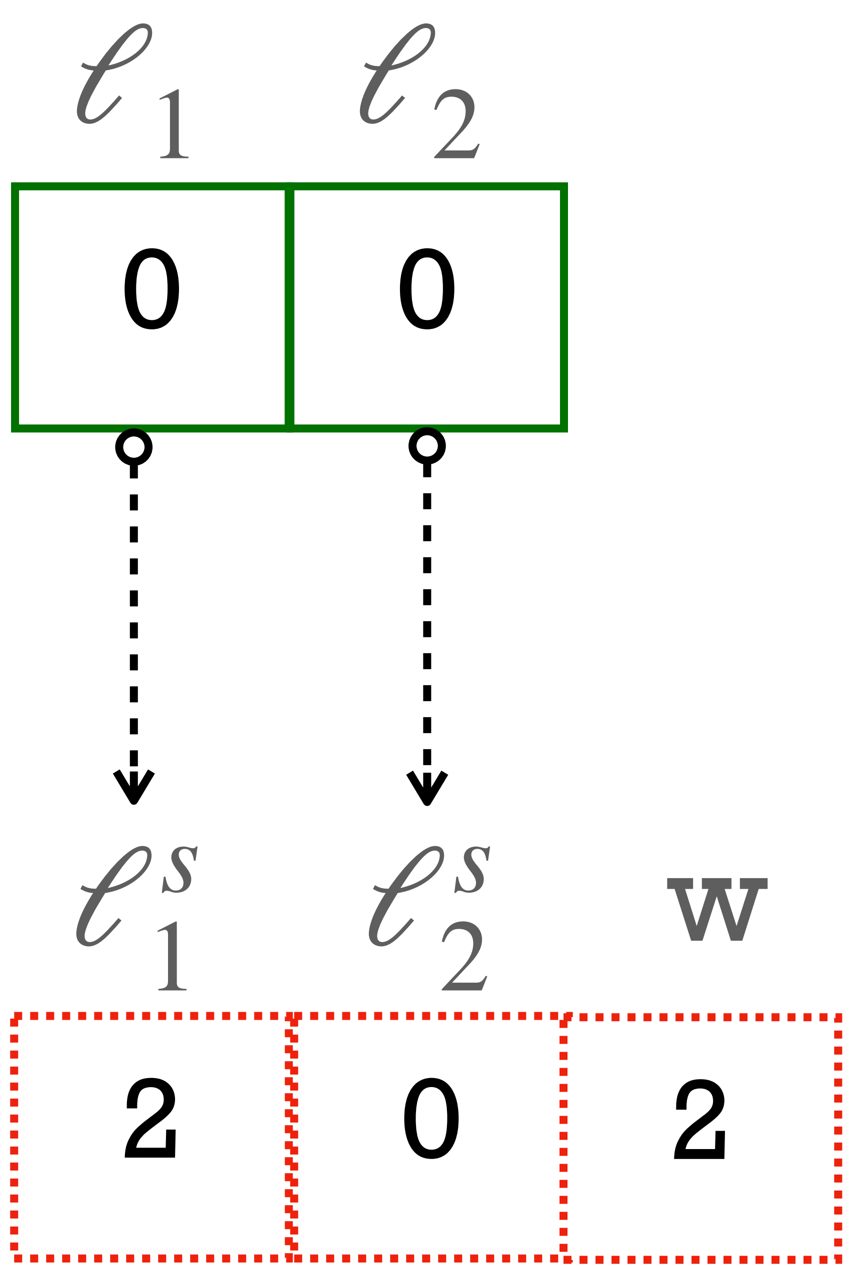


Final memory state we expect:

ℓ_1	ℓ_2
2	2

Solution: Checkpointing blocks (checkpoint-restore-finalize)

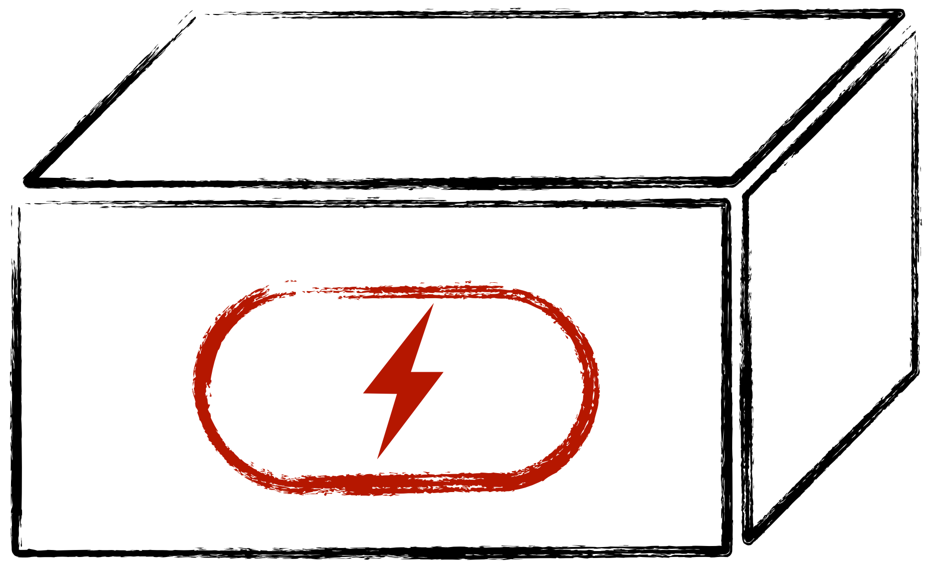
Power failure



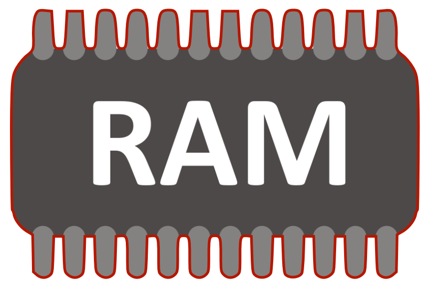
Checkpoint

pc →

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```



Nonvolatile memory
Stable values



Volatile memory
Unstable values

Final memory state we expect:

ℓ_1	ℓ_2
2	2

Solution: Checkpointing blocks (checkpoint-restore-finalize)

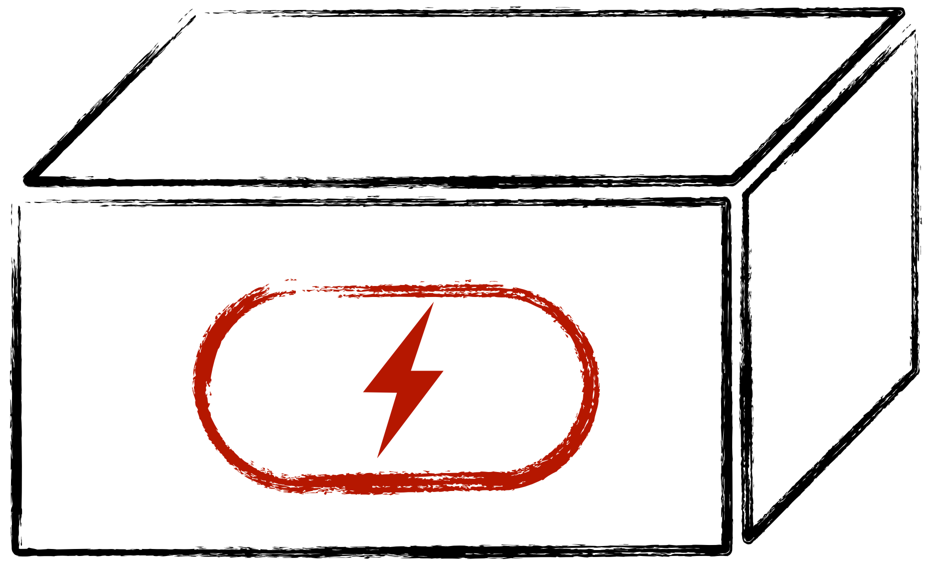
Power failure

ℓ_1	ℓ_2
0	0

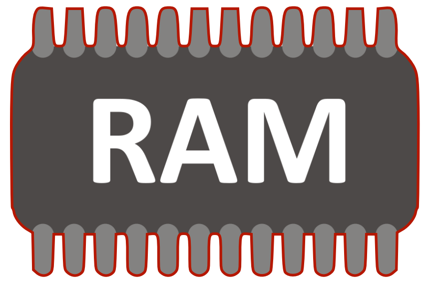
Checkpoint

pc →

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```



Nonvolatile memory
Stable values



Volatile memory
Unstable values

Final memory state we expect:

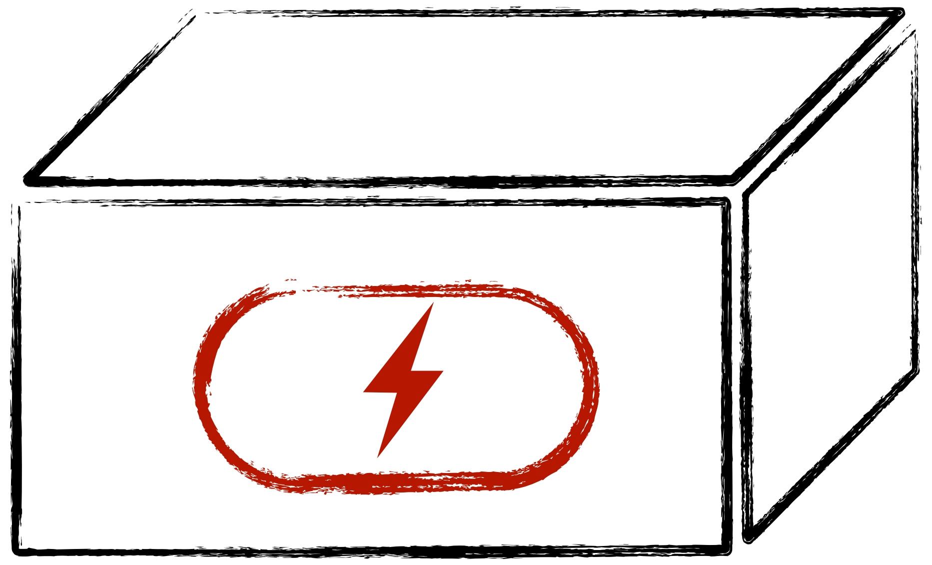
ℓ_1	ℓ_2
2	2

Solution: Checkpointing blocks (checkpoint-restore-finalize)

Power failure

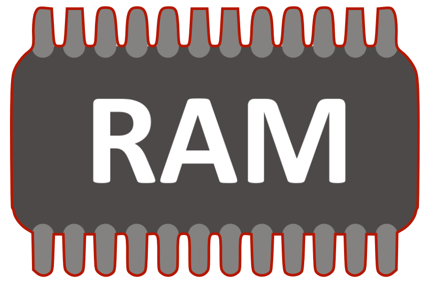
pc → Checkpoint

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```



Nonvolatile memory
Stable values

ℓ_1	ℓ_2
0	0



Volatile memory
Unstable values

Final memory state we expect:

ℓ_1	ℓ_2
2	2

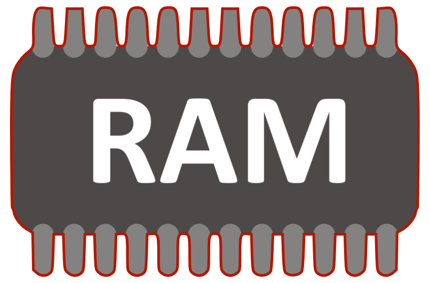
Solution: Checkpointing blocks (checkpoint-restore-finalize)

pc ⟶ Checkpoint

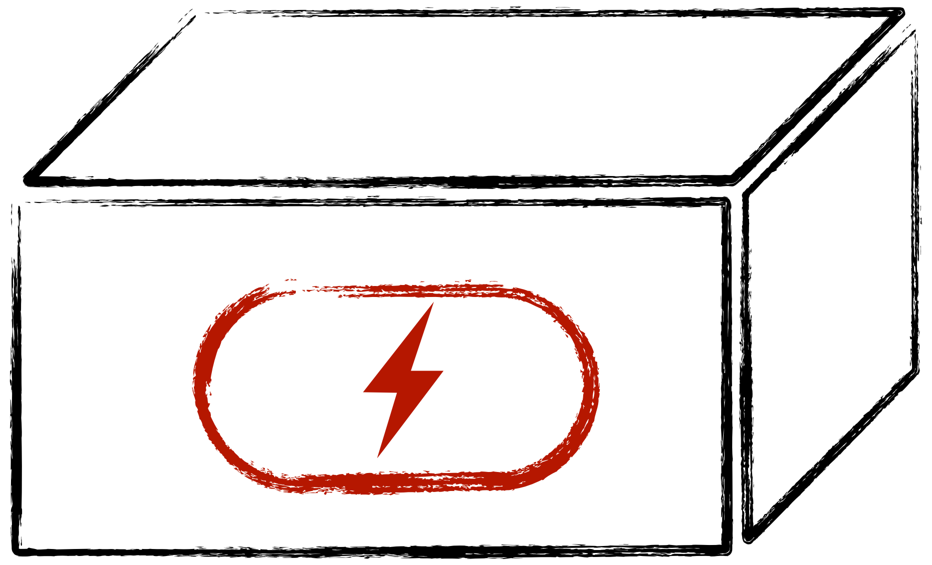
```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```

Nonvolatile memory
Stable values

ℓ_1	ℓ_2
0	0



Volatile memory
Unstable values



Final memory state we expect:

ℓ_1	ℓ_2
2	2

Solution: Checkpointing blocks (checkpoint-restore-finalize)

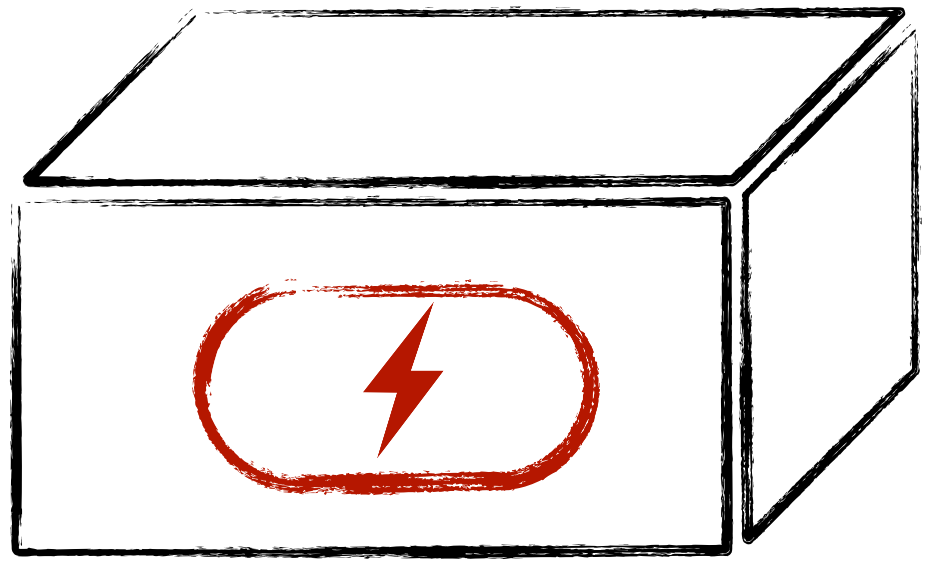
Recharge

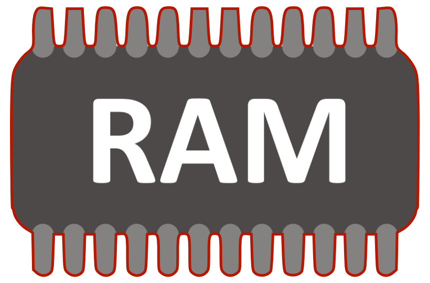
pc → Checkpoint

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```

Nonvolatile memory
Stable values

ℓ_1	ℓ_2
0	0





Volatile memory
Unstable values

Final memory state we expect:

ℓ_1	ℓ_2
2	2

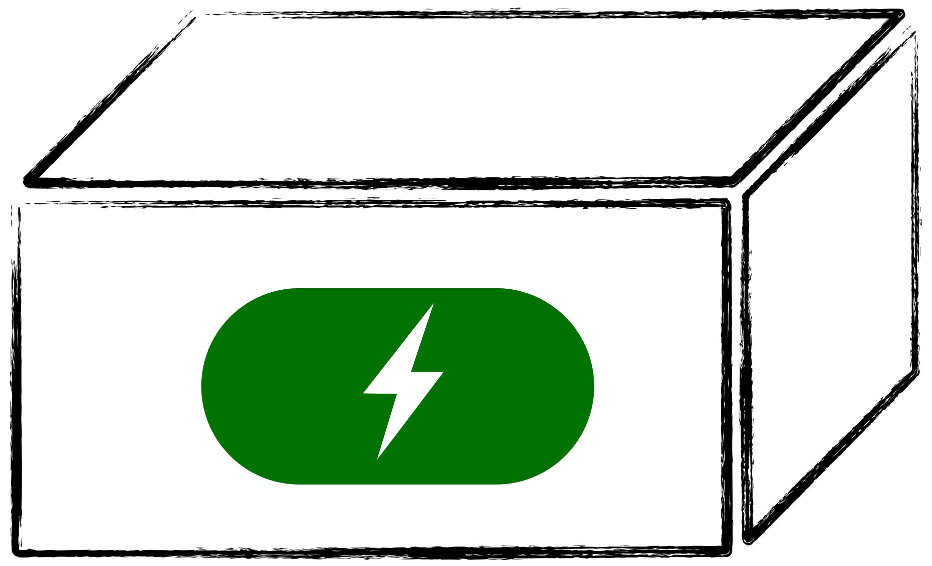
Solution: Checkpointing blocks (checkpoint-restore-finalize)

Recharge

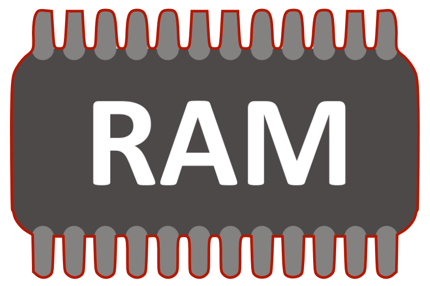
ℓ_1	ℓ_2
0	0

pc → Checkpoint

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```



Nonvolatile memory
Stable values



Volatile memory
Unstable values

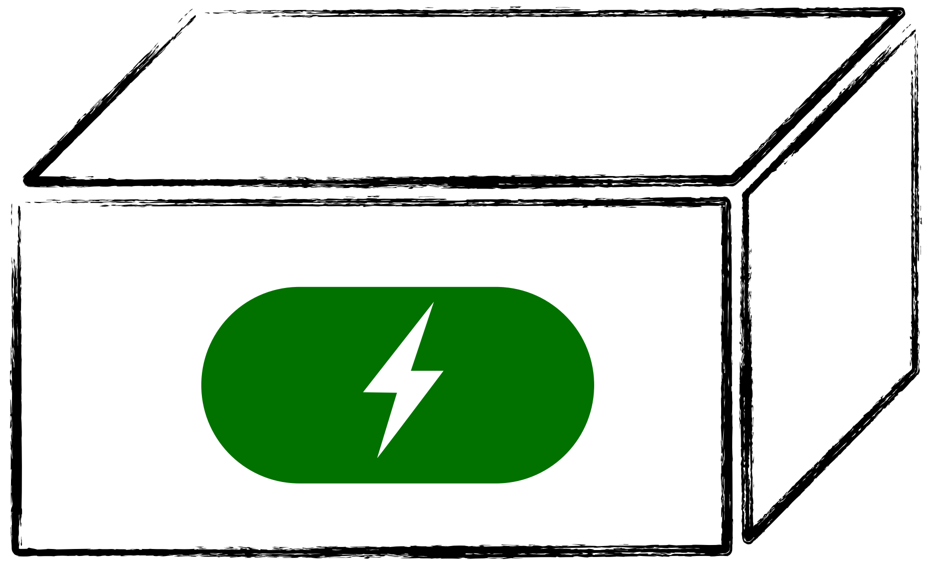
Final memory state we expect:

ℓ_1	ℓ_2
2	2

Solution: Checkpointing blocks (checkpoint-restore-finalize)

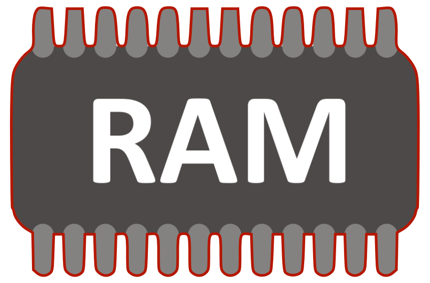
pc → Checkpoint

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```



Nonvolatile memory
Stable values

ℓ_1	ℓ_2
0	0



Volatile memory
Unstable values

Final memory state we expect:

ℓ_1	ℓ_2
2	2

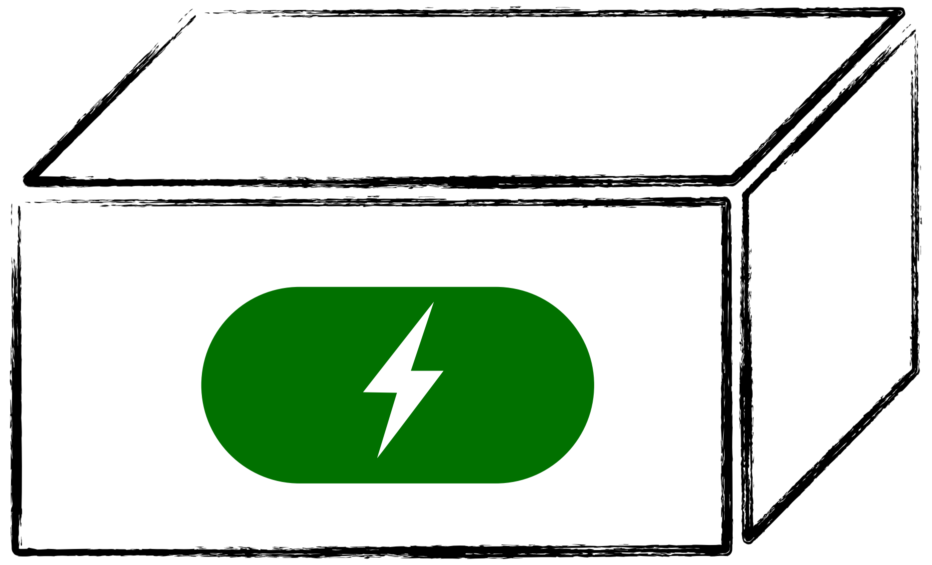
Solution: Checkpointing blocks (checkpoint-restore-finalize)

Restore

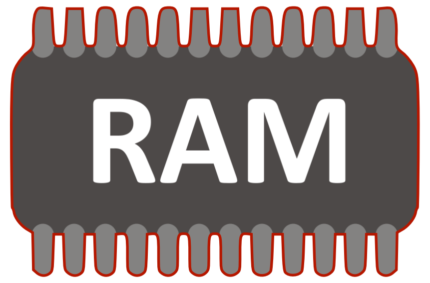
ℓ_1	ℓ_2
0	0

pc → Checkpoint

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```



Nonvolatile memory
Stable values



Volatile memory
Unstable values

Final memory state we expect:

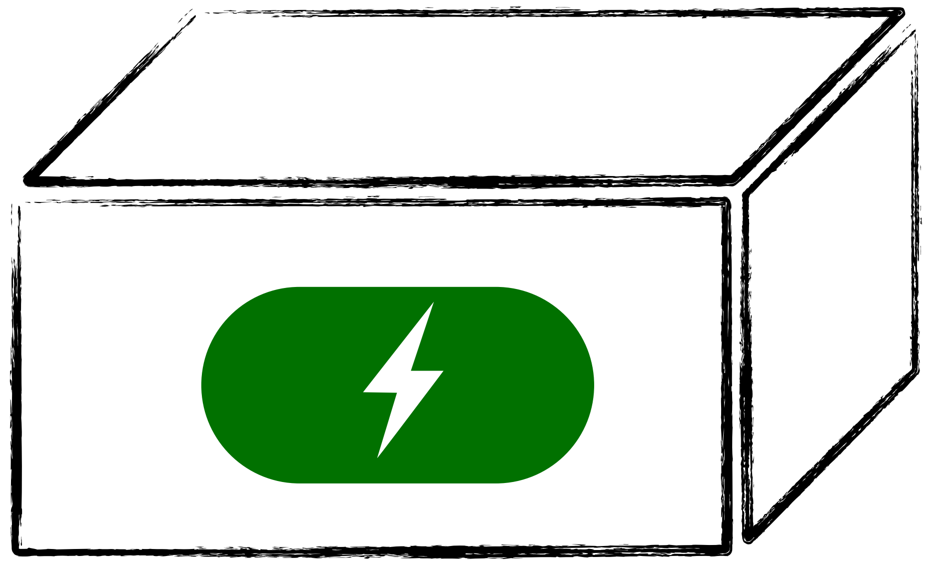
ℓ_1	ℓ_2
2	2

Solution: Checkpointing blocks (checkpoint-restore-finalize)

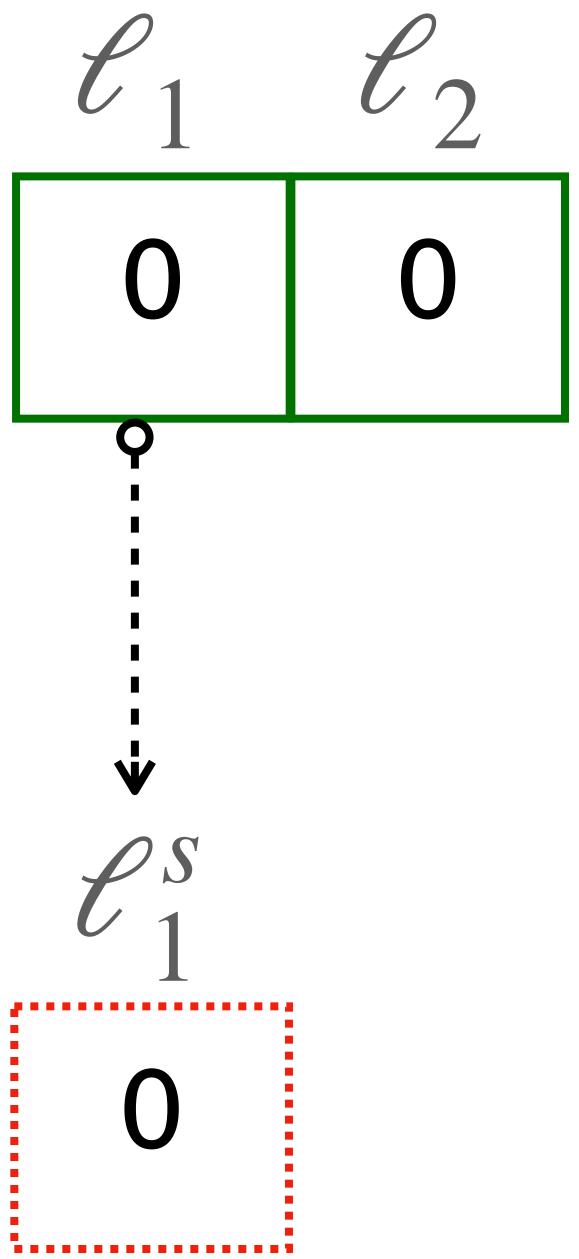
Restore

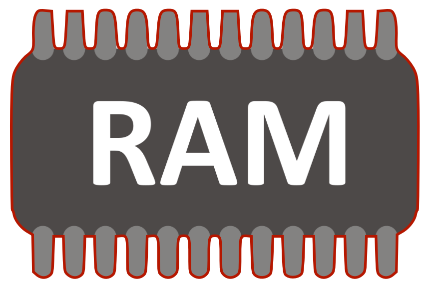
pc → Checkpoint

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```



Nonvolatile memory
Stable values





Volatile memory
Unstable values

Final memory state we expect:

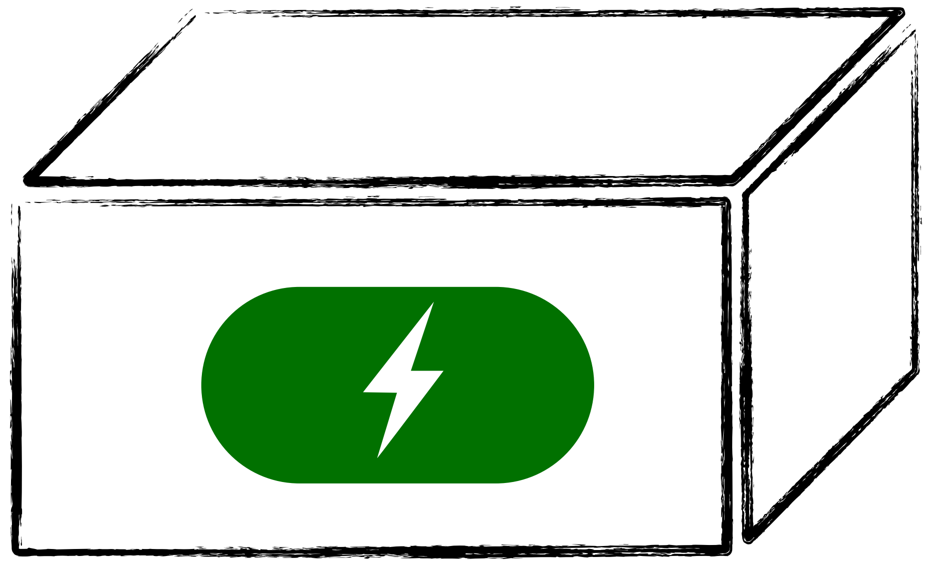
ℓ_1	ℓ_2
2	2

Solution: Checkpointing blocks (checkpoint-restore-finalize)

Restore

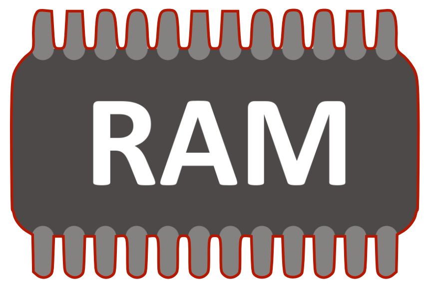
pc → Checkpoint

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```



Nonvolatile memory
Stable values

ℓ_1	ℓ_2
0	0



Volatile memory
Unstable values

ℓ^s_1	ℓ^s_2
0	0

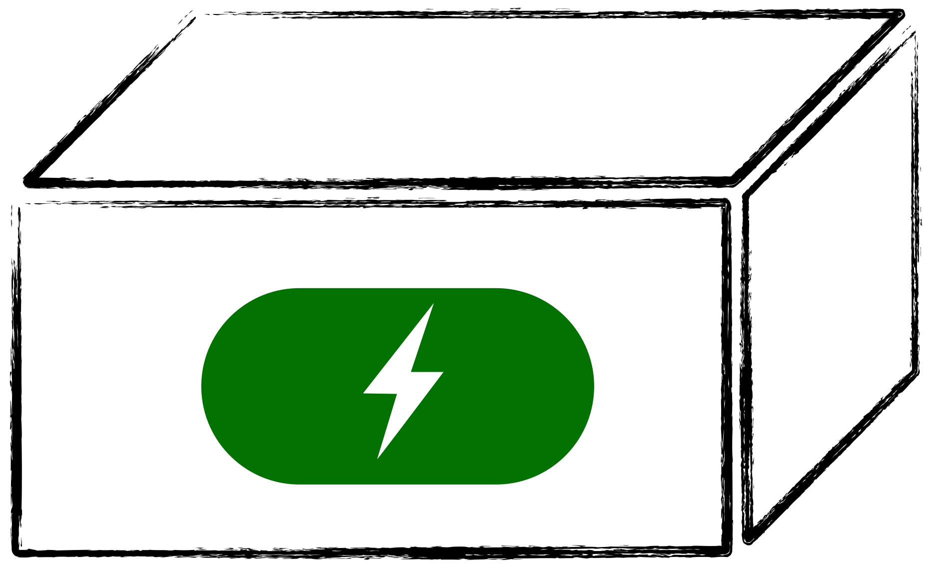
Final memory state we expect:

ℓ_1	ℓ_2
2	2

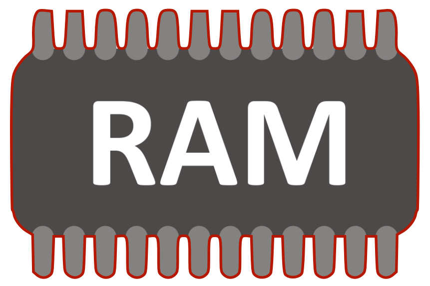
Solution: Checkpointing blocks (checkpoint-restore-finalize)

pc ⟶ Checkpoint

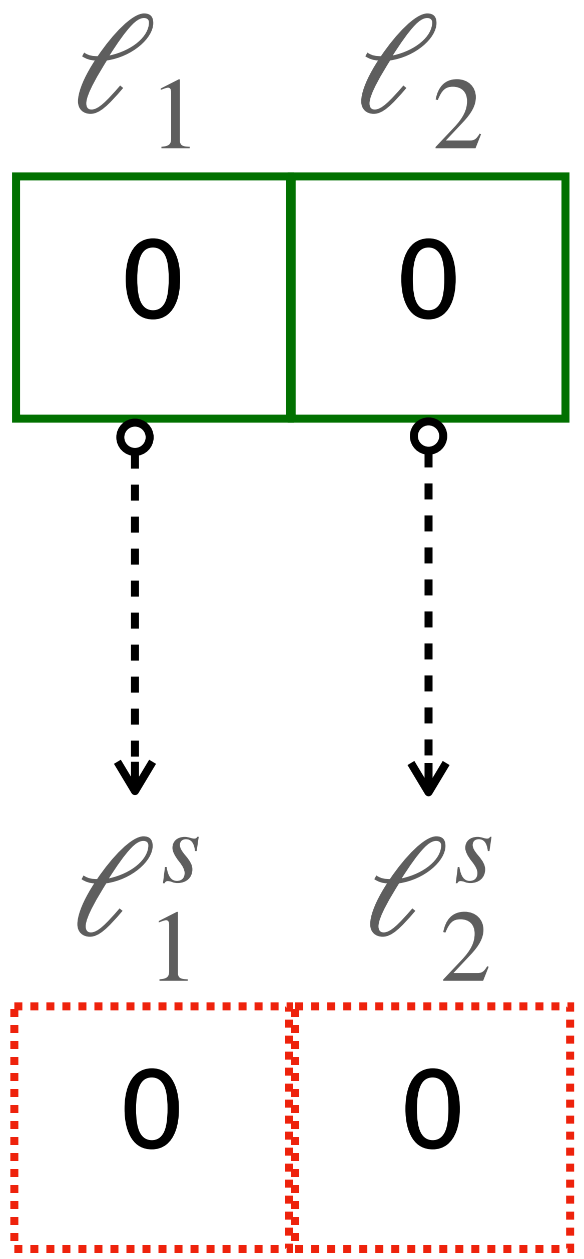
```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```



Nonvolatile memory
Stable values



Volatile memory
Unstable values

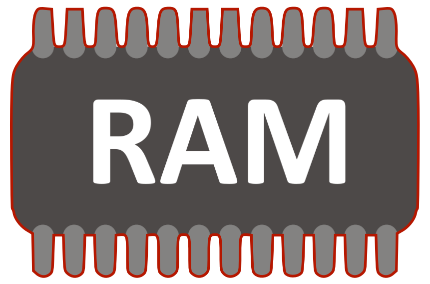


Final memory state we expect:

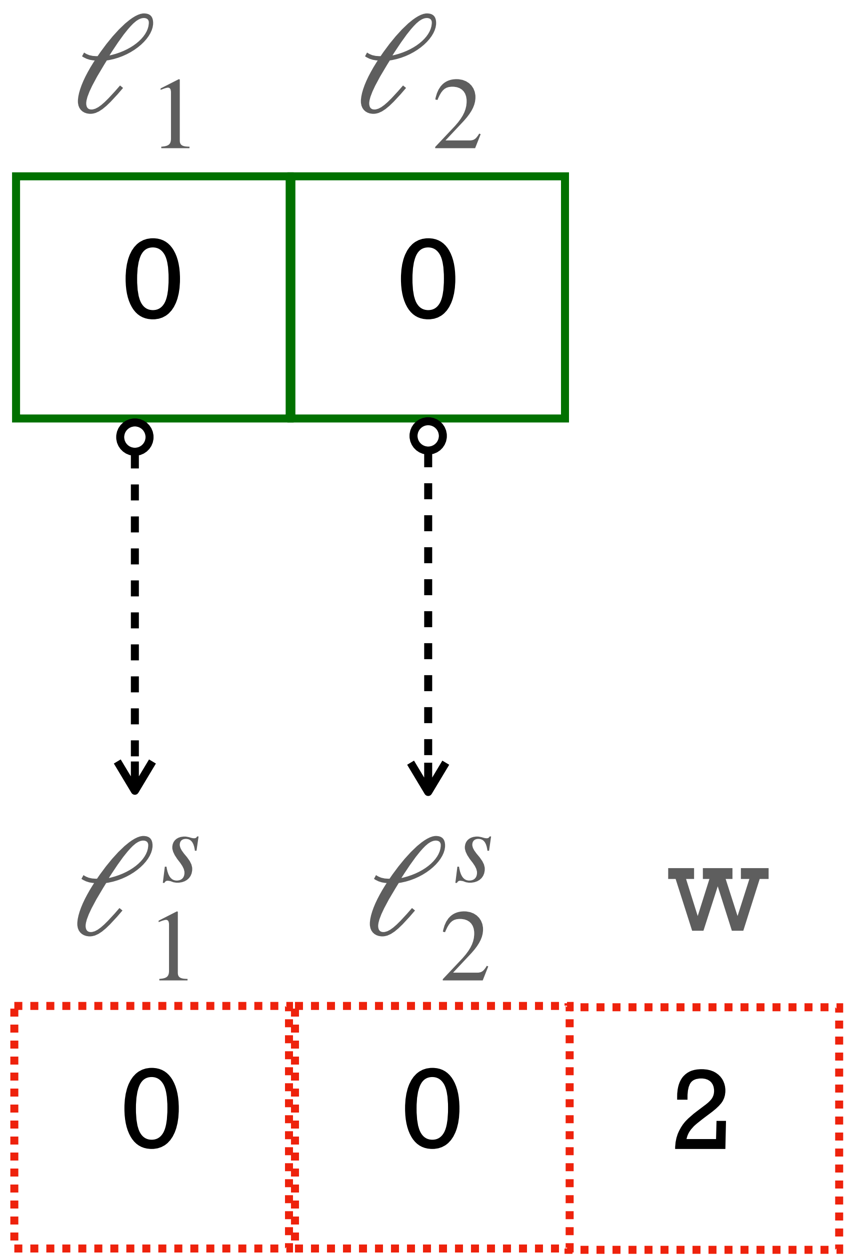
ℓ_1	ℓ_2
2	2

Solution: Checkpointing blocks (checkpoint-restore-finalize)

Nonvolatile memory
Stable values



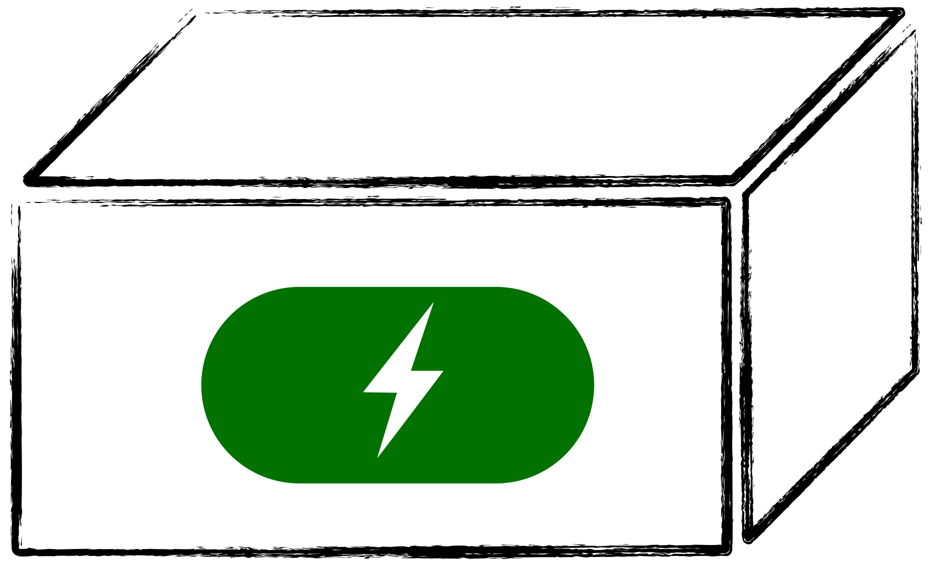
Volatile memory
Unstable values



Checkpoint

pc →

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```

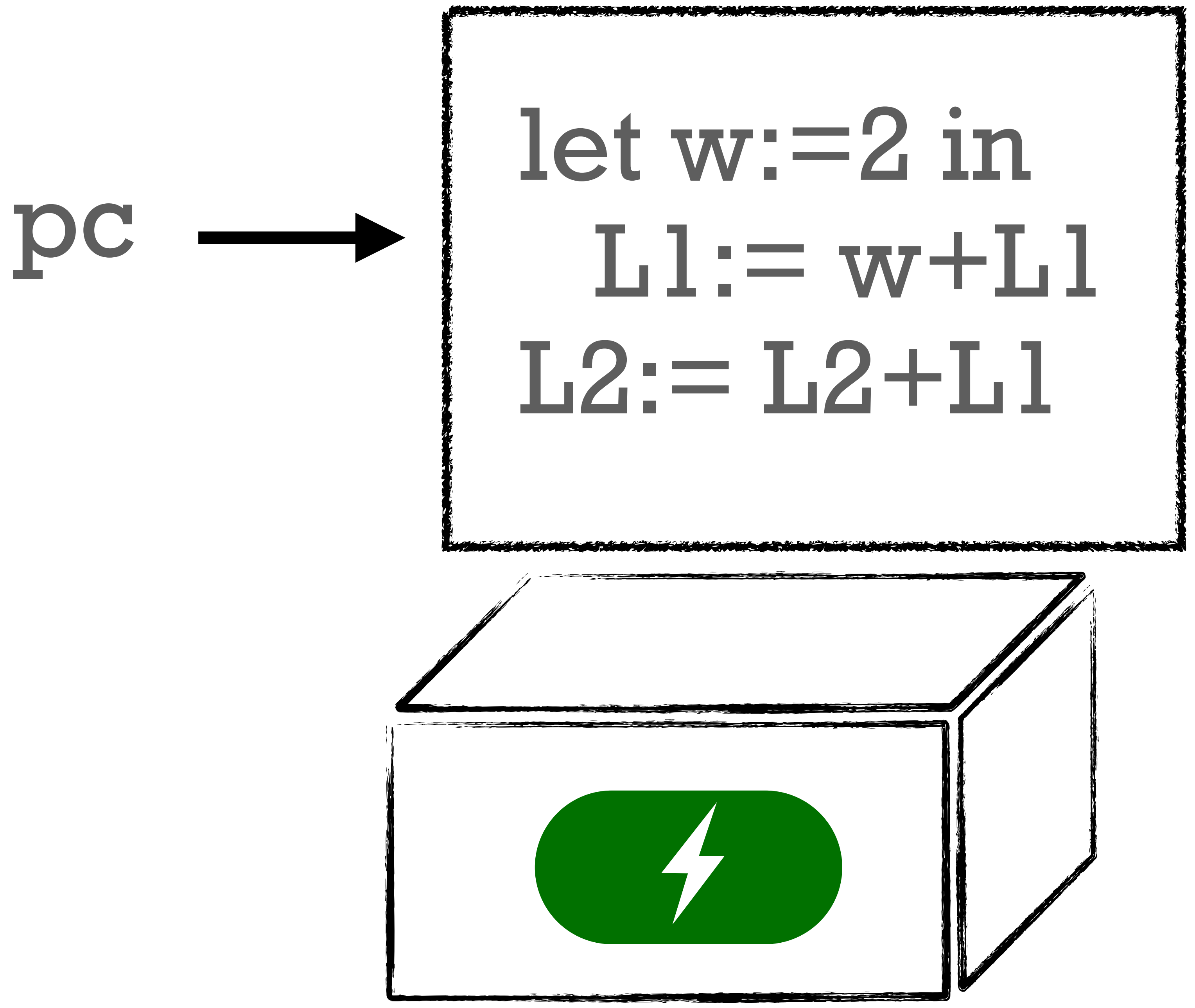


Final memory state we expect:

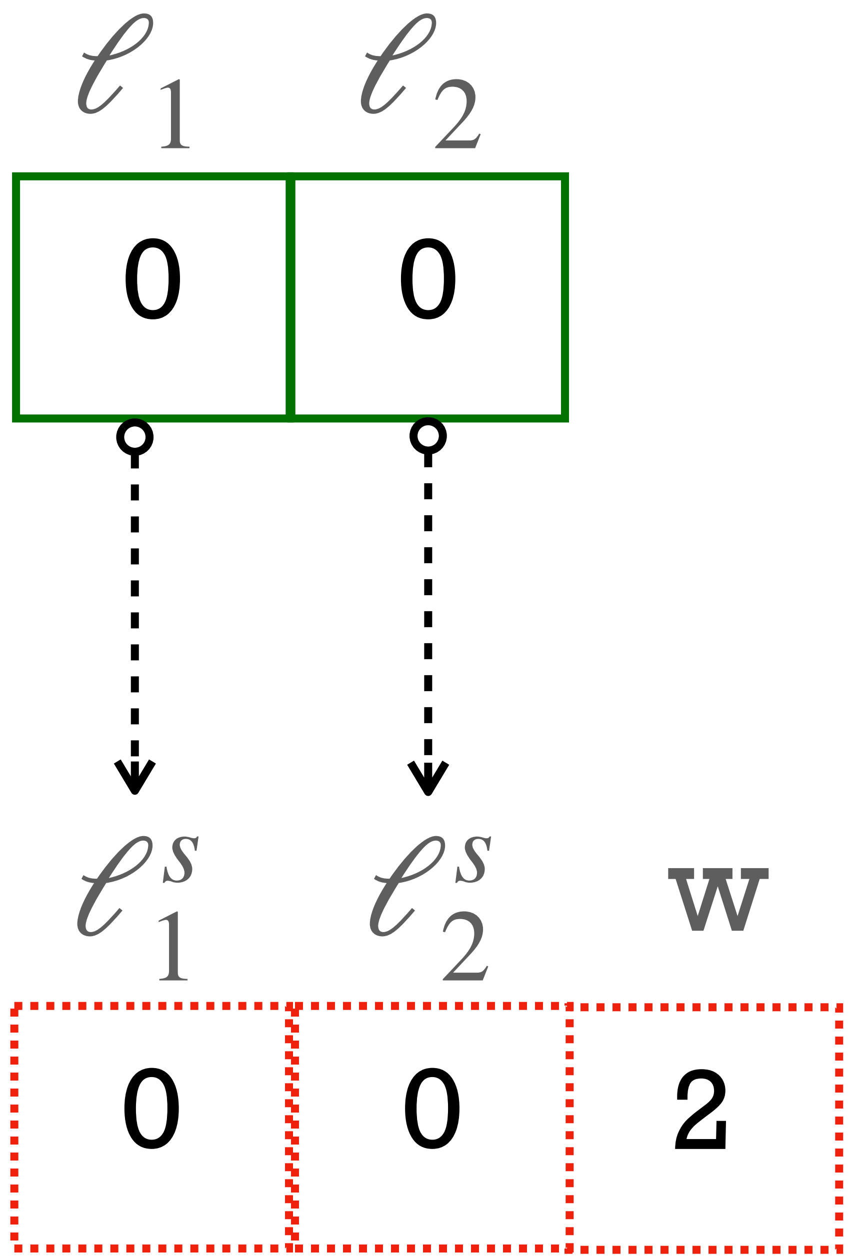
ℓ_1	ℓ_2
2	2

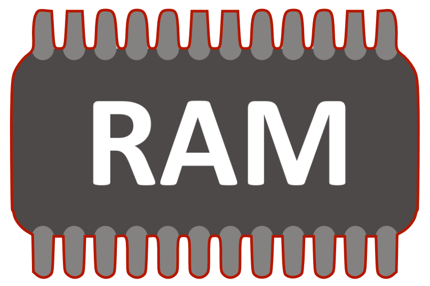
Solution: Checkpointing blocks (checkpoint-restore-finalize)

Checkpoint



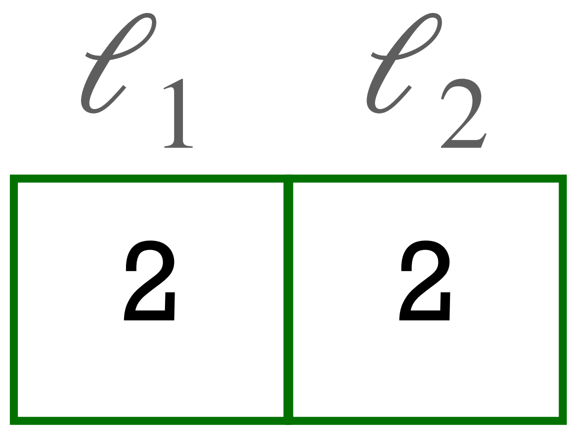
Nonvolatile memory
Stable values





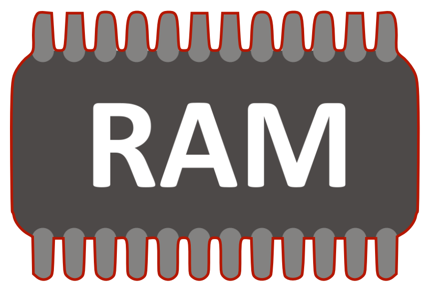
Volatile memory
Unstable values

Final memory state we expect:

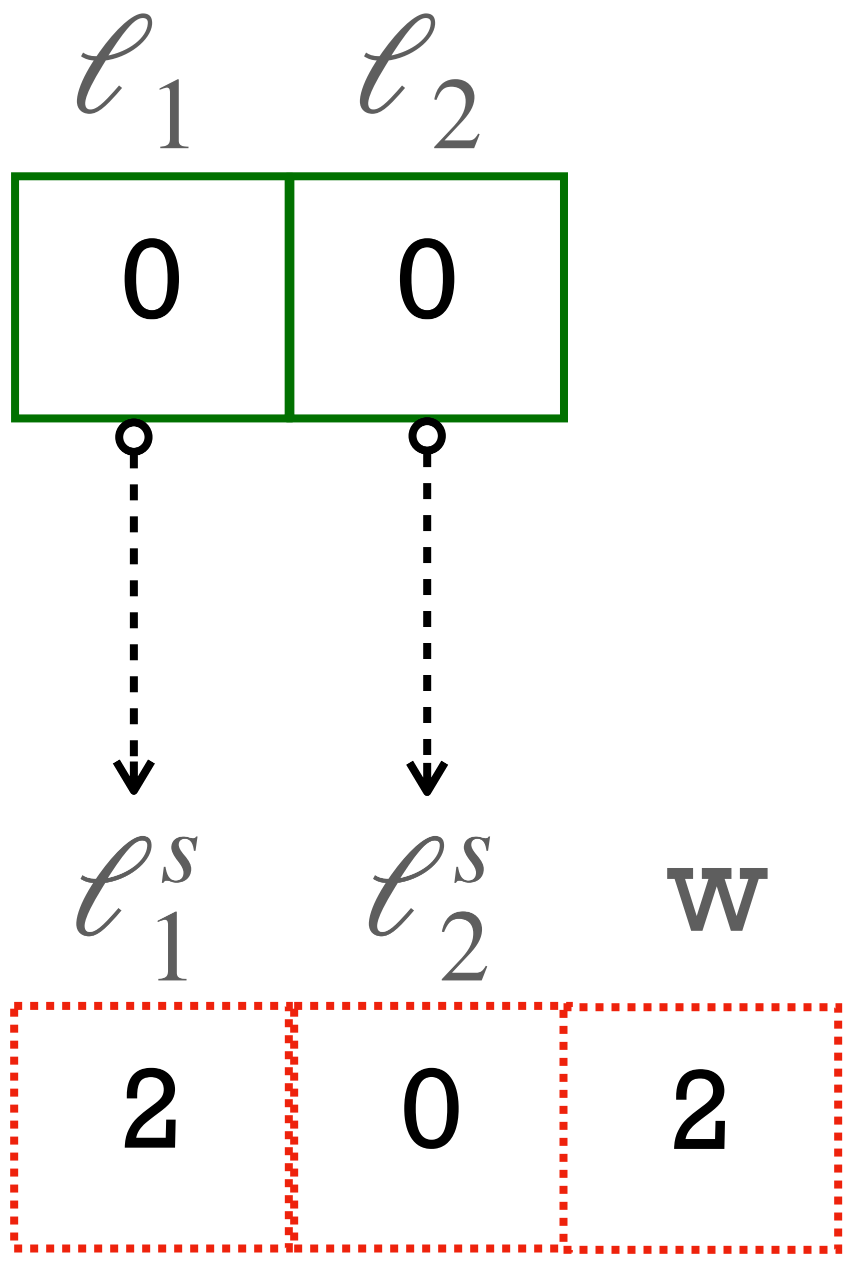


Solution: Checkpointing blocks (checkpoint-restore-finalize)

Nonvolatile memory
Stable values



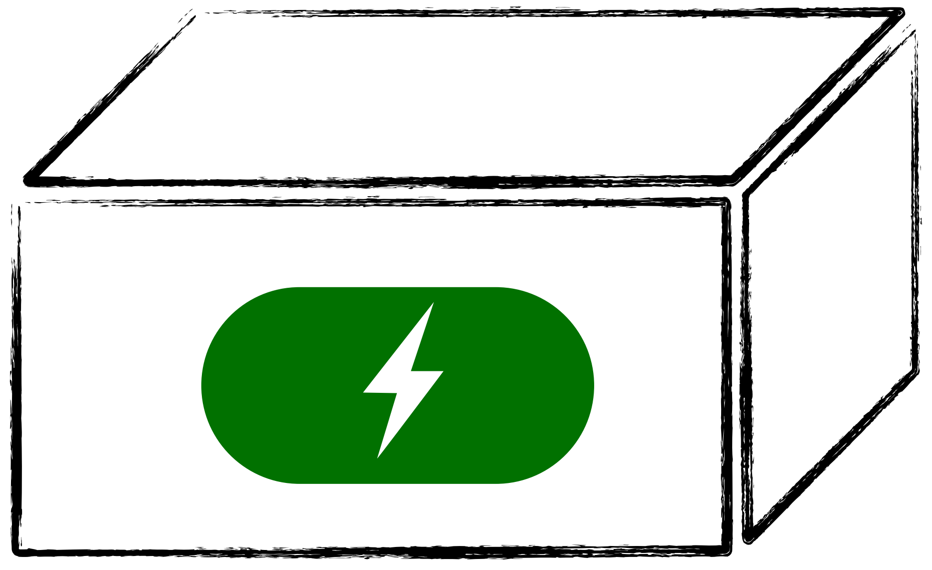
Volatile memory
Unstable values



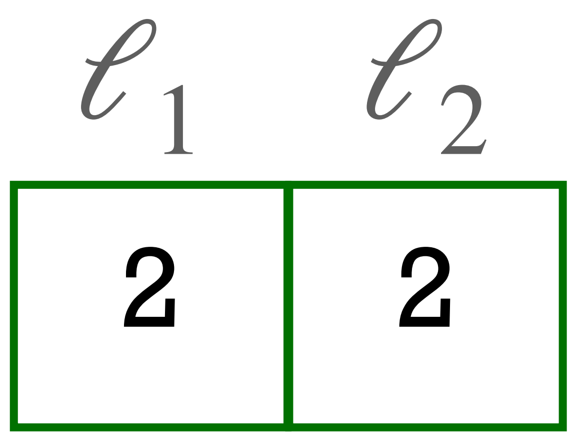
pc →

Checkpoint

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```

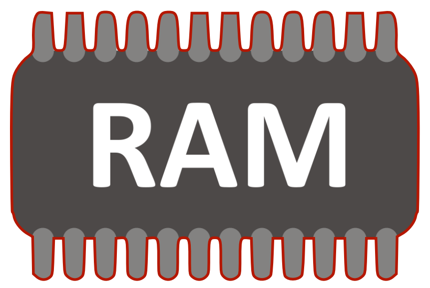


Final memory state we expect:

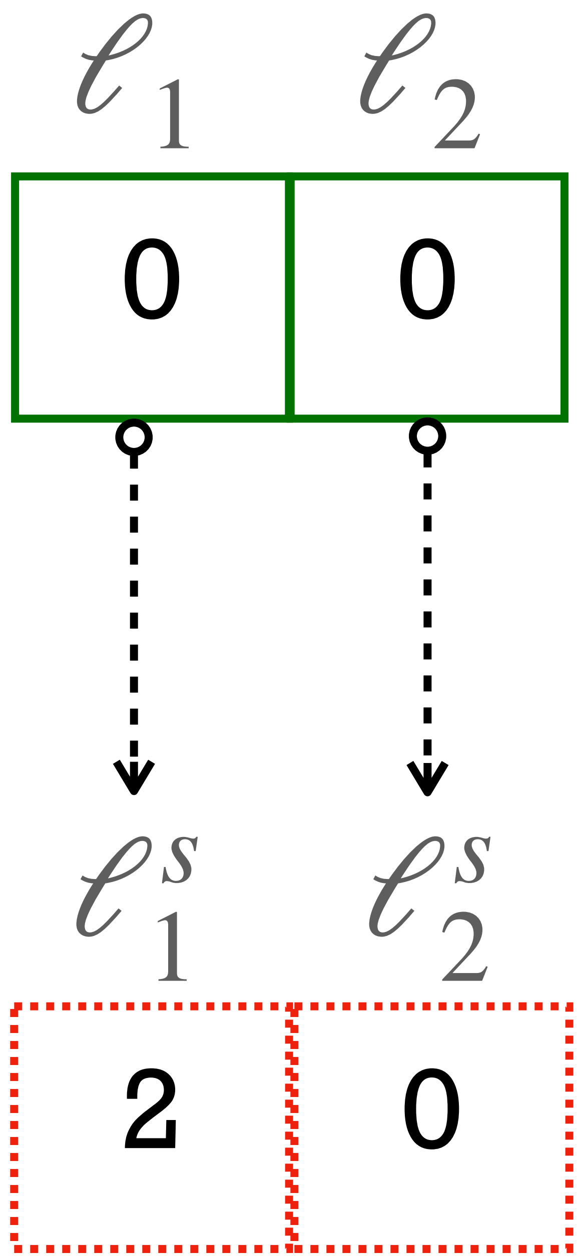


Solution: Checkpointing blocks (checkpoint-restore-finalize)

Nonvolatile memory
Stable values



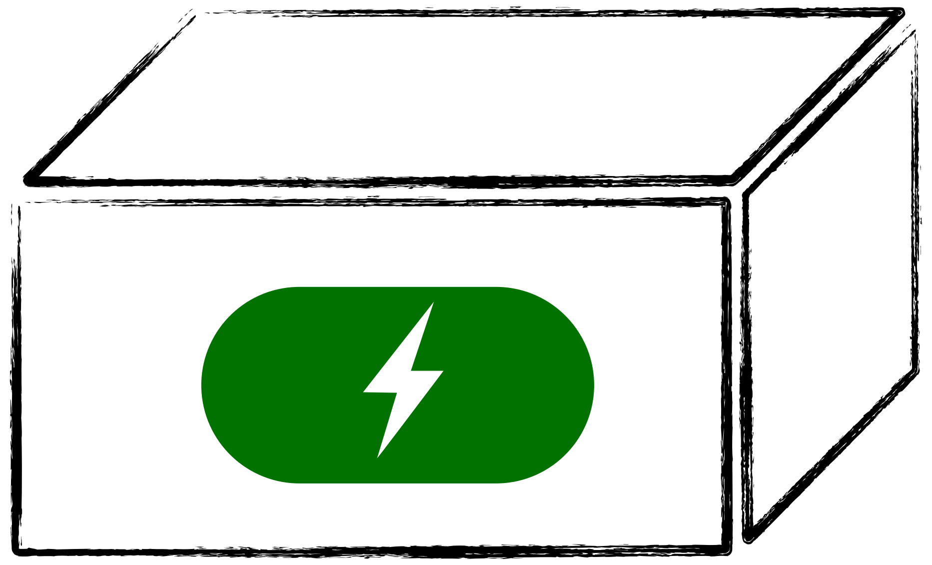
Volatile memory
Unstable values



Checkpoint

pc →

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```



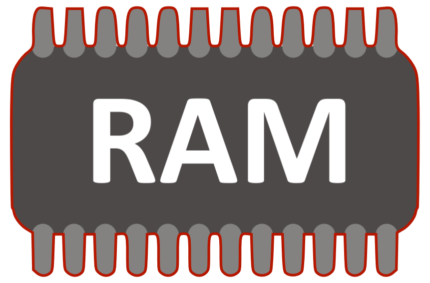
Final memory state we expect:

ℓ_1	ℓ_2
2	2

Solution: Checkpointing blocks (checkpoint-restore-finalize)

Nonvolatile memory
Stable values

ℓ_1	ℓ_2
0	0



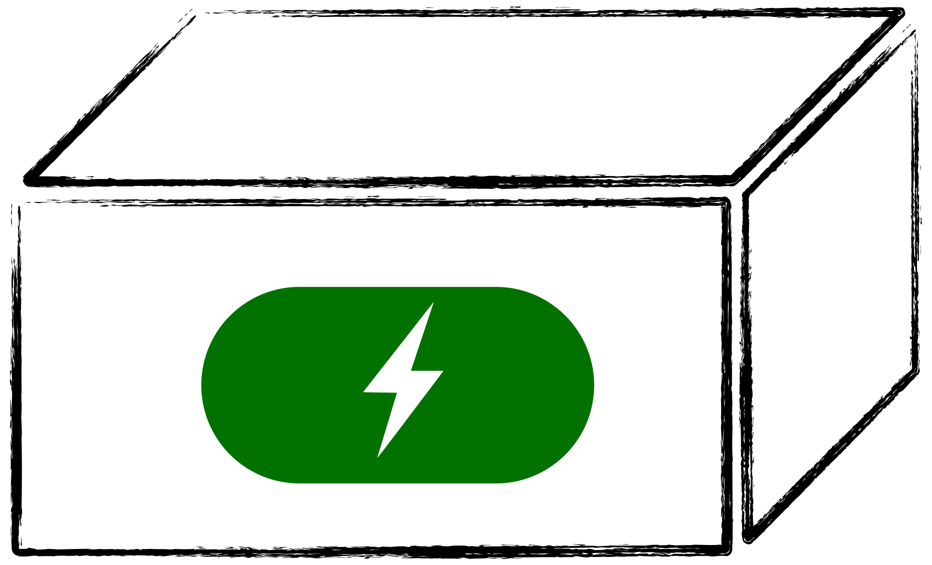
Volatile memory
Unstable values

ℓ_1^s	ℓ_2^s
2	2

Checkpoint

pc →

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```

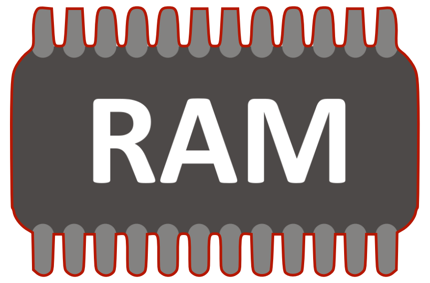


Final memory state we expect:

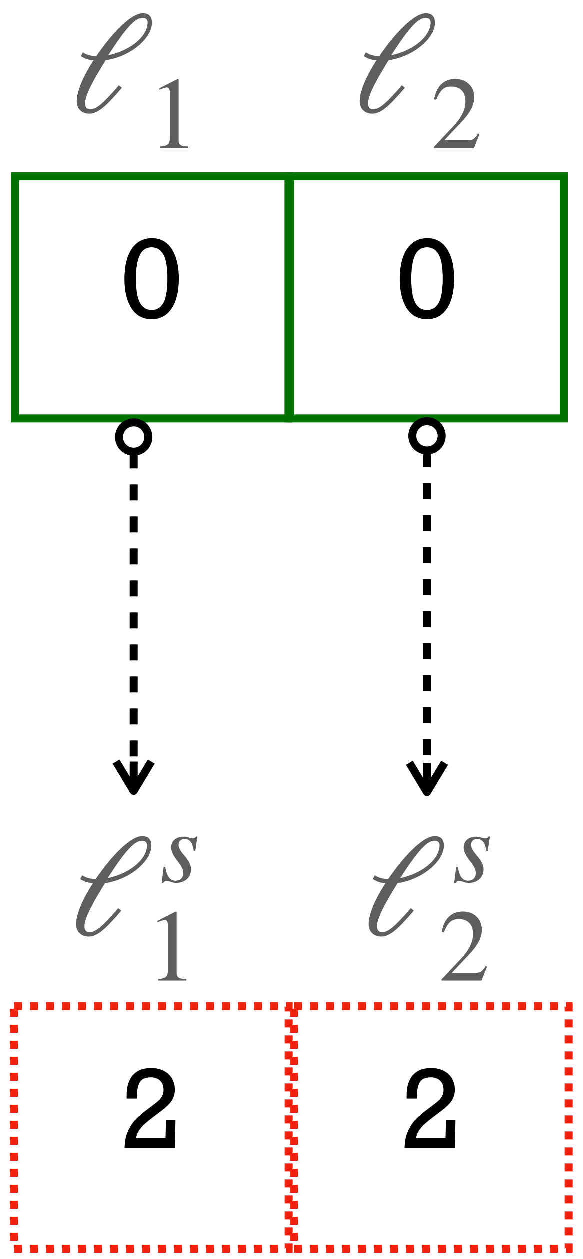
ℓ_1	ℓ_2
2	2

Solution: Checkpointing blocks (checkpoint-restore-finalize)

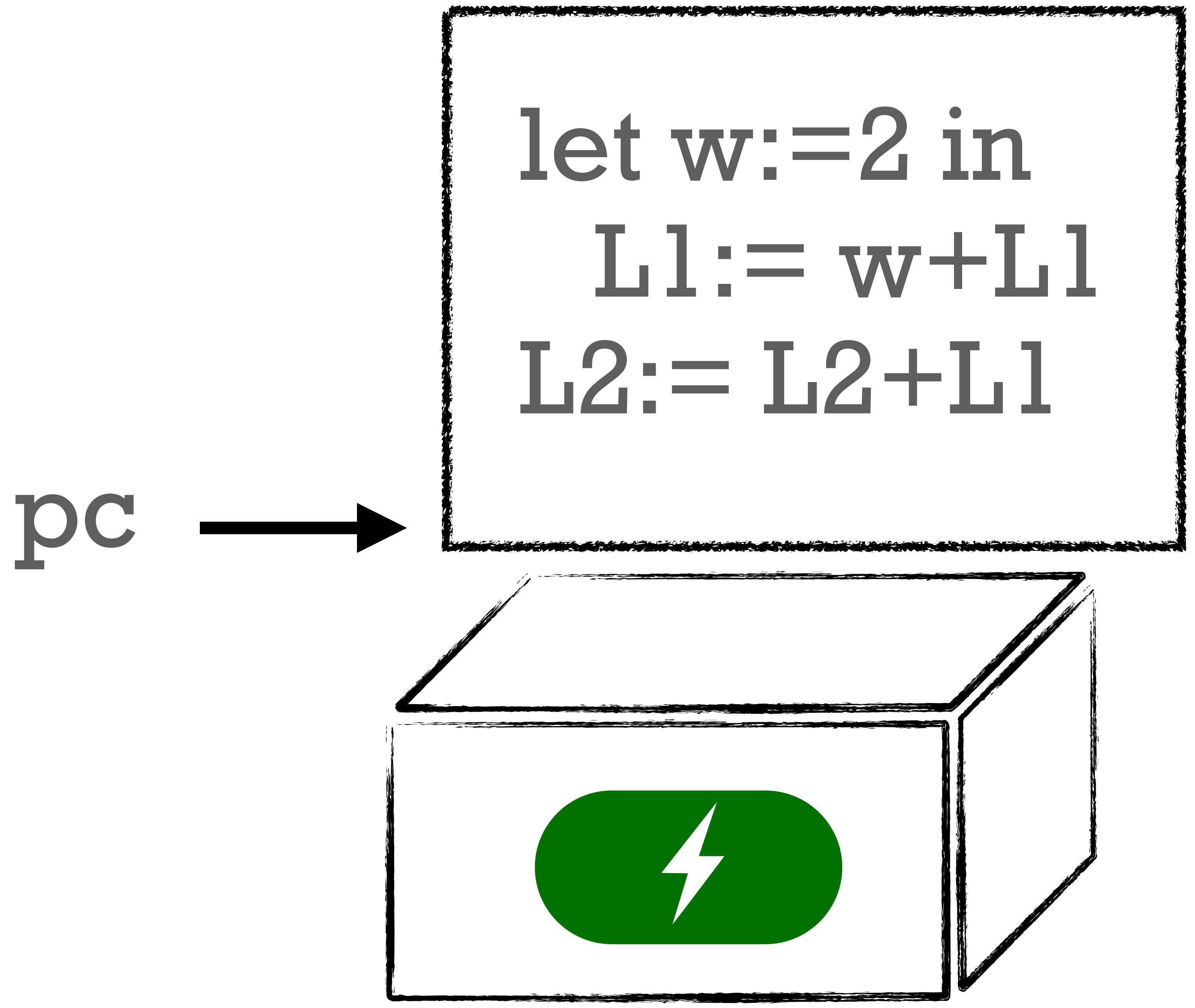
Nonvolatile memory
Stable values



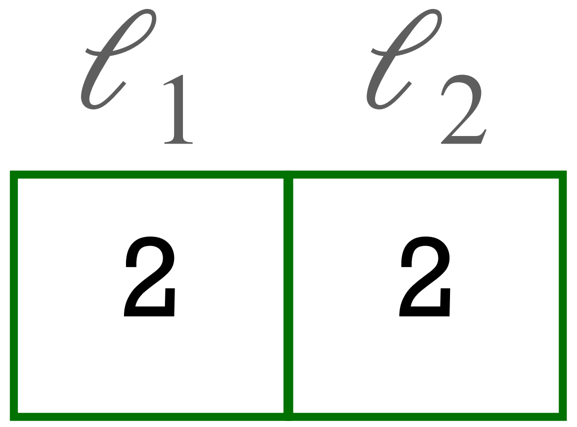
Volatile memory
Unstable values



Checkpoint



Final memory state we expect:

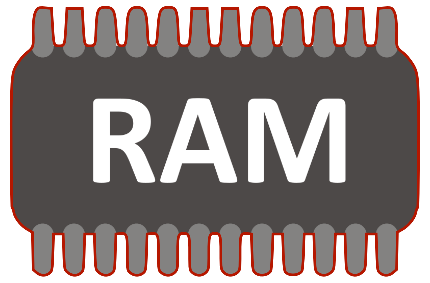


Solution: Checkpointing blocks (checkpoint-restore-finalize)

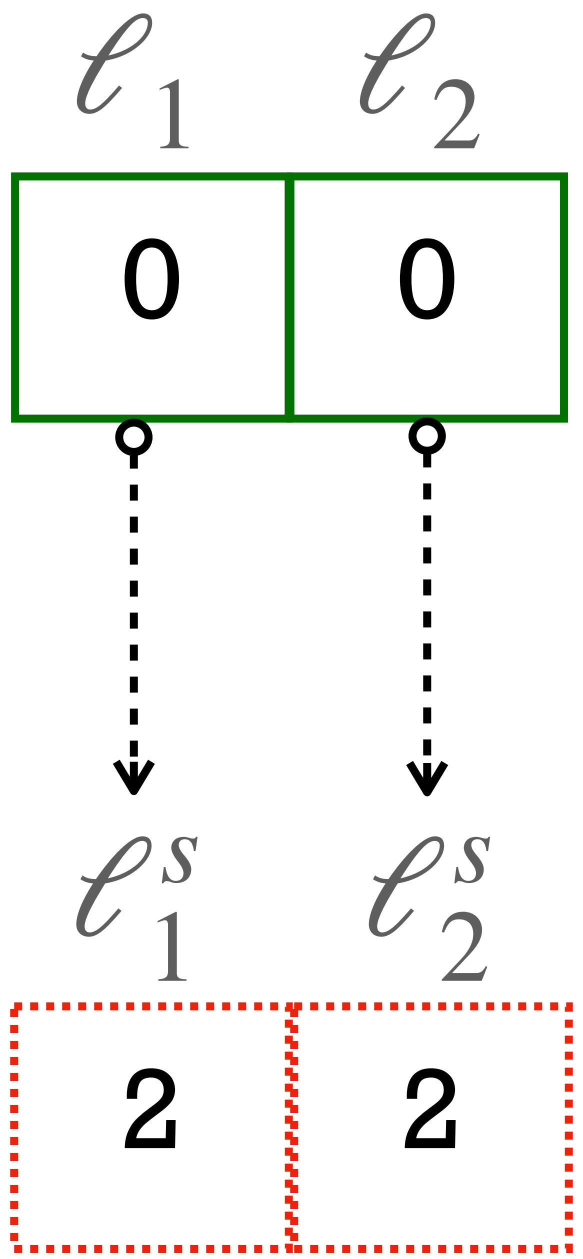
Finalize state

Checkpoint

Nonvolatile memory
Stable values

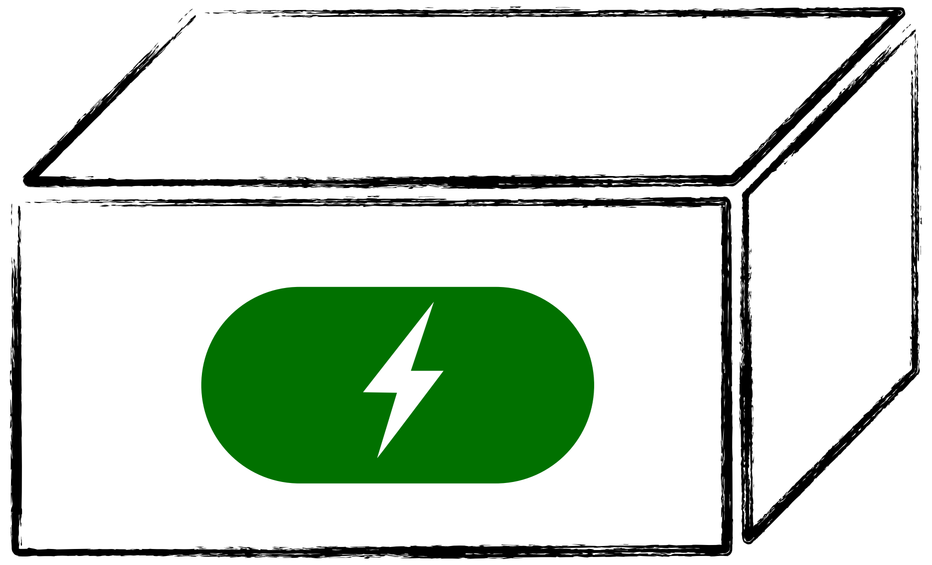


Volatile memory
Unstable values

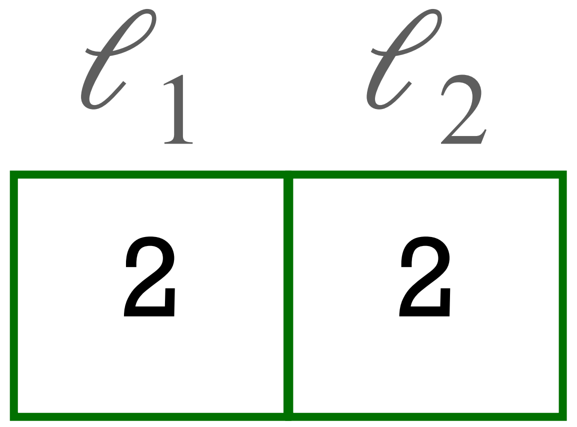



```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```

pc →



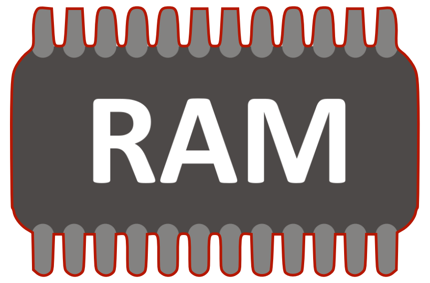
Final memory state we expect:



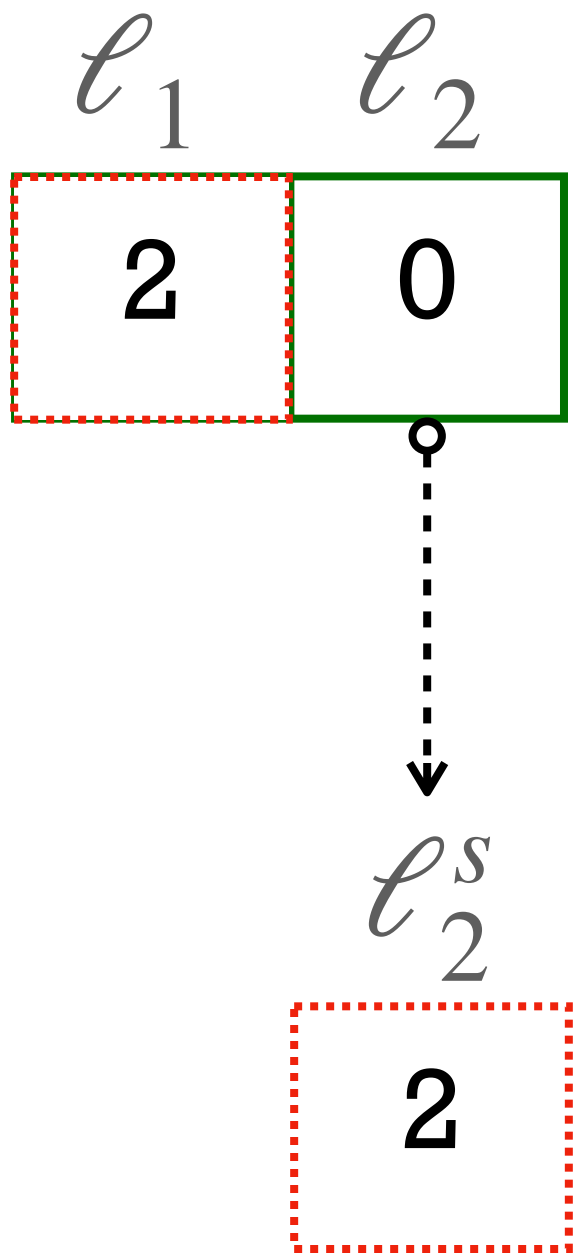
Solution: Checkpointing blocks (checkpoint-restore-finalize)

Finalize state

Nonvolatile memory
Stable values



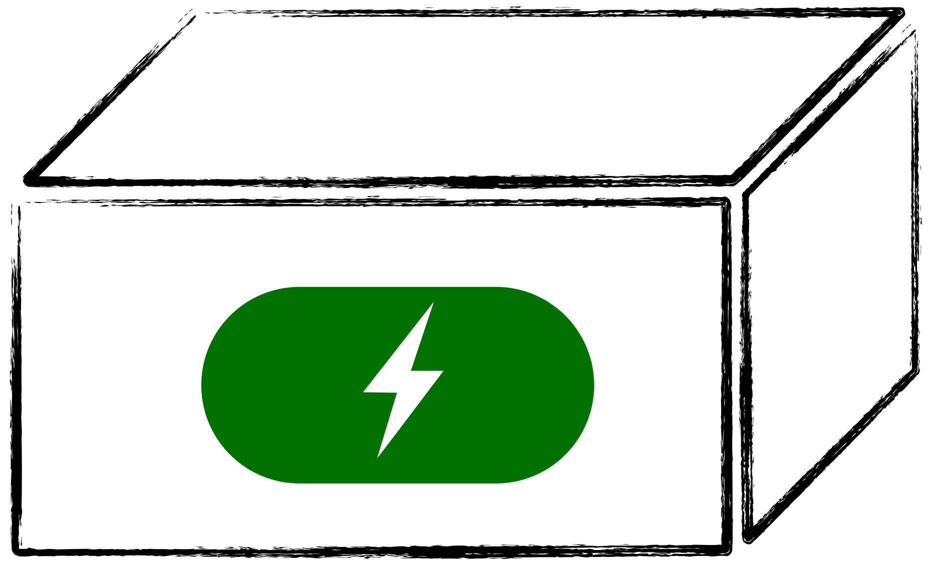
Volatile memory
Unstable values



Checkpoint

let w:=2 in
  L1:= w+L1
L2:= L2+L1

pc →



Final memory state we expect:

ℓ_1	ℓ_2
2	2

Solution: Checkpointing blocks (checkpoint-restore-finalize)

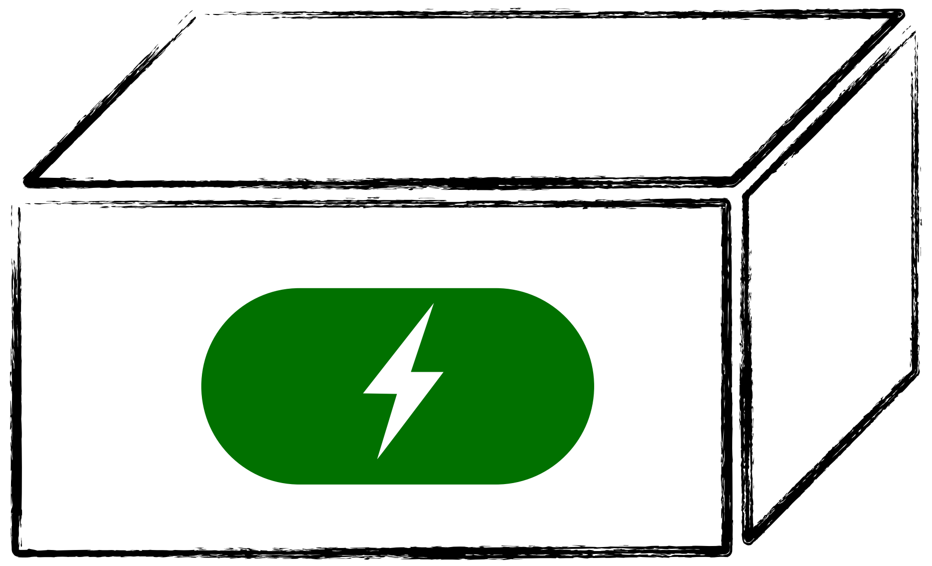
Finalize state

ℓ_1	ℓ_2
2	2

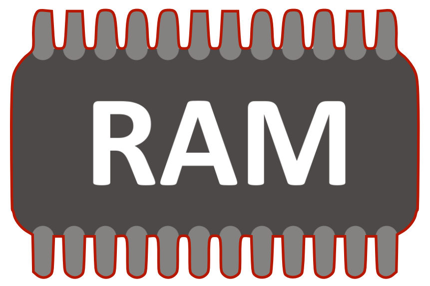
Checkpoint

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```

pc →



Nonvolatile memory
Stable values



Volatile memory
Unstable values

Final memory state we expect:

ℓ_1	ℓ_2
2	2

Solution: Checkpointing blocks (checkpoint-restore-finalize)

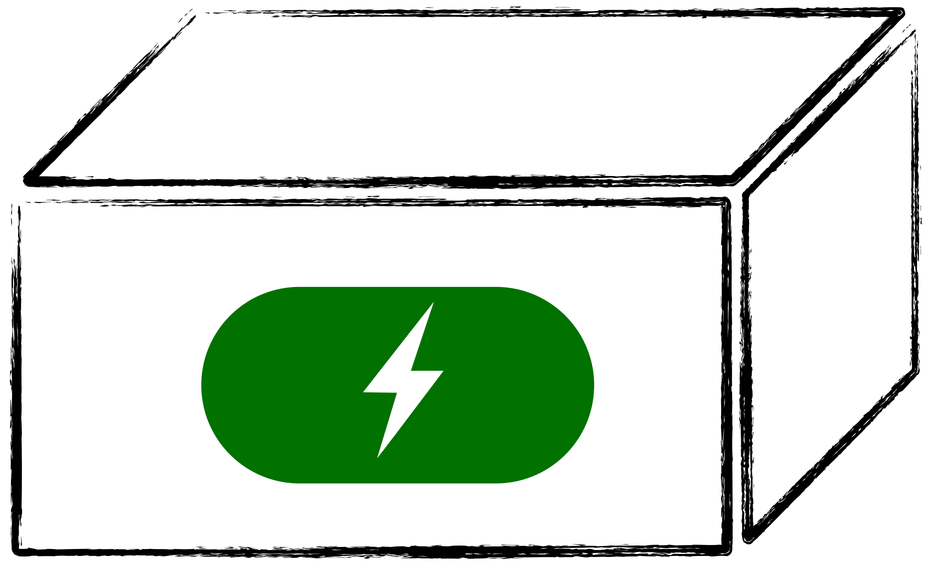
Finalize state

ℓ_1	ℓ_2
2	2

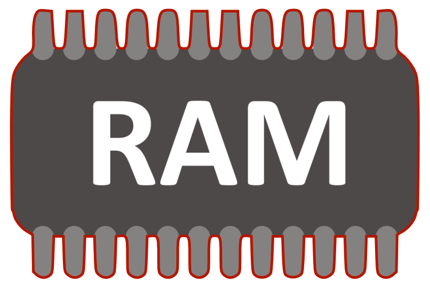
Checkpoint

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```

pc →



Nonvolatile memory
Stable values



Volatile memory
Unstable values

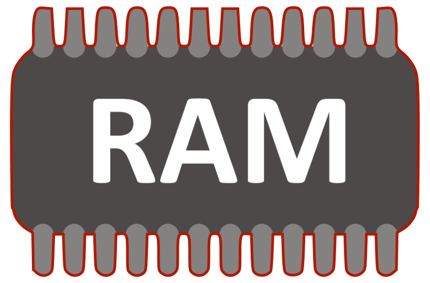
Final memory state we expect:

ℓ_1	ℓ_2
2	2

Solution: Checkpointing blocks (checkpoint-restore-finalize)

Nonvolatile memory
Stable values

ℓ_1	ℓ_2
2	2

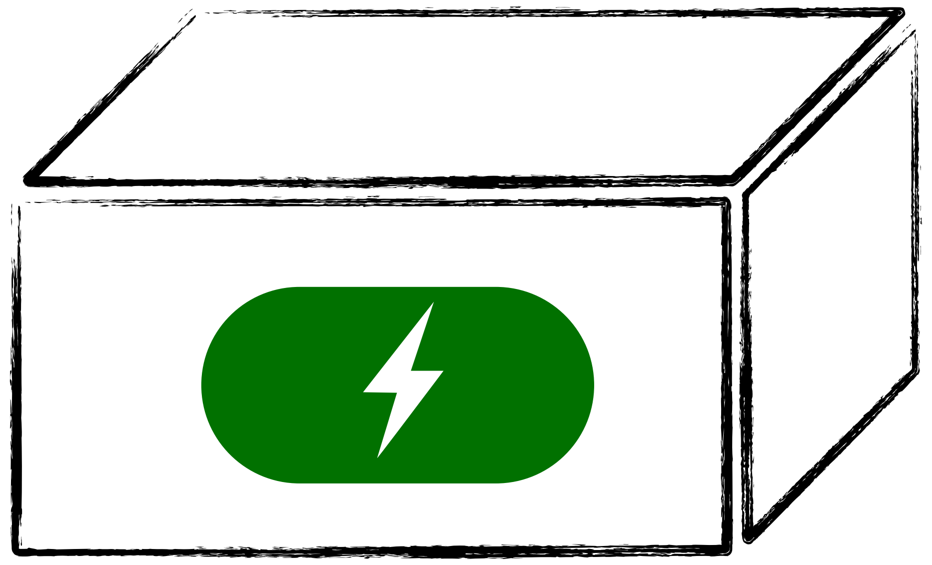


Volatile memory
Unstable values

Checkpoint

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```

pc →



Final memory state we expect:

ℓ_1	ℓ_2
2	2

Solution: Checkpointing blocks (checkpoint-restore-finalize)

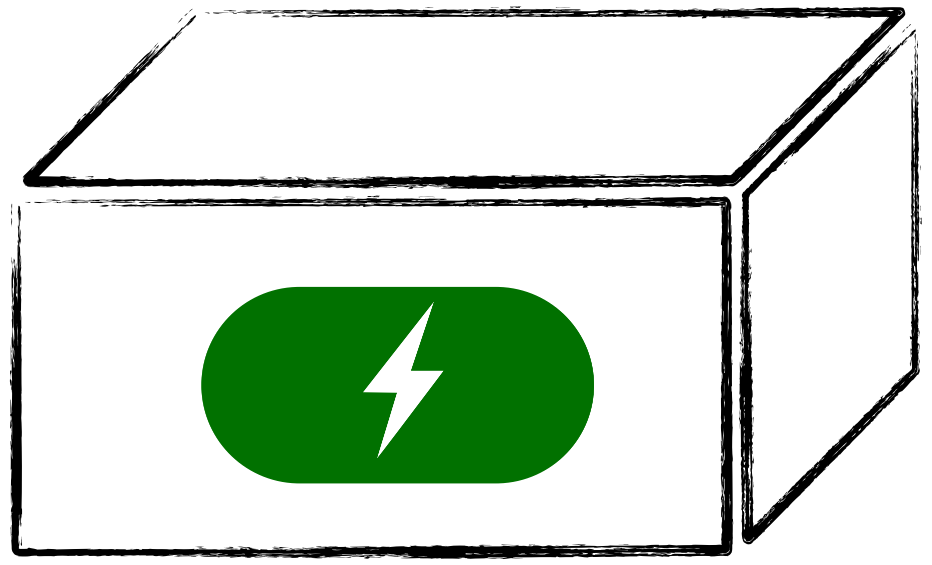
Nonvolatile memory
Stable values

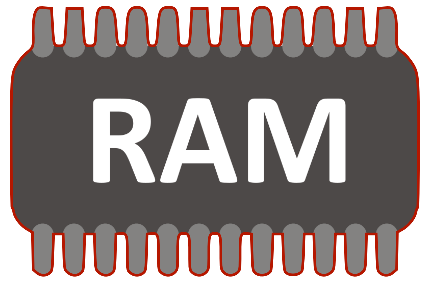
ℓ_1	ℓ_2
2	2

Checkpoint

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```

pc →





Volatile memory
Unstable values

Final memory state we expect:

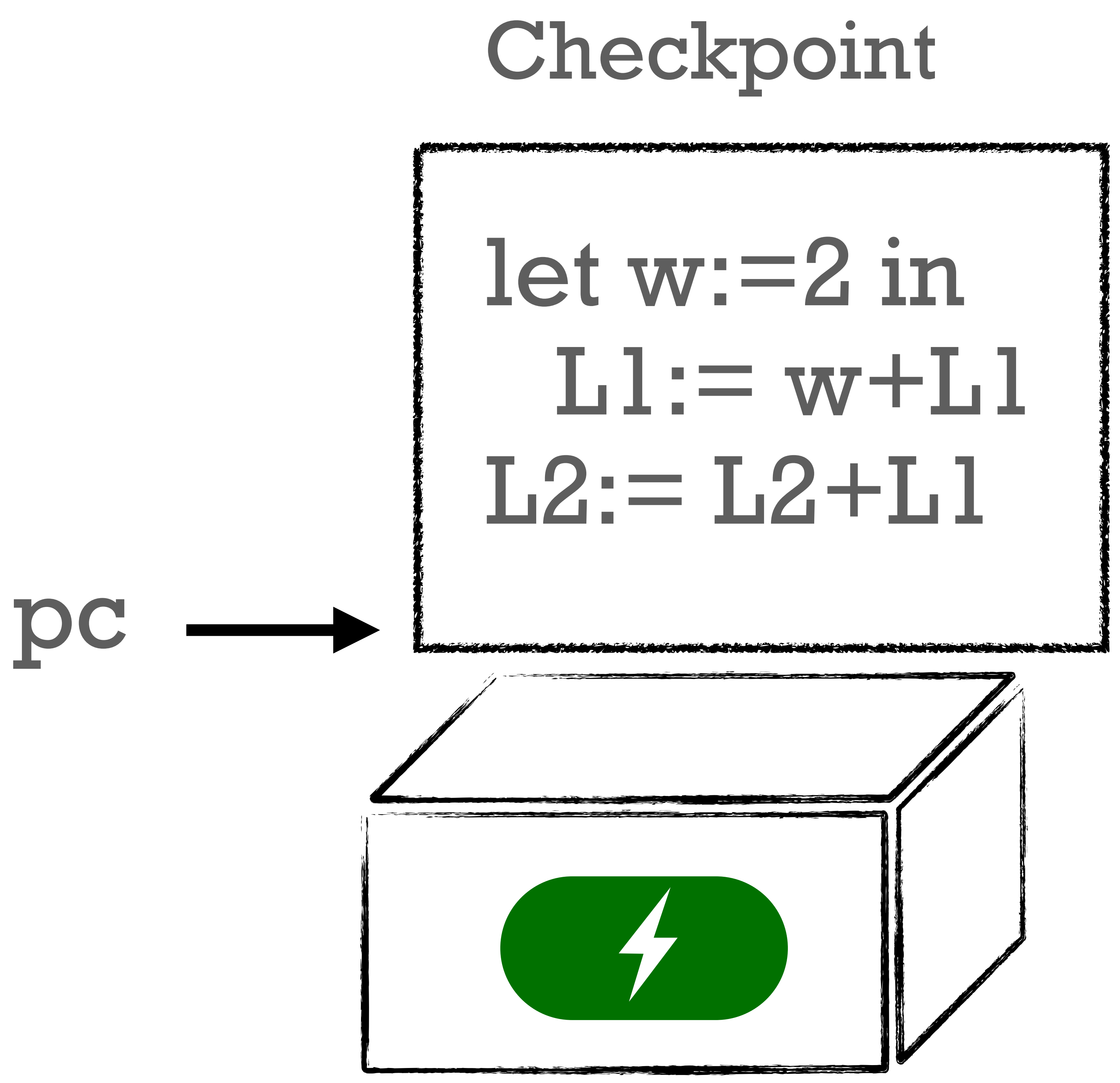
ℓ_1	ℓ_2
2	2

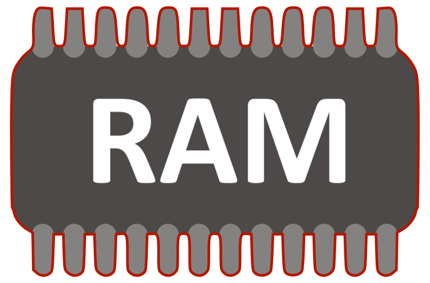
Correct final state:

Solution: Checkpointing blocks (checkpoint-restore-finalize)

Nonvolatile memory
Stable values

ℓ_1	ℓ_2
2	2





Volatile memory
Unstable values

Final memory state we expect:

ℓ_1	ℓ_2
2	2

Correct final state:
equivalent to the continuous execution.

Intermittent execution in energy harvesting devices

Intermittent execution in energy harvesting devices

Prior work:

Practical solutions based on *checkpointing*, *restoring*, and *finalizing state*.
(Surbatovich et al. OOPSLA 2019, OOPSLA 2020, PLDI 2021, PLDI2023),

Intermittent execution in energy harvesting devices

Prior work:

Practical solutions based on *checkpointing*, *restoring*, and *finalizing state*.
(Surbatovich et al. OOPSLA 2019, OOPSLA 2020, PLDI 2021, PLDI2023),

This work:

Fundamental logical underpinning of these operations.

Intermittent execution in energy harvesting devices

Prior work:

Practical solutions based on *checkpointing*, *restoring*, and *finalizing state*.
(Surbatovich et al. OOPSLA 2019, OOPSLA 2020, PLDI 2021, PLDI2023),

This work:

Fundamental logical underpinning of these operations.

- Crash types based on adjoint logic
- A type system to rule out incorrect intermittent executions
- A logical relation to prove that all well-typed programs are correct

Outline

- A type system based on adjoint modalities and independence principle
- Correctness as a logical relation
- Conclusion

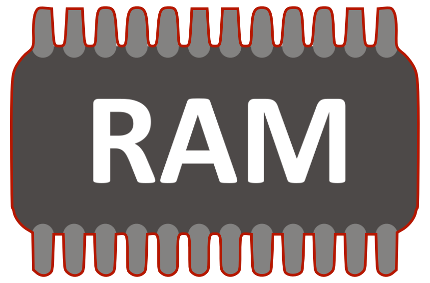
A type system based on adjoint logic

Nonvolatile memory
Stable values

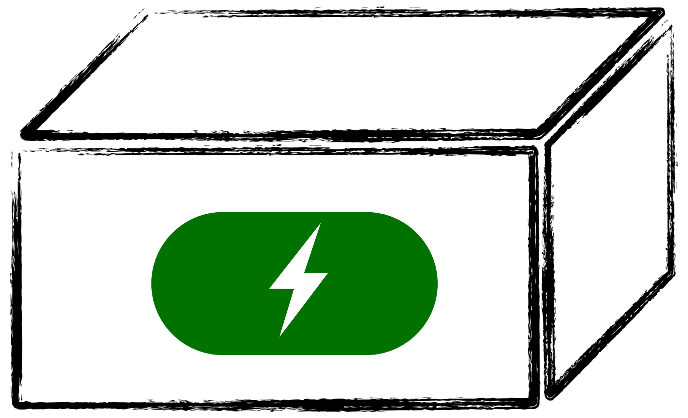
ℓ_1	ℓ_2
0	0

Checkpoint

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```



Volatile memory
Unstable values



A type system based on adjoint logic

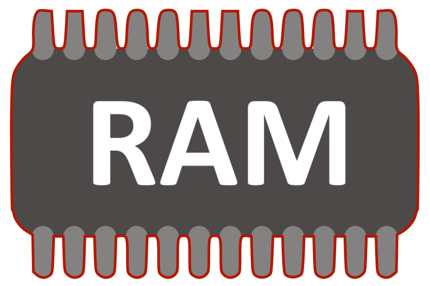
Nonvolatile memory
Stable values
Int

ℓ_1 ℓ_2

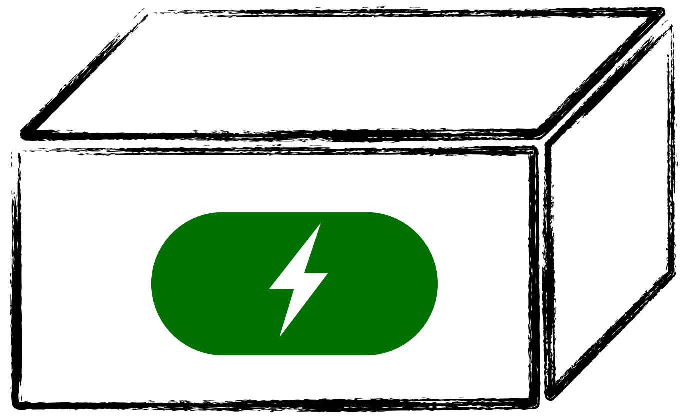
ℓ_1	ℓ_2
0	0

Checkpoint

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```



Volatile memory
Unstable values



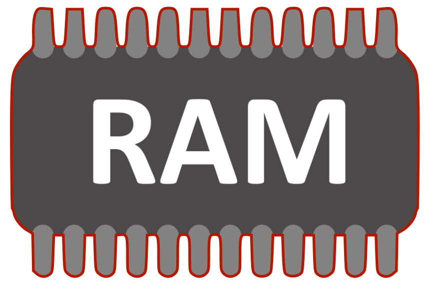
A type system based on adjoint logic

Nonvolatile memory
Stable values
$\uparrow$Int

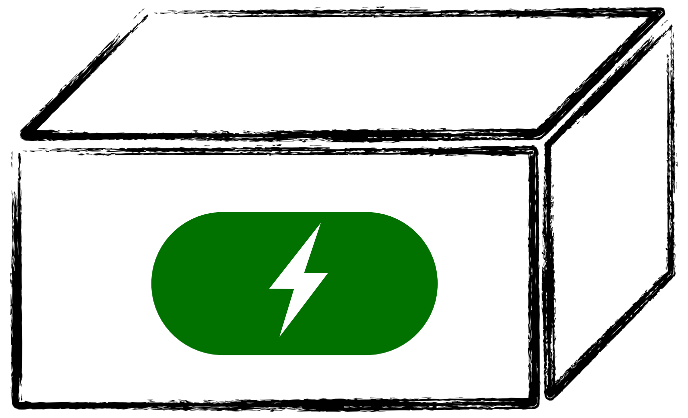
ℓ_1	ℓ_2
0	0

Checkpoint

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

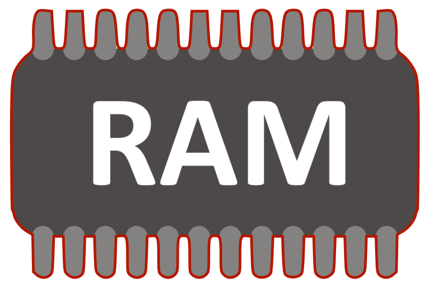


Volatile memory
Unstable values

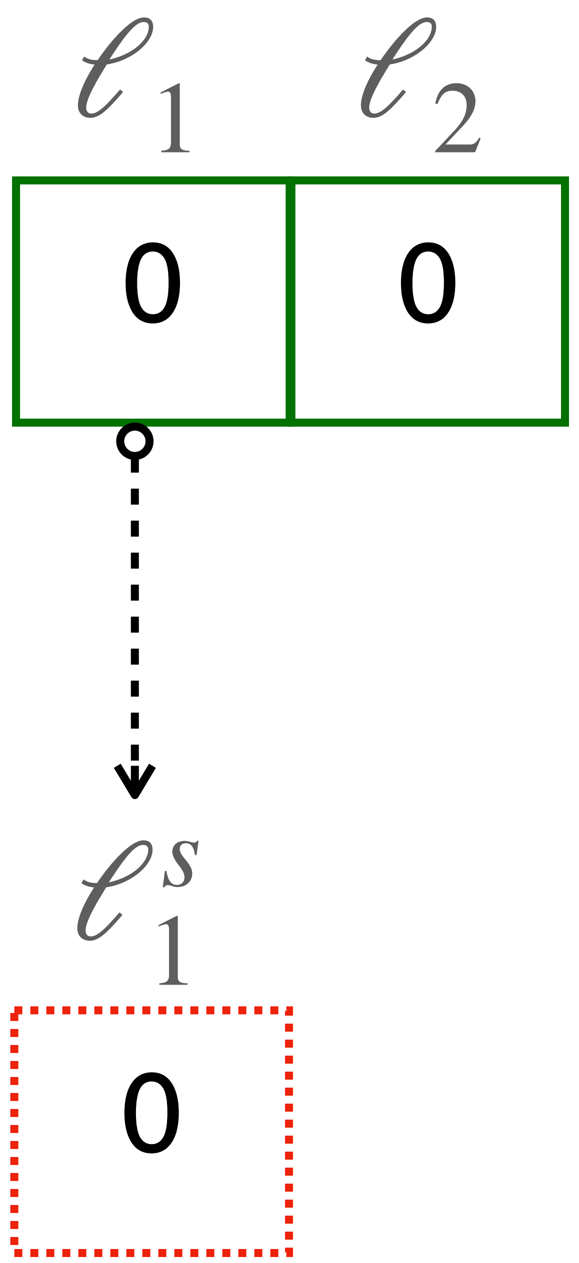


A type system based on adjoint logic

Nonvolatile memory
Stable values
$\uparrow$Int

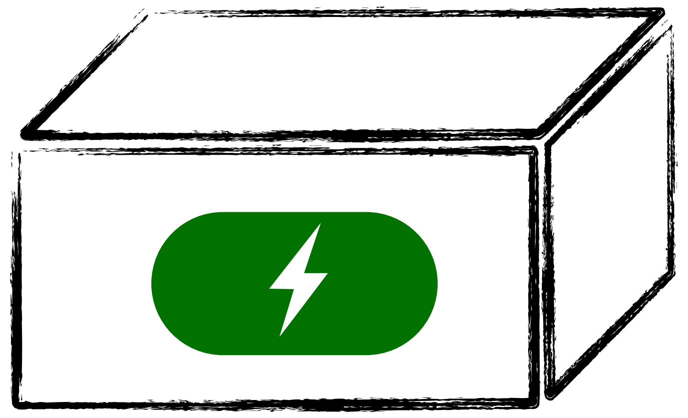


Volatile memory
Unstable values



Checkpoint

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

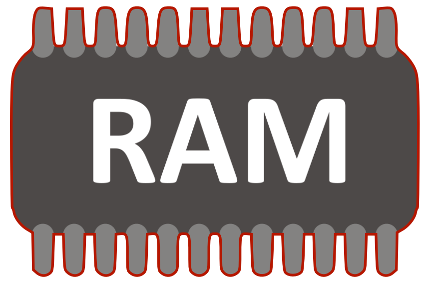


A type system based on adjoint logic

Nonvolatile memory

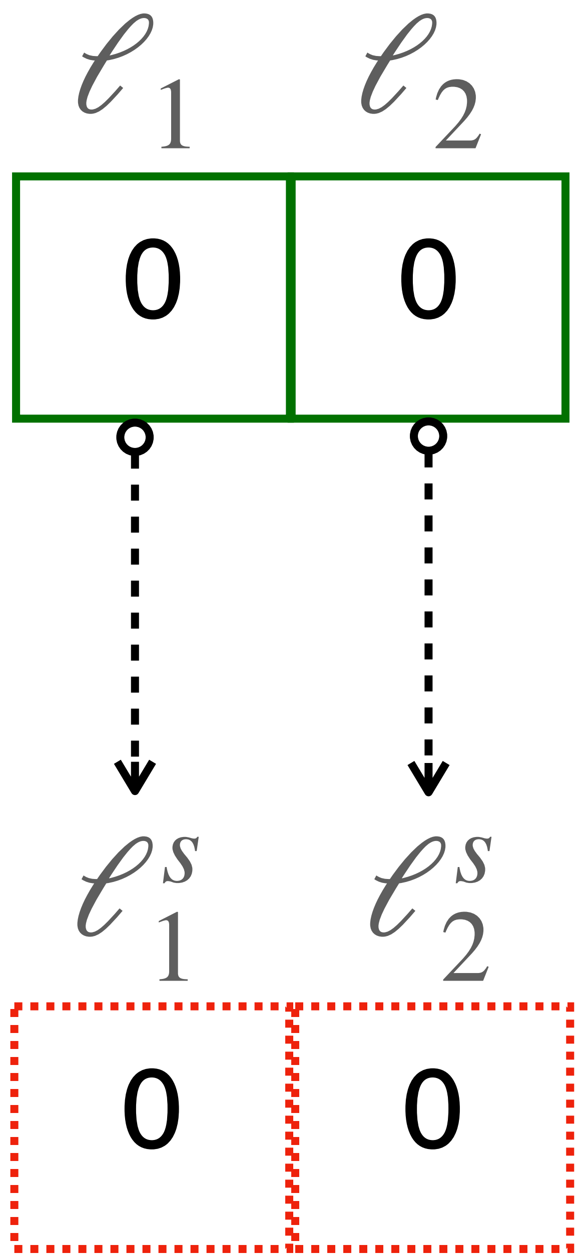
Stable values

$\uparrow$Int



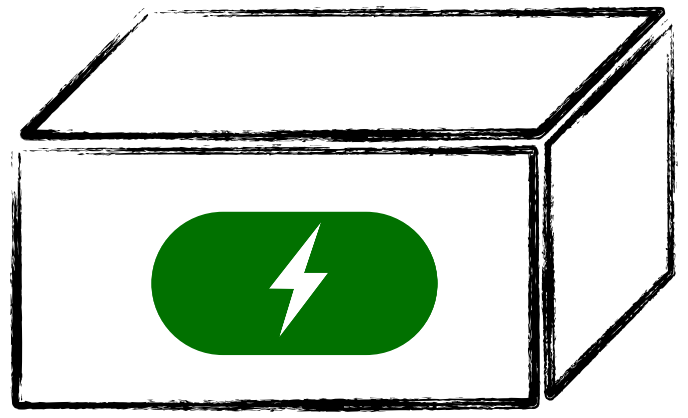
Volatile memory

Unstable values



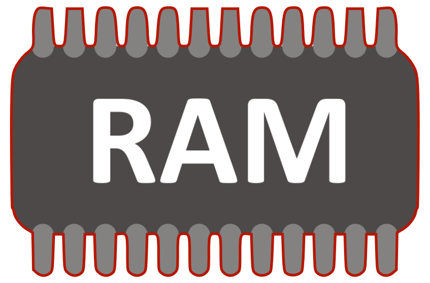
Checkpoint

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

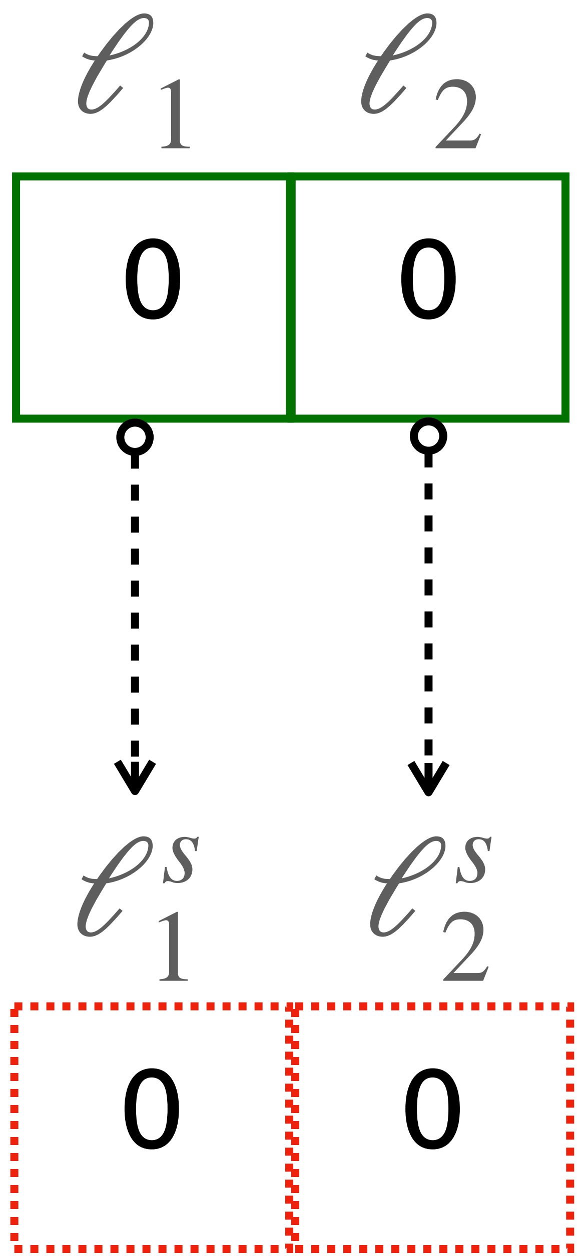


A type system based on adjoint logic

Nonvolatile memory
Stable values
$\uparrow$Int

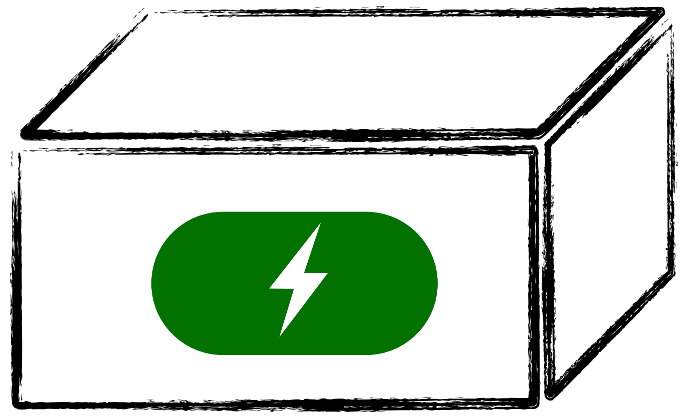


Volatile memory
Unstable values
$\uparrow$Int



Checkpoint

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

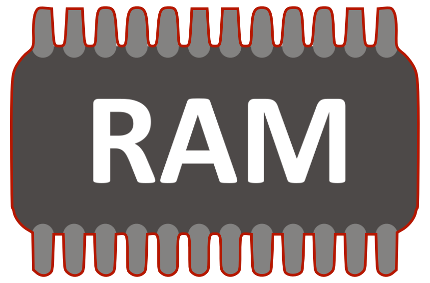


A type system based on adjoint logic

Nonvolatile memory

Stable values

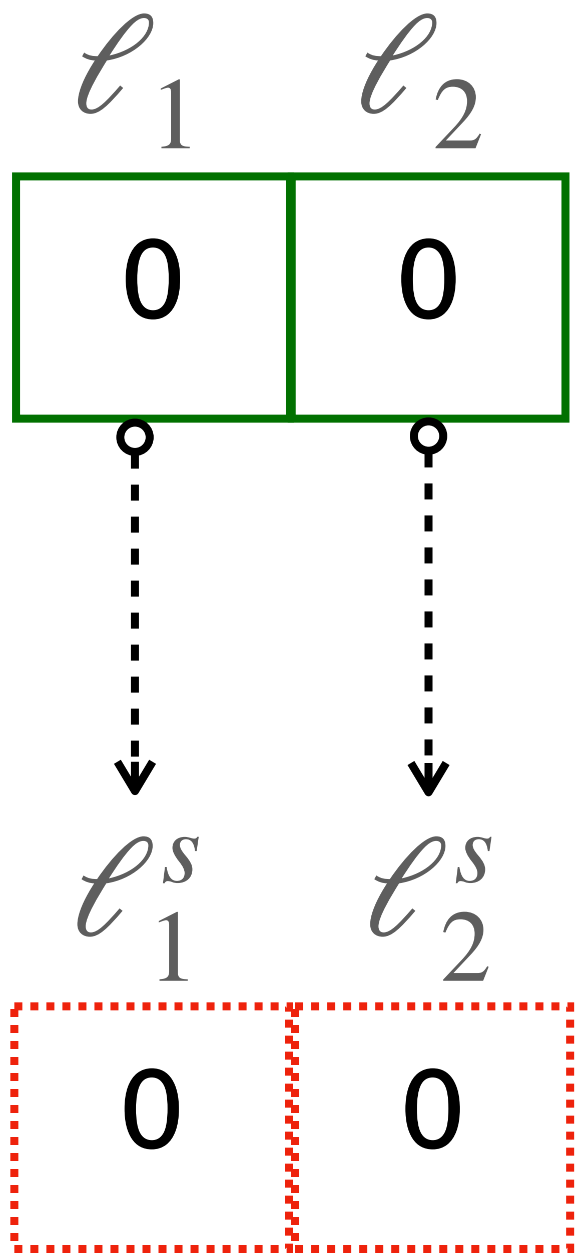
$\uparrow$Int



Volatile memory

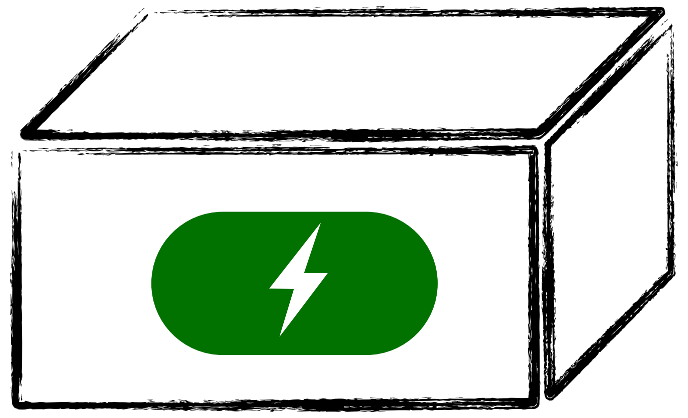
Unstable values

$\downarrow\uparrow$Int



Checkpoint

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```



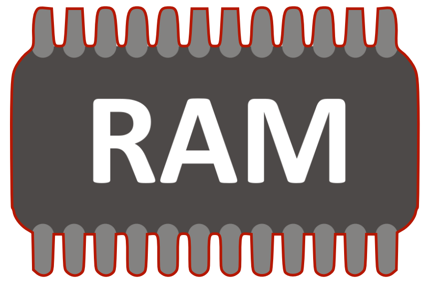
A type system based on adjoint logic

Stable types	$\tau_s := \uparrow \tau_u \mid \mathtt{nat} \rightsquigarrow \tau_s$
Unstable types	$\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid v_t$
Basic types	$T := A \mid unit$
Store types	$A := int \mid bool$

Nonvolatile memory

Stable values

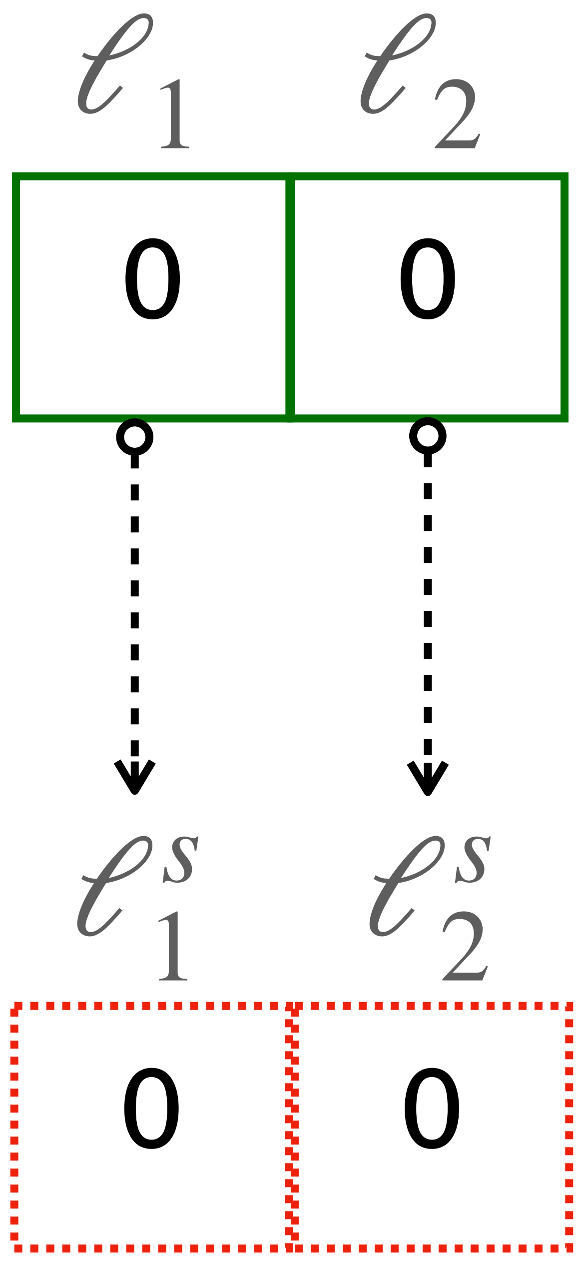
$\uparrow$Int



Volatile memory

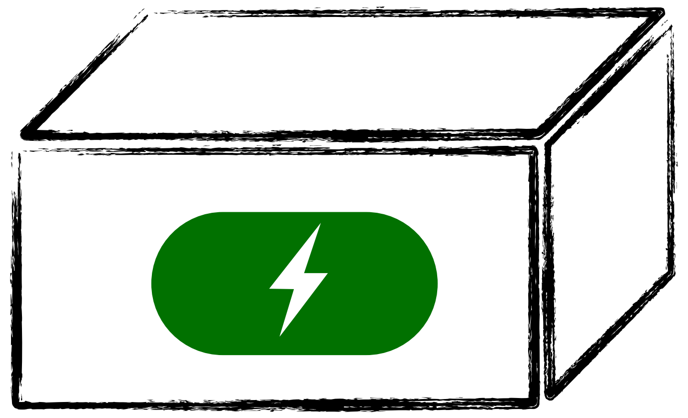
Unstable values

$\downarrow\uparrow$Int



Checkpoint

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```



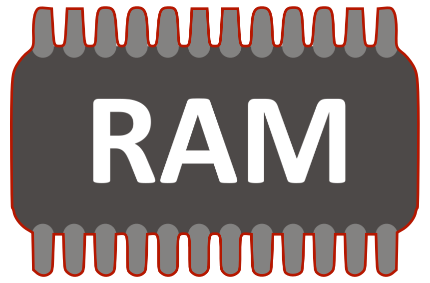
A type system based on adjoint logic

Stable types	$\tau_s := \uparrow \tau_u \mid \mathtt{nat} \rightsquigarrow \tau_s$
Unstable types	$\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid v_t$
Basic types	$T := A \mid unit$
Store types	$A := int \mid bool$

Nonvolatile memory

Stable values

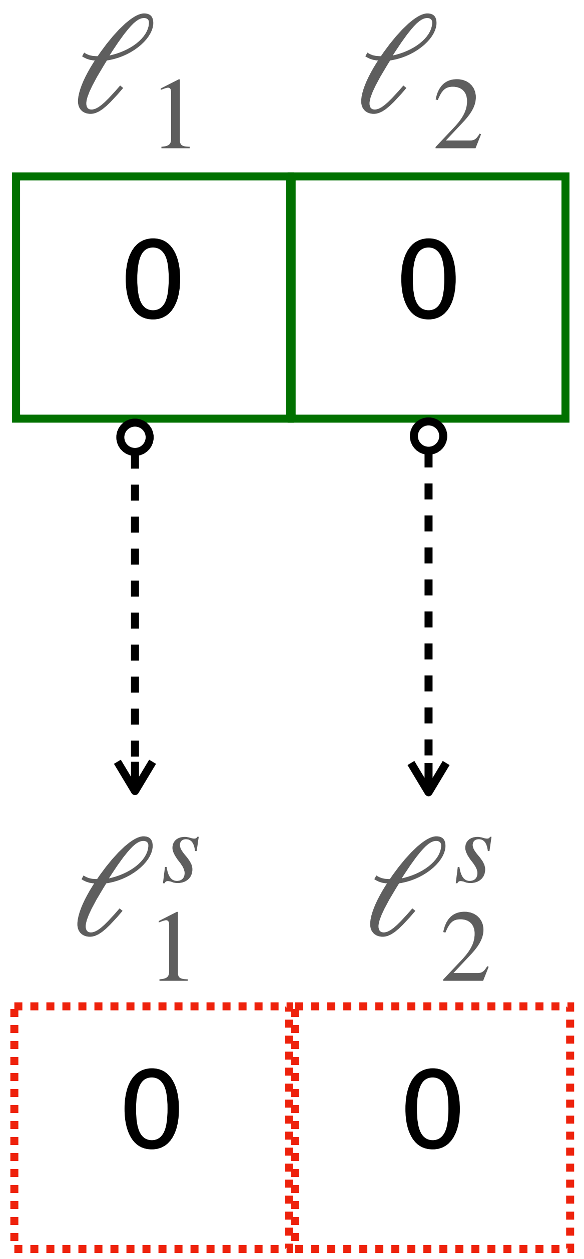
$\uparrow$Int



Volatile memory

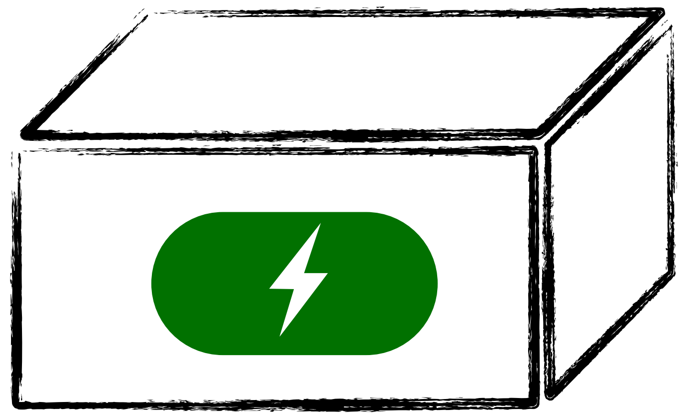
Unstable values

$\downarrow\uparrow$Int



Checkpoint

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

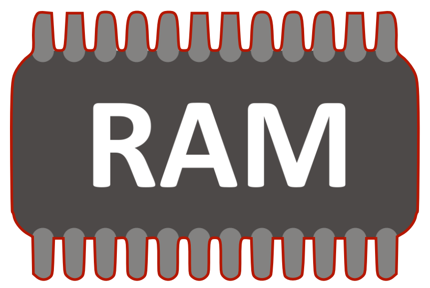


Computation return type

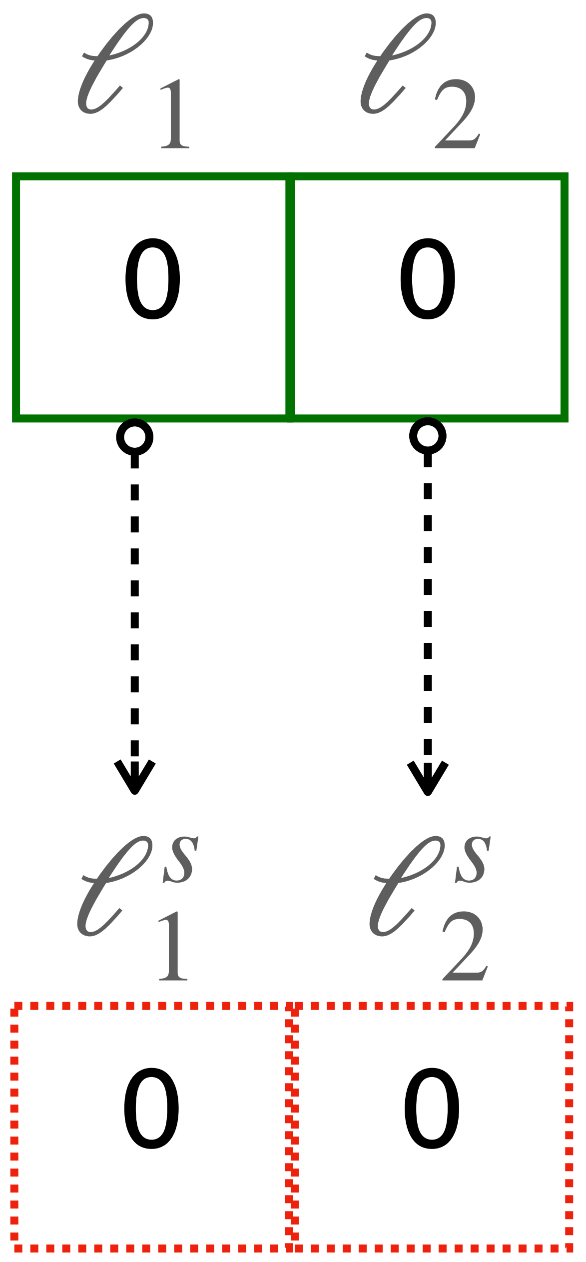
A type system based on adjoint logic

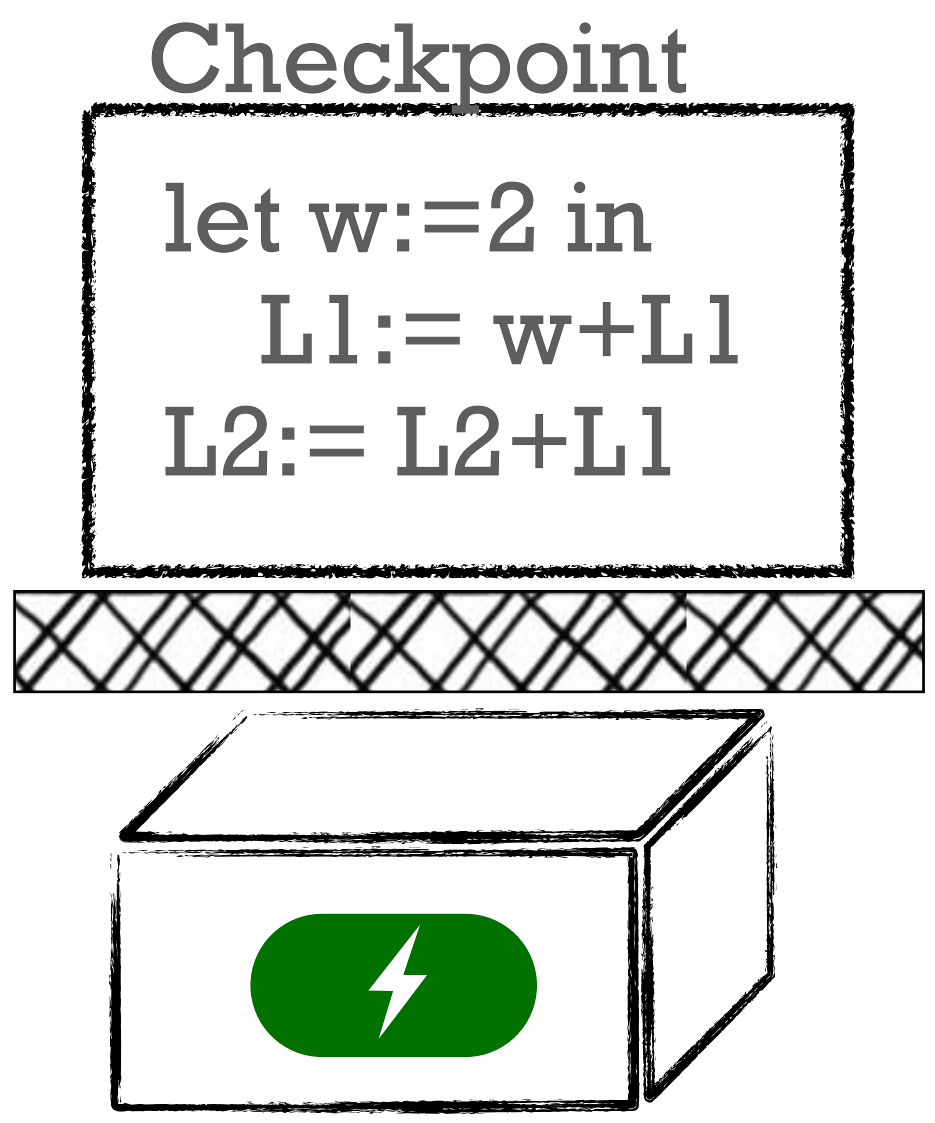
Stable types	$\tau_s := \uparrow \tau_u \mid \texttt{nat} \rightsquigarrow \tau_s$
Unstable types	$\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid \nu_t$
Basic types	$T := A \mid \mathit{unit}$
Store types	$A := \mathit{int} \mid \mathit{bool}$

Nonvolatile memory
Stable values
$\uparrow$Int



Volatile memory
Unstable values
$\downarrow\uparrow$Int





Computation return type

τ_s Stable computation

A type system based on adjoint logic

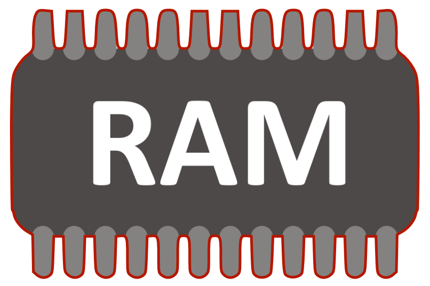
Stable types $\tau_s := \uparrow \tau_u \mid \texttt{nat} \rightsquigarrow \tau_s$

Unstable types $\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid \nu_t$

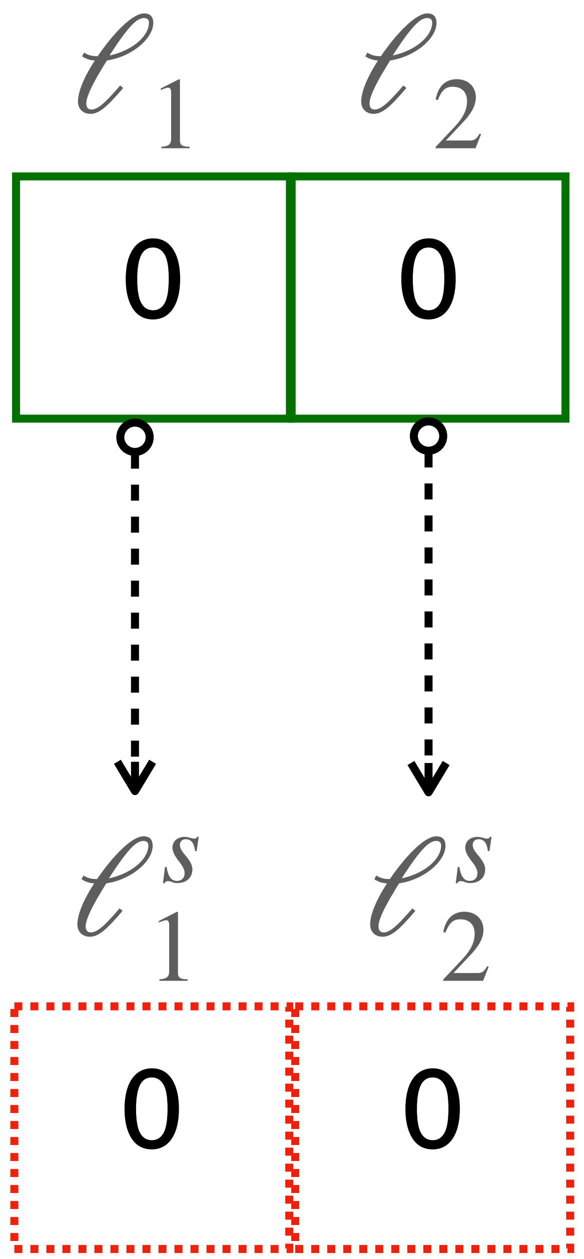
Basic types $T := A \mid unit$

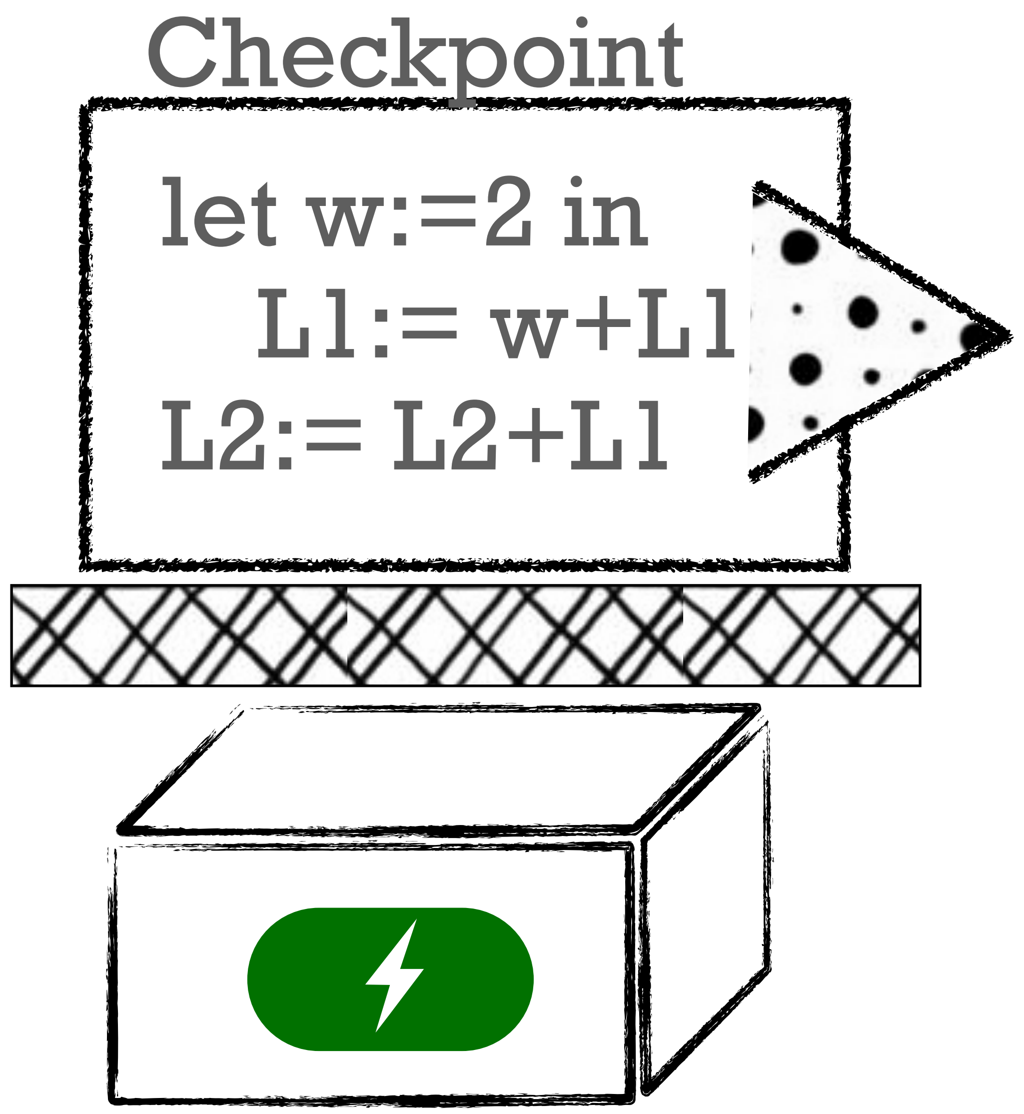
Store types $A := int \mid bool$

Nonvolatile memory
Stable values
$\uparrow$Int



Volatile memory
Unstable values
$\downarrow\uparrow$Int





Computation return type

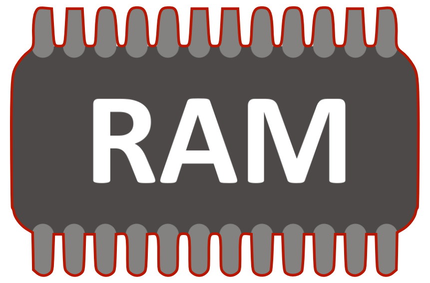
τ_u Unstable computation

τ_s Stable computation

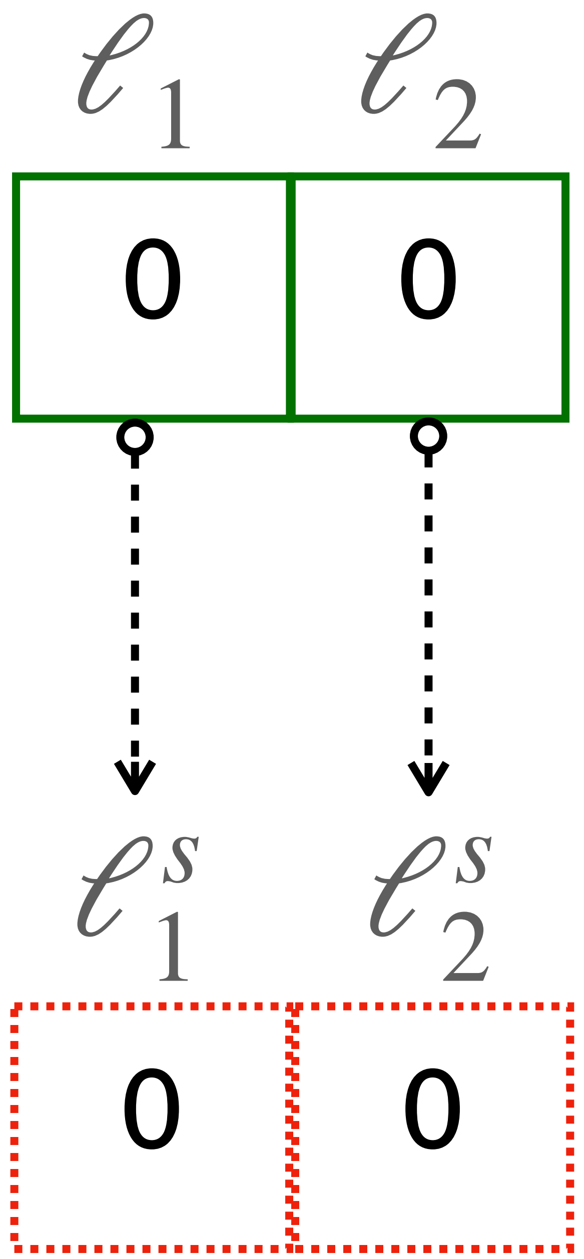
A type system based on adjoint logic

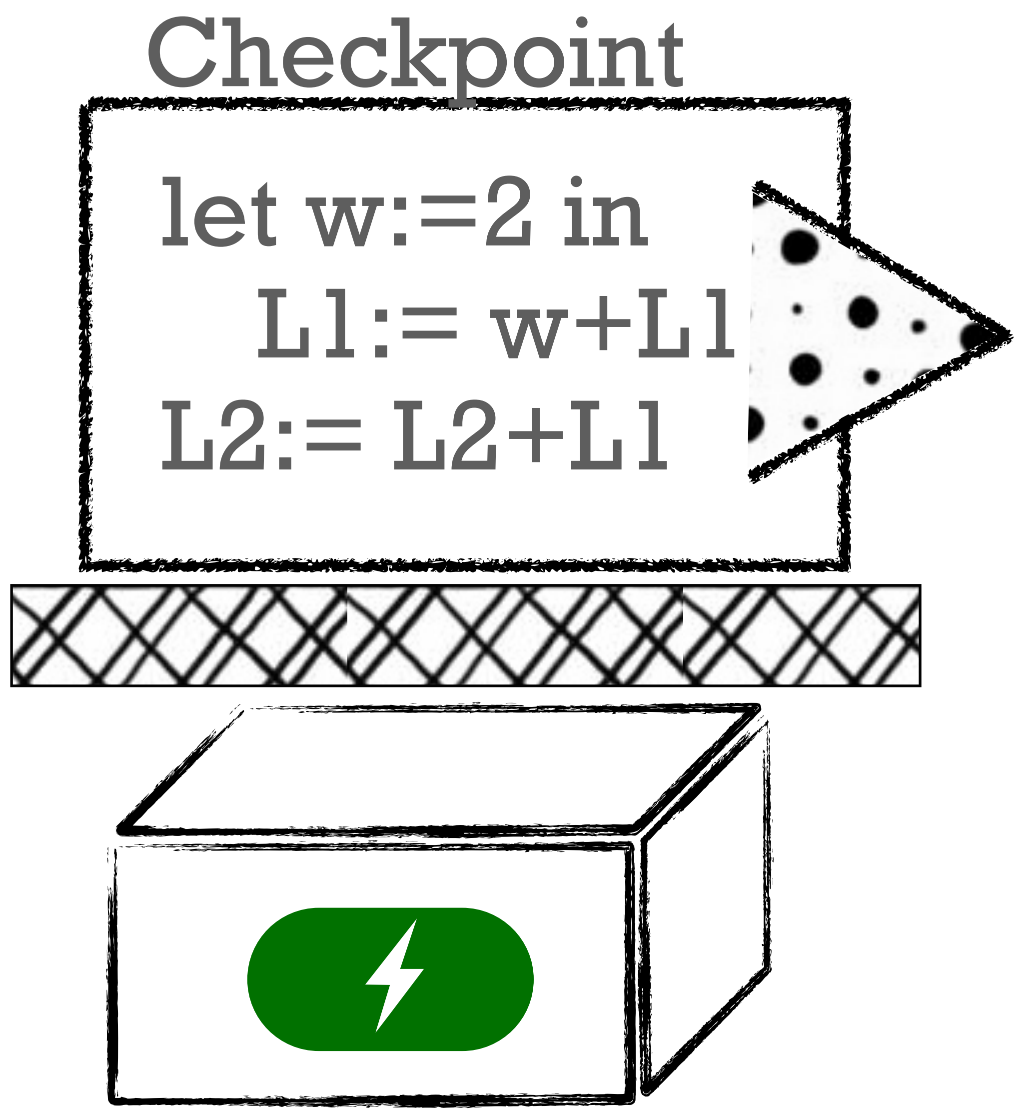
Stable types	$\tau_s := \uparrow \tau_u \mid \texttt{nat} \rightsquigarrow \tau_s$
Unstable types	$\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid v_t$
Basic types	$T := A \mid unit$
Store types	$A := int \mid bool$

Nonvolatile memory
Stable values
$\uparrow$Int



Volatile memory
Unstable values
$\downarrow\uparrow$Int





Computation return type

C — Unstable computation

τ_s — Stable computation

A type system based on adjoint logic

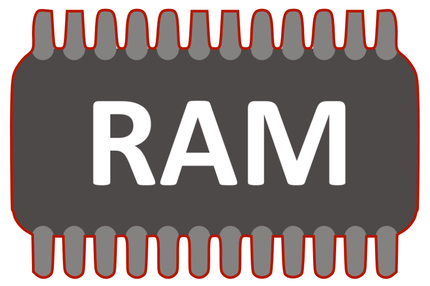
Stable types $\tau_s := \uparrow \tau_u \mid \mathtt{nat} \rightsquigarrow \tau_s$

Unstable types $\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid v_t$

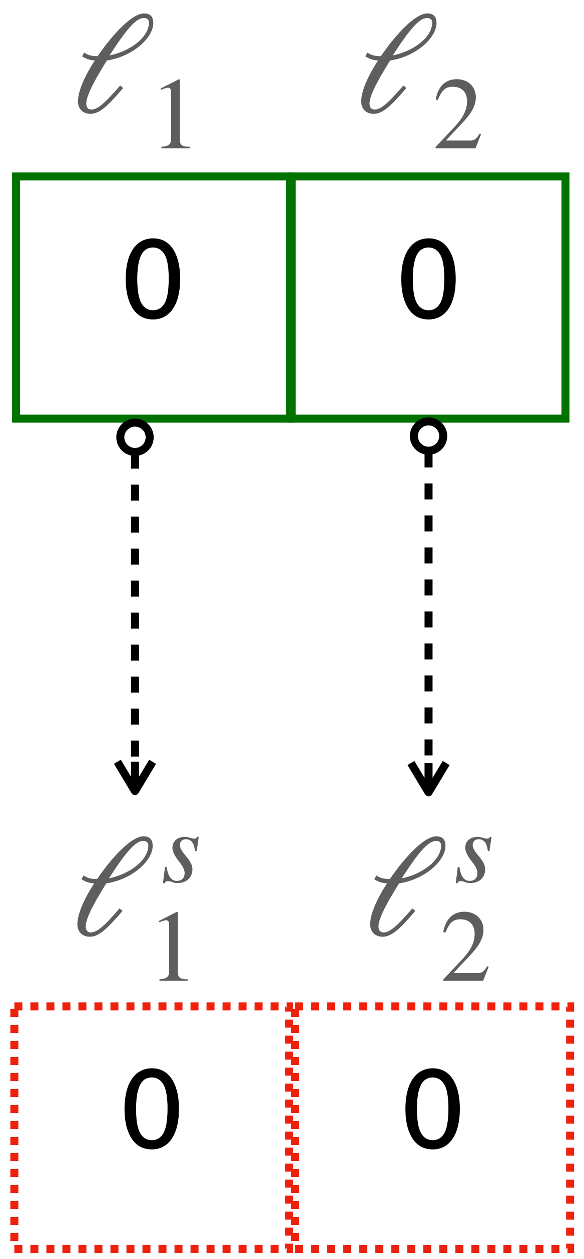
Basic types $T := A \mid unit$

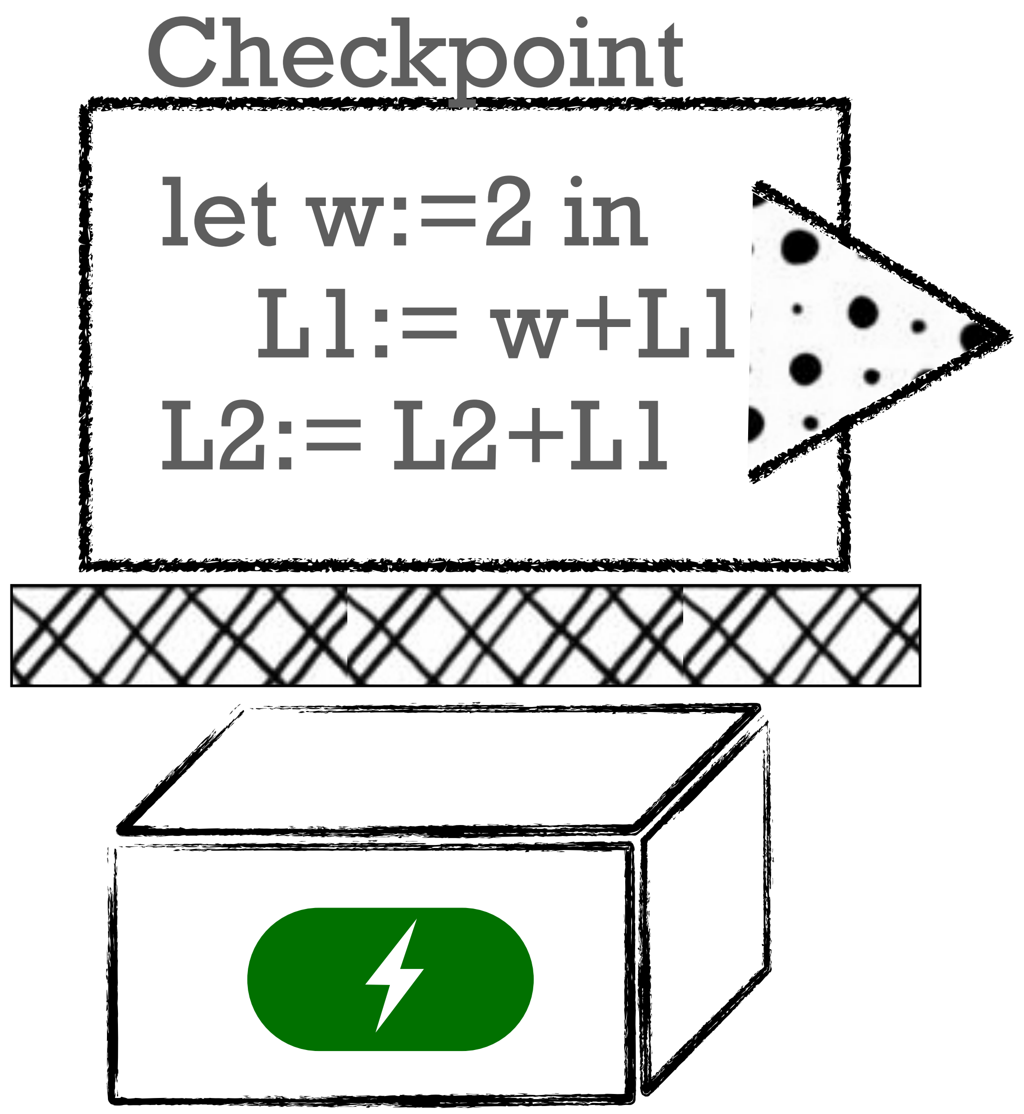
Store types $A := int \mid bool$

Nonvolatile memory
Stable values
$\uparrow$Int



Volatile memory
Unstable values
$\downarrow\uparrow$Int





Computation return type

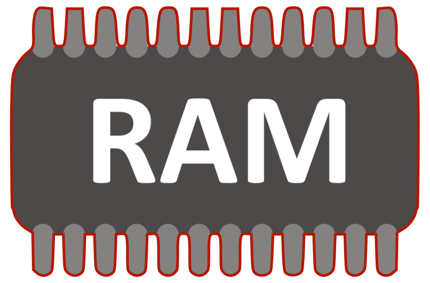
C Unstable computation

$\uparrow$C Stable computation

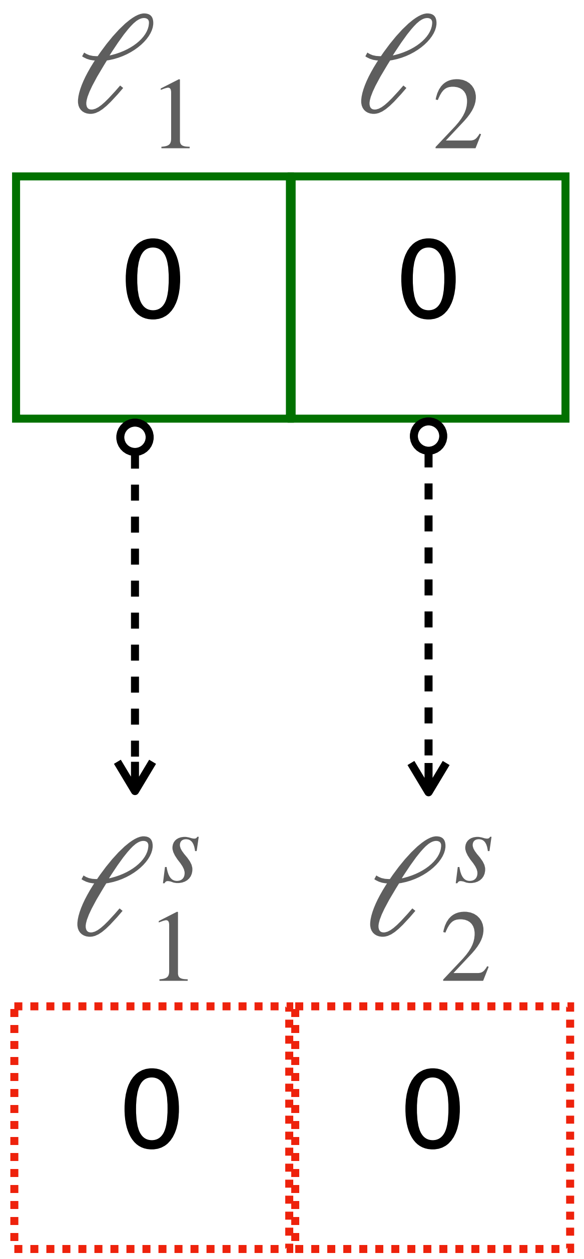
A type system based on adjoint logic

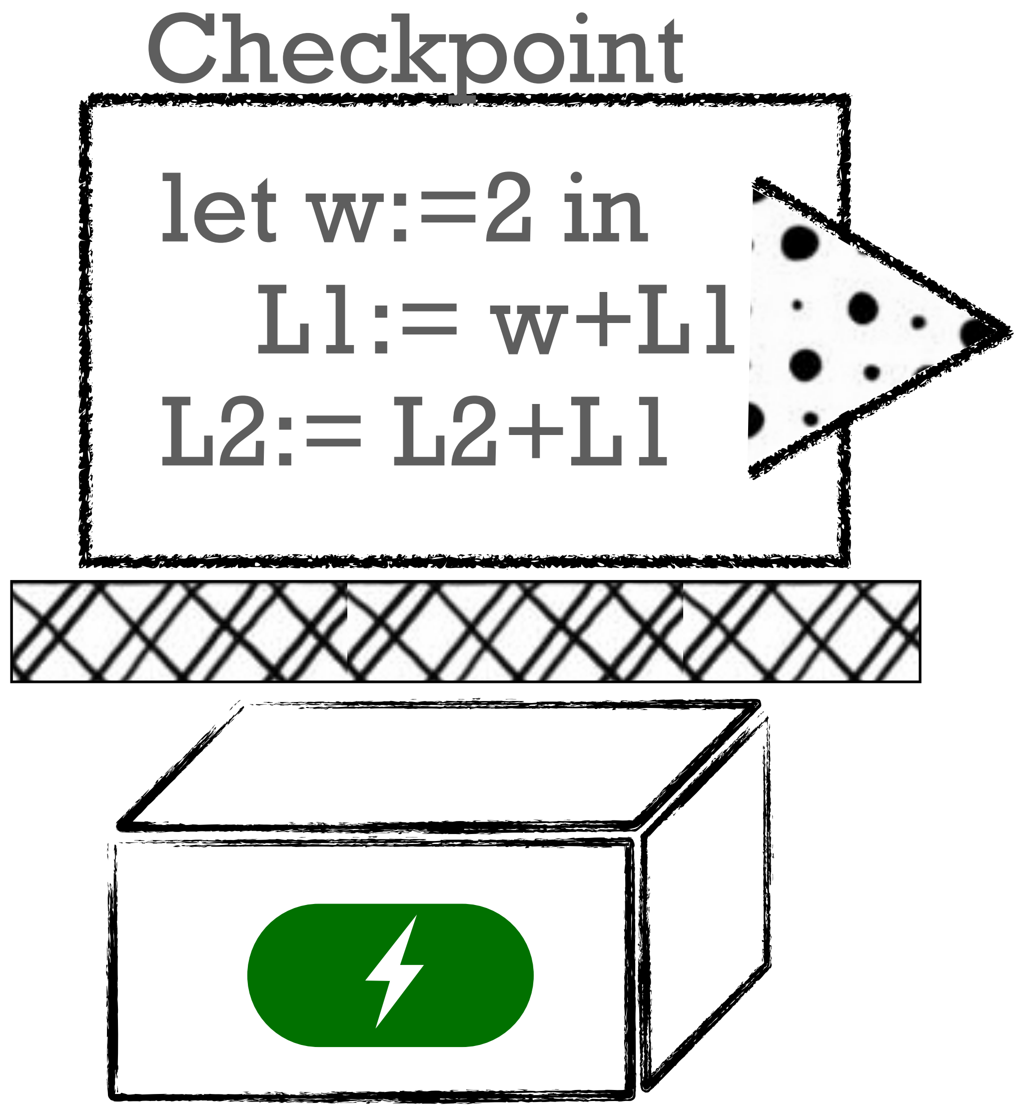
Stable types	$\tau_s := \uparrow \tau_u \mid \text{nat} \rightsquigarrow \tau_s$
Unstable types	$\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid \nu_t$
Basic types	$T := A \mid unit$
Store types	$A := int \mid bool$

Nonvolatile memory
Stable values
$\uparrow$Int



Volatile memory
Unstable values
$\downarrow\uparrow$Int





Computation return type

C Unstable computation

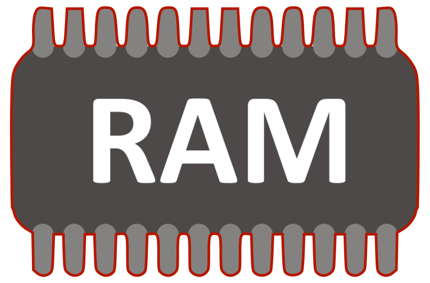
$\uparrow$C Stable computation

Independence principle: stable values cannot depend on unstable values.

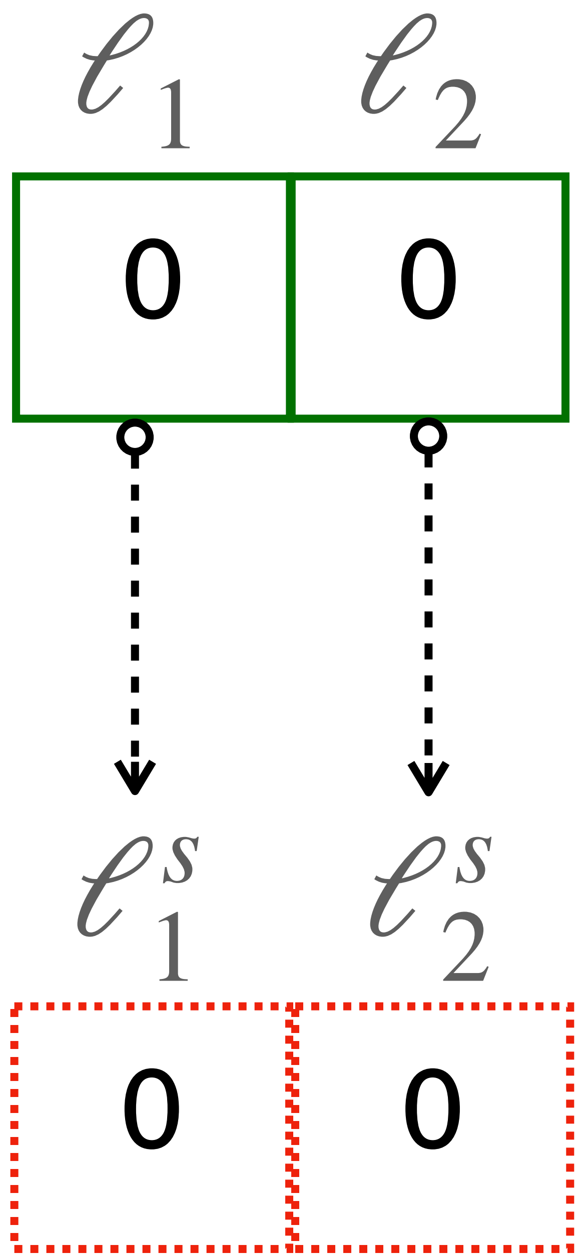
A type system based on adjoint logic

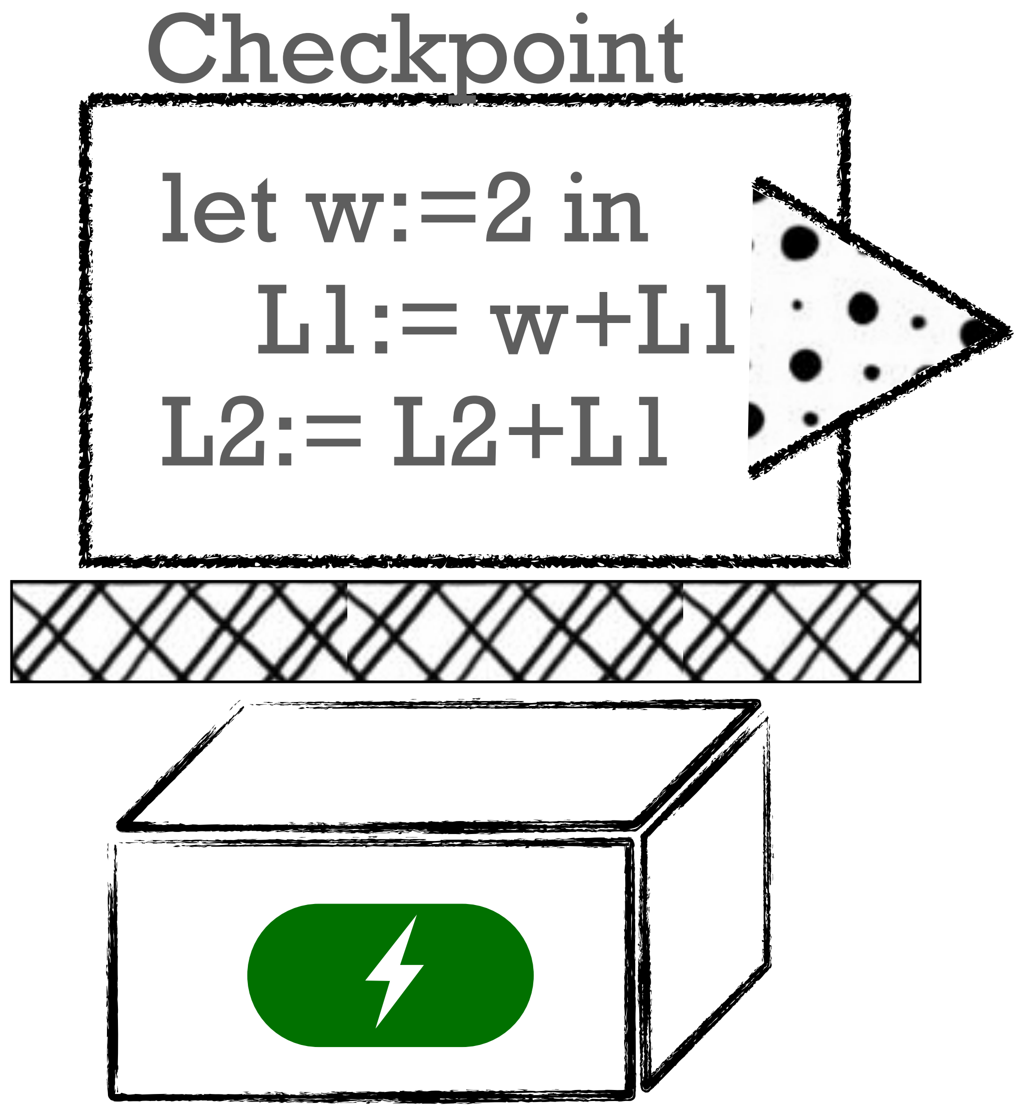
Stable types	$\tau_s := \uparrow \tau_u \mid \mathtt{nat} \rightsquigarrow \tau_s$
Unstable types	$\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid v_t$
Basic types	$T := A \mid unit$
Store types	$A := int \mid bool$

Nonvolatile memory
Stable values
$\uparrow$Int



Volatile memory
Unstable values
$\downarrow\uparrow$Int





Computation return type

C Unstable computation

$\uparrow$C Stable computation

Independence principle: stable values cannot depend on unstable values.

We borrow $\uparrow$ and $\downarrow$ from adjoint logic.

Different judgments judge different things

Stable types	$\tau_s := \uparrow \tau_u \mid \texttt{nat} \rightsquigarrow \tau_s$
Unstable types	$\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid v_t$
Basic types	$T := A \mid unit$
Store types	$A := int \mid bool$

Different judgments judge different things

Stable types	$\tau_s := \uparrow \tau_u \mid \texttt{nat} \rightsquigarrow \tau_s$
Unstable types	$\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid v_t$
Basic types	$T := A \mid unit$
Store types	$A := int \mid bool$

Stable judgments:

Different judgments judge different things

Stable types	$\tau_s := \uparrow \tau_u \mid \texttt{nat} \rightsquigarrow \tau_s$
Unstable types	$\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid v_t$
Basic types	$T := A \mid unit$
Store types	$A := int \mid bool$

Stable judgments:

Unstable judgments:

Different judgments judge different things

Stable types	$\tau_s := \uparrow \tau_u \mid \texttt{nat} \rightsquigarrow \tau_s$
Unstable types	$\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid v_t$
Basic types	$T := A \mid unit$
Store types	$A := int \mid bool$

Stable judgments:

Nonvolatile memory

$\Omega \vdash$ Checkpoint $:\uparrow C$

$\ell : \uparrow A$

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```

Unstable judgments:

Different judgments judge different things

Stable types	$\tau_s := \uparrow \tau_u \mid \texttt{nat} \rightsquigarrow \tau_s$
Unstable types	$\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid v_t$
Basic types	$T := A \mid unit$
Store types	$A := int \mid bool$

Stable judgments:

Nonvolatile memory

Checkpoint

$$\Omega \vdash \boxed{\begin{array}{l}\text{let w:=2 in}\\ \quad \text{L1:= w+L1}\\ \text{L2:= L2+L1}\end{array}} : \uparrow C$$

$\ell : \uparrow A$

Unstable judgments:

Nonvolatile memory | Volatile memory

$$\Omega\,;\ \Sigma \vdash \begin{array}{l}\text{let w:=2 in}\\ \quad \text{L1:= w+L1}\\ \text{L2:= L2+L1}\end{array} : C$$

$\ell : \uparrow A$ $\quad$ $\ell : \downarrow\uparrow A$

Different judgments judge different things

Stable types	$\tau_s := \uparrow \tau_u \mid \mathtt{nat} \rightsquigarrow \tau_s$
Unstable types	$\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid \nu_t$
Basic types	$T := A \mid unit$
Store types	$A := int \mid bool$

Stable judgments:

Nonvolatile memory

$\Omega \vdash$ Checkpoint $:\uparrow C$

$\ell : \uparrow A$

Checkpoint:

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

Unstable judgments:

Nonvolatile memory ; Volatile memory

$\Omega ; \Sigma \vdash$

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

$: C\{$

$\ell : \uparrow A$ $\ell : \downarrow\uparrow A$

Unstable computation (commit volatiles)

Different judgments judge different things

Stable types	$\tau_s := \uparrow \tau_u \mid \texttt{nat} \rightsquigarrow \tau_s$
Unstable types	$\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid v_t$
Basic types	$T := A \mid unit$
Store types	$A := int \mid bool$

Stable judgments:

Nonvolatile memory

$\Omega \vdash$ Checkpoint

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

$: \uparrow C$

$\ell : \uparrow A$

Unstable judgments:

Nonvolatile memory; Volatile memory

$\Omega \, ; \, \Sigma \vdash$

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

$: C \{$

Unstable computation (commit volatiles)

or

$\ell : \uparrow A \quad \ell : \downarrow\uparrow A$

Different judgments judge different things

Stable types	$\tau_s := \uparrow \tau_u \mid \texttt{nat} \rightsquigarrow \tau_s$
Unstable types	$\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid v_t$
Basic types	$T := A \mid unit$
Store types	$A := int \mid bool$

Stable judgments:

Nonvolatile memory

$\Omega \vdash$ Checkpoint : $\uparrow$C

$\ell : \uparrow A$

Checkpoint:

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

Unstable judgments:

Nonvolatile memory ; Volatile memory

$\Omega \; ; \; \Sigma \vdash$

$\ell : \uparrow A \quad \ell : \downarrow\uparrow A$

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

: C {
- Unstable computation (commit volatiles)
- or
- Unstable computation (delete volatiles)

Different judgments judge different things

Stable types	$\tau_s := \uparrow \tau_u \mid \texttt{nat} \rightsquigarrow \tau_s$
Unstable types	$\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid v_t$
Basic types	$T := A \mid unit$
Store types	$A := int \mid bool$

Stable judgments:

Nonvolatile memory

$\Omega \vdash$ Checkpoint

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

$: \uparrow C$

$\ell : \uparrow A$

Unstable judgments:

Nonvolatile memory — Volatile memory

$\Omega \; ; \; \Sigma \vdash$

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

$: C$

- Unstable computation (commit volatiles)
- or
- Unstable computation (delete volatiles)

$\ell : \uparrow A$ $\quad \ell : \downarrow\uparrow A$

Different judgments judge different things

Stable types	$\tau_s := \uparrow \tau_u \mid \mathtt{nat} \rightsquigarrow \tau_s$
Unstable types	$\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid v_t$
Basic types	$T := A \mid unit$
Store types	$A := int \mid bool$

Stable judgments:

Nonvolatile memory

$\Omega \vdash$ **Checkpoint** $: \uparrow C$

$\ell : \uparrow A$

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

Unstable judgments:

Nonvolatile memory; Volatile memory

$\Omega \,;\, \Sigma \vdash$

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

$: C$

$\ell : \uparrow A$ $\quad$ $\ell : \downarrow\uparrow A$

- Unstable computation (commit volatiles)
- or
- Unstable computation (delete volatiles)

Different judgments judge different things

Stable types	$\tau_s := \uparrow \tau_u \mid \texttt{nat} \rightsquigarrow \tau_s$
Unstable types	$\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid v_t$
Basic types	$T := A \mid unit$
Store types	$A := int \mid bool$

Stable judgments:

Nonvolatile memory

$\Omega \vdash$ Checkpoint [let w:=2 in L1:= w+L1 L2:= L2+L1] $: \uparrow C$

$\ell : \uparrow A$

Unstable judgments:

Nonvolatile memory ; Volatile memory

$\Omega ; \Sigma \vdash$ let w:=2 in L1:= w+L1 L2:= L2+L1 $: C$

$\ell : \uparrow A$ $\ell : \downarrow\uparrow A$

C is either:

- Unstable computation (commit volatiles) $\downarrow$
- or
- Unstable computation (delete volatiles)

Different judgments judge different things

Stable types	$\tau_s := \uparrow \tau_u \mid \mathtt{nat} \rightsquigarrow \tau_s$
Unstable types	$\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid v_t$
Basic types	$T := A \mid unit$
Store types	$A := int \mid bool$

Stable judgments:

Nonvolatile memory

$\Omega \vdash$ Checkpoint

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

$: \uparrow C$

$\ell : \uparrow A$

Unstable judgments:

Nonvolatile memory | Volatile memory

$\Omega \; ; \; \Sigma \vdash$

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

$: C$ {
- Unstable computation (commit volatiles) $\downarrow\uparrow$
- or
- Unstable computation (delete volatiles)

$\ell : \uparrow A$ $\ell : \downarrow\uparrow A$

Different judgments judge different things

Stable types	$\tau_s := \uparrow \tau_u \mid \text{nat} \rightsquigarrow \tau_s$
Unstable types	$\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid v_t$
Basic types	$T := A \mid unit$
Store types	$A := int \mid bool$

Stable judgments:

Nonvolatile memory

$\Omega \vdash$ Checkpoint (let w:=2 in L1:= w+L1 L2:= L2+L1) $: \uparrow C$

$\ell : \uparrow A$

Unstable judgments:

Nonvolatile memory ; Volatile memory

$\Omega \, ; \, \Sigma \vdash$ let w:=2 in L1:= w+L1 L2:= L2+L1 $: C$

$\ell : \uparrow A$ $\ell : \downarrow\uparrow A$

C is either:

- Unstable computation (commit volatiles) $\downarrow\uparrow$ unit
- or
- Unstable computation (delete volatiles)

Different judgments judge different things

Stable types	$\tau_s := \uparrow \tau_u \mid \mathtt{nat} \rightsquigarrow \tau_s$
Unstable types	$\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid v_t$
Basic types	$T := A \mid unit$
Store types	$A := int \mid bool$

Stable judgments:

Nonvolatile memory

$\Omega \vdash$ **Checkpoint**

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

$: \uparrow C$

$\ell : \uparrow A$

Unstable judgments:

Nonvolatile memory; Volatile memory

$\Omega \, ; \, \Sigma \vdash$

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

$: C$

- Unstable computation (commit volatiles) $\downarrow\uparrow$ unit
- or
- Unstable computation (delete volatiles) $\downarrow$

$\ell : \uparrow A$ (nonvolatile) $\quad \ell : \downarrow\uparrow A$ (volatile)

Different judgments judge different things

Stable types	$\tau_s := \uparrow \tau_u \mid \mathtt{nat} \rightsquigarrow \tau_s$
Unstable types	$\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid v_t$
Basic types	$T := A \mid unit$
Store types	$A := int \mid bool$

Stable judgments:

Nonvolatile memory

$\Omega \vdash$ Checkpoint

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

$: \uparrow C$

$\ell : \uparrow A$

Unstable judgments:

Nonvolatile memory ; Volatile memory

$\Omega \; ; \; \Sigma \vdash$

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

$: C$

- Unstable computation (commit volatiles) $\downarrow\uparrow$ unit
- or
- Unstable computation (delete volatiles) $\downarrow(\mathrm{nat} \rightsquigarrow \uparrow C)$

$\ell : \uparrow A$ $\quad \ell : \downarrow\uparrow A$

Different judgments judge different things

Stable types	$\tau_s := \uparrow \tau_u \mid \mathtt{nat} \rightsquigarrow \tau_s$
Unstable types	$\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid v_t$
Basic types	$T := A \mid unit$
Store types	$A := int \mid bool$

Stable judgments:

Nonvolatile memory

$\Omega \vdash$ **Checkpoint**

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

$: \uparrow C$

$\ell : \uparrow A$

Unstable judgments:

Nonvolatile memory ; Volatile memory

$\Omega \, ; \, \Sigma \vdash$

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

$: C$

- Unstable computation (commit volatiles) $\downarrow\uparrow$ unit
- or
- Unstable computation (delete volatiles) $\downarrow(\mathbf{nat} \rightsquigarrow \uparrow C)$ recharge

$\ell : \uparrow A$ $\quad$ $\ell : \downarrow\uparrow A$

Different judgments judge different things

Stable types	$\tau_s := \uparrow \tau_u \mid \mathtt{nat} \rightsquigarrow \tau_s$
Unstable types	$\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid v_t$
Basic types	$T := A \mid unit$
Store types	$A := int \mid bool$

Stable judgments:

Nonvolatile memory

$\Omega \vdash$ Checkpoint $: \uparrow C$

$\ell : \uparrow A$

Checkpoint:

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

Unstable judgments:

Nonvolatile memory; Volatile memory

$\Omega \,;\, \Sigma \vdash$ (program below) $: C$

$\ell : \uparrow A$; $\ell : \downarrow\uparrow A$

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

$: C$ {

- Unstable computation (commit volatiles) $\downarrow\uparrow$ unit
- or
- Unstable computation (delete volatiles) $\downarrow(\mathbf{nat} \rightsquigarrow \uparrow C)$ recharge

Different judgments judge different things

Stable types	$\tau_s := \uparrow \tau_u \mid \mathtt{nat} \rightsquigarrow \tau_s$
Unstable types	$\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid \nu_t$
Basic types	$T := A \mid unit$
Store types	$A := int \mid bool$

Stable judgments:

Nonvolatile memory

Ω $\vdash$

$\ell : \uparrow A$

Checkpoint

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

$: \uparrow C$

Unstable judgments:

Nonvolatile memory ; Volatile memory

$\Omega \; ; \; \Sigma \vdash$

$\ell : \uparrow A$ $\quad$ $\ell : \downarrow\uparrow A$

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

$: C$

- Unstable computation (commit volatiles) $\downarrow\uparrow$ unit
- or
- Unstable computation (delete volatiles) $\downarrow(\mathtt{nat} \rightsquigarrow \uparrow C)$ recharge

Different judgments judge different things

Stable types	$\tau_s := \uparrow \tau_u \mid \mathtt{nat} \rightsquigarrow \tau_s$
Unstable types	$\tau_u := T \mid \downarrow \tau_s \mid \tau_u \vee \tau_u \mid v_t$
Basic types	$T := A \mid unit$
Store types	$A := int \mid bool$

Stable judgments:

Nonvolatile memory

$\Omega \vdash$

Checkpoint

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

$: \uparrow C$

$\ell : \uparrow A$

$C = \downarrow\uparrow \text{unit} \vee \downarrow(\text{nat} \rightsquigarrow \uparrow C)$

Unstable judgments:

Nonvolatile memory — Volatile memory

$\Omega \; ; \; \Sigma \vdash$

```
let w:=2 in
   L1:= w+L1
L2:= L2+L1
```

$: C$

$\ell : \uparrow A$ — $\ell : \downarrow\uparrow A$

Unstable computation (commit volatiles) — $\downarrow\uparrow \text{unit}$

or

Unstable computation (delete volatiles) — $\downarrow(\text{nat} \rightsquigarrow \uparrow C)$ recharge

Background: adjoint logic

Background: adjoint logic

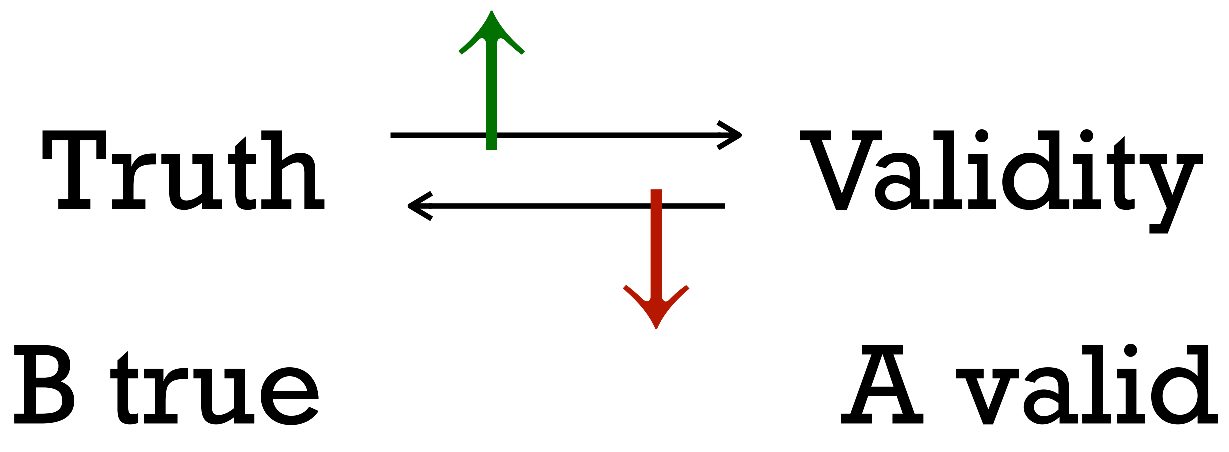
- Combine modes of truth

  (Benton 1994),
  (Pfenning and Davies 1999),
  (Reed 2009)
  (Pruiksma et al 2020)

Background: adjoint logic

- Combine modes of truth

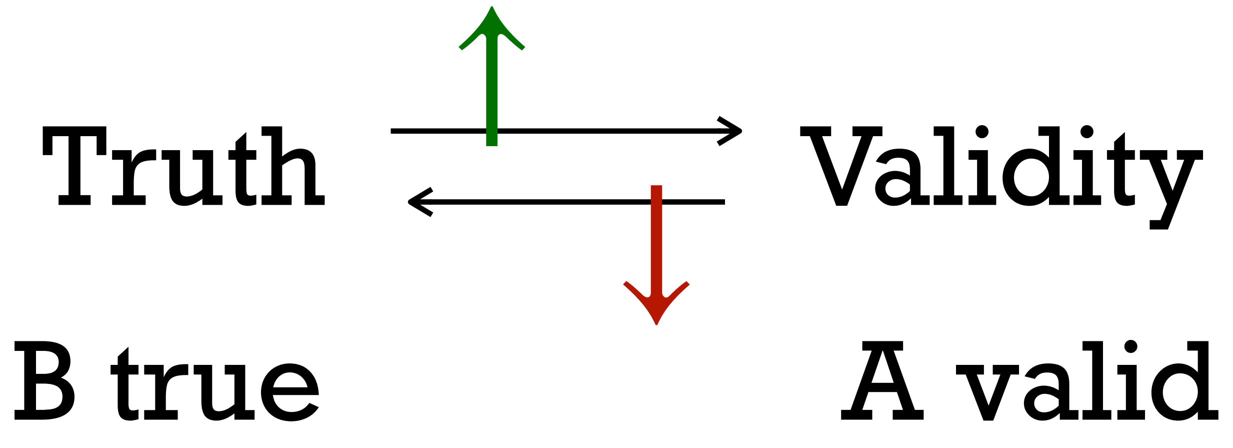
  (Benton 1994),
  (Pfenning and Davies 1999),
  (Reed 2009)
  (Pruiksma et al 2020)



Background: adjoint logic

- Combine modes of truth

  (Benton 1994),
  (Pfenning and Davies 1999),
  (Reed 2009)
  (Pruiksma et al 2020)



Independence principle:

Validity cannot depend on truth

Background: adjoint logic

- Combine modes of truth

  (Benton 1994),
  (Pfenning and Davies 1999),
  (Reed 2009)
  (Pruiksma et al 2020)

Truth ⇄ Validity

B true   A valid

Independence principle:

Validity cannot depend on truth

- <u>Combine stable and unstable values</u>

Unstable ⇄ Stable

$x : \tau_u$ $x : \tau_s$

Stable values cannot depend on unstable values

Different judgments judge different things

Different judgments judge different things

Stable judgments: $\underset{x:\tau_s}{\Omega} \vdash \tau_s$

Different judgments judge different things

Stable judgments: $\underset{x:\tau_s}{\Omega} \vdash \tau_s$

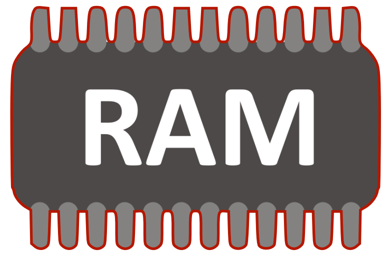
Unstable judgments: $\underset{x:\tau_s}{\Omega}\,;\,\underset{x:\tau_u}{\Sigma} \vdash \tau_u$

Type system based on adjoint logic rules

Nonvolatile memory

Stable values $\uparrow$Int

ℓ_1	ℓ_2
0	0

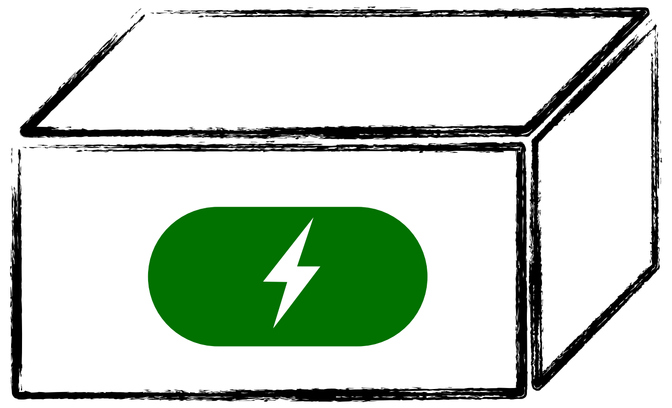


Volatile memory

Unstable values $\downarrow\uparrow$Int

pc → Checkpoint

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```

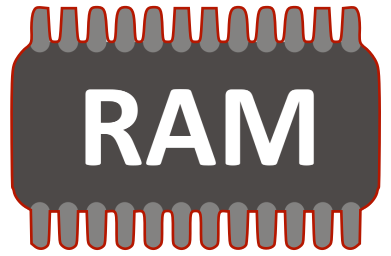


Type system based on adjoint logic rules

Nonvolatile memory

Stable values ↑Int

ℓ_1	ℓ_2
0	0

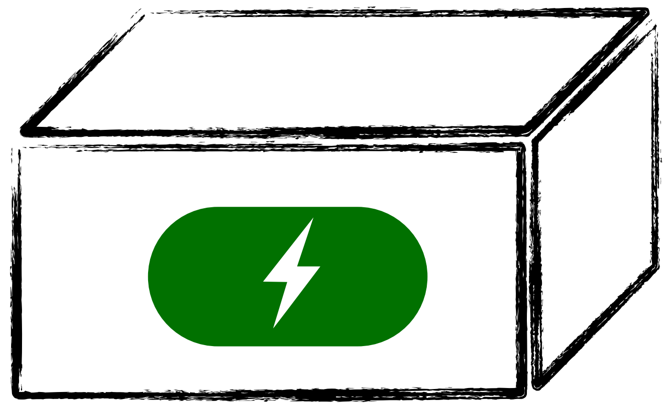


Volatile memory

Unstable values ↓↑Int

pc → Checkpoint

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```



Check point

$\Omega \vdash$

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```

$: \uparrow C_{\text{Unit}}$

Type system based on adjoint logic rules

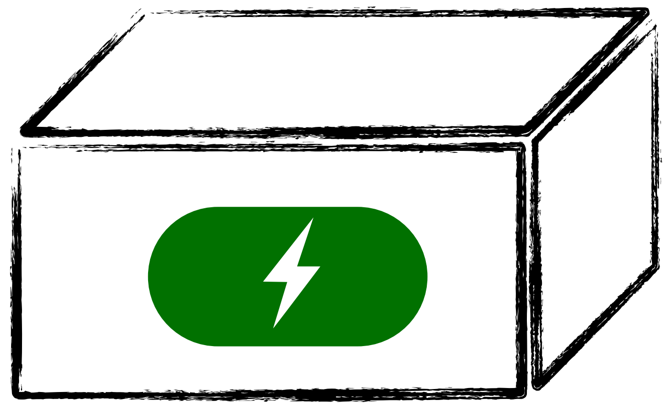
Nonvolatile memory

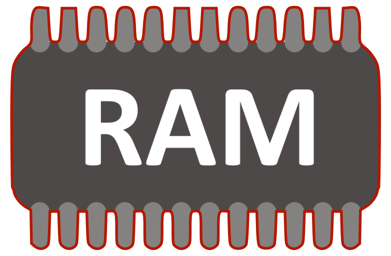
Stable values $\uparrow$Int

ℓ_1	ℓ_2
0	0

pc → Checkpoint

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```





Volatile memory

Unstable values $\downarrow\uparrow$Int

Typing rule:

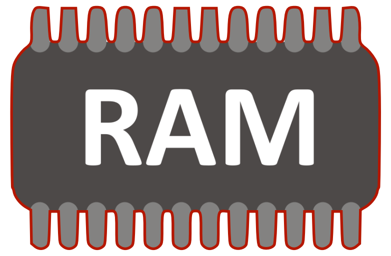
Check point

$$\Omega \vdash \boxed{\begin{array}{l}\text{let w:=2 in}\\ \quad \text{L1:= w+L1}\\ \text{L2:= L2+L1}\end{array}} : \uparrow C_{\text{Unit}}$$

Type system based on adjoint logic rules

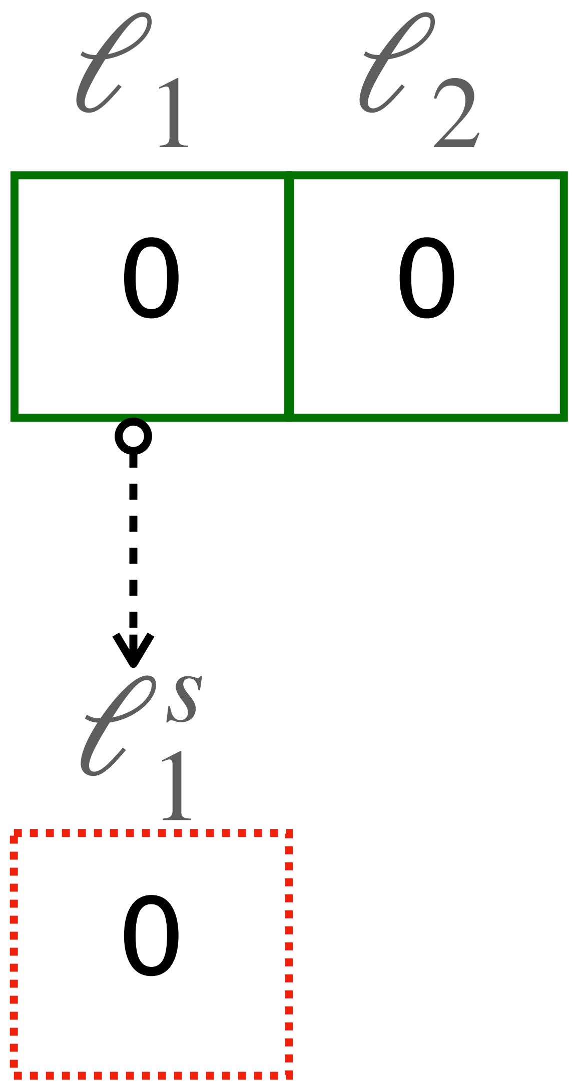
Nonvolatile memory

Stable values $\uparrow$Int



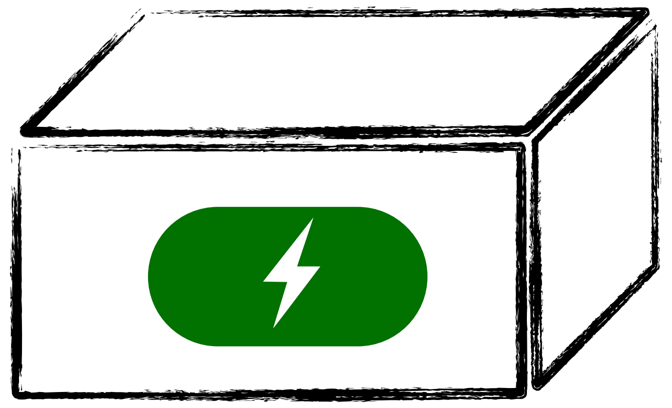
Volatile memory

Unstable values $\downarrow\uparrow$Int



pc → Checkpoint

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```



Typing rule:

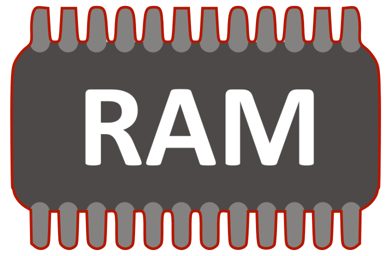
Check point

$\Omega \vdash$
```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```
$: \uparrow C_{Unit}$

Type system based on adjoint logic rules

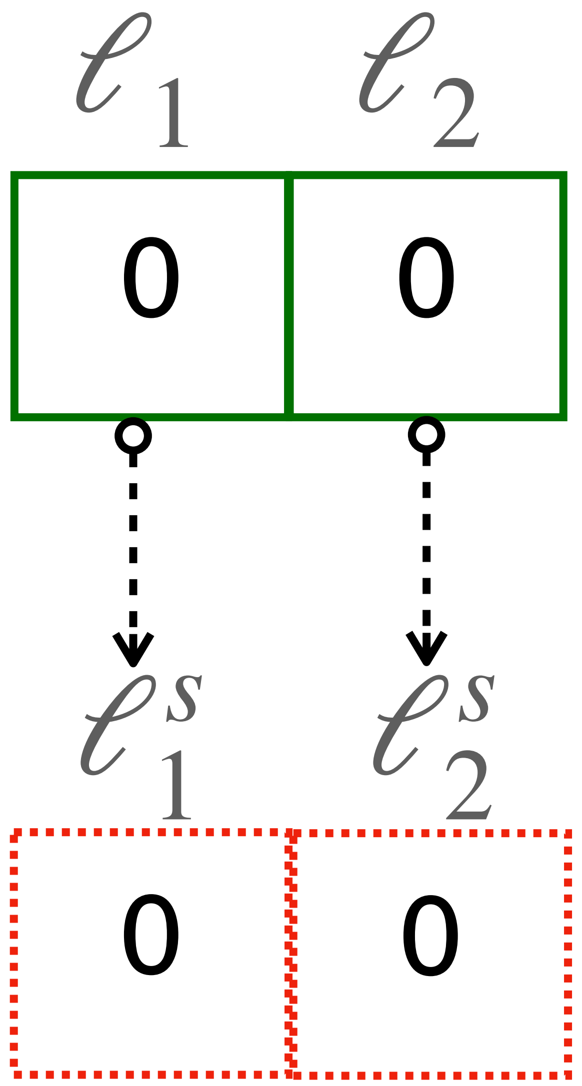
Nonvolatile memory

Stable values $\uparrow$Int



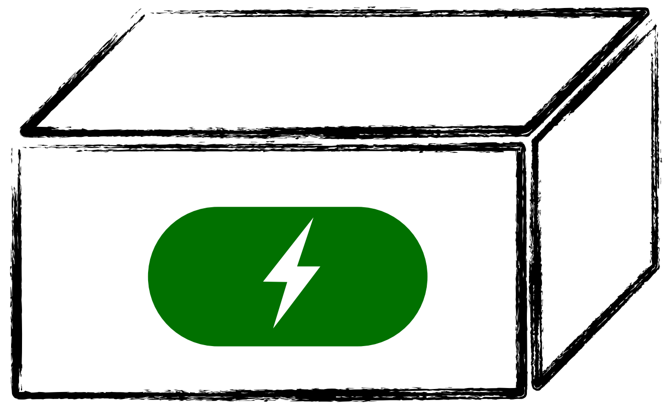
Volatile memory

Unstable values $\downarrow\uparrow$Int



pc → Checkpoint

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```



Typing rule:

Check point

$\Omega \vdash$

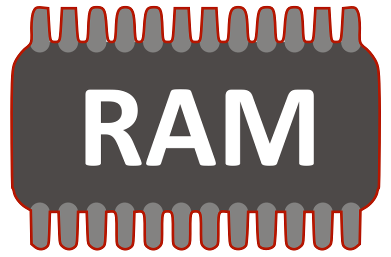
```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```

$: \uparrow C_{Unit}$

Type system based on adjoint logic rules

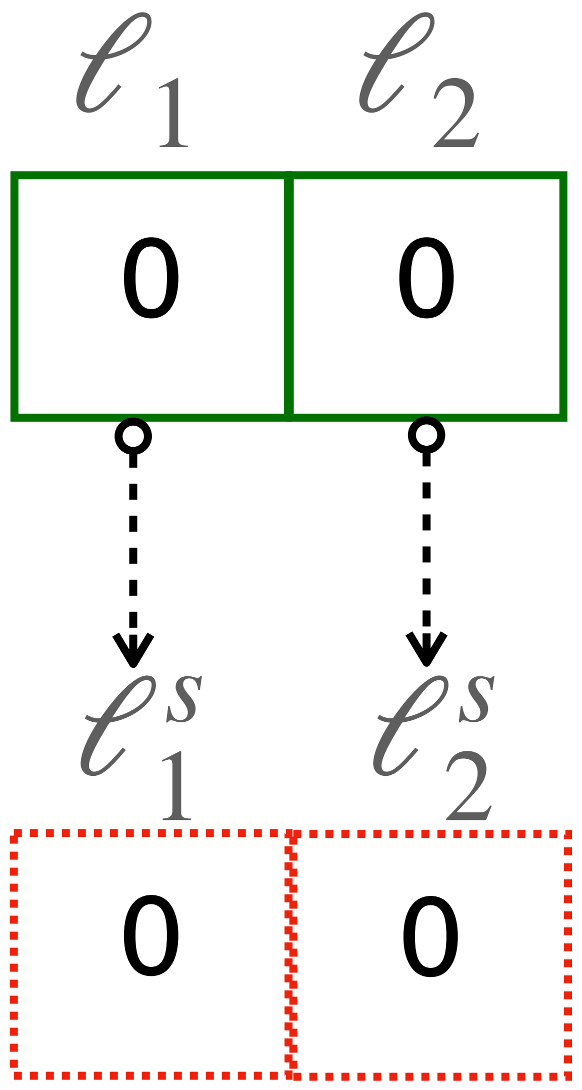
Nonvolatile memory

Stable values $\uparrow$Int



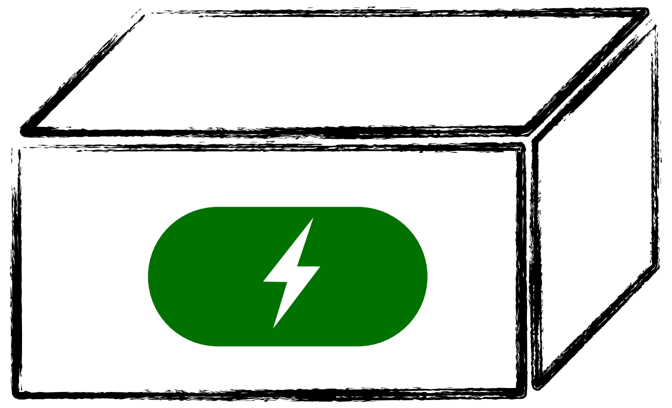
Volatile memory

Unstable values $\downarrow\uparrow$Int



pc → Checkpoint

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```



Typing rule:

$$\overline{\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad} \text{ Ckpt}$$

Check point

$\Omega \vdash$

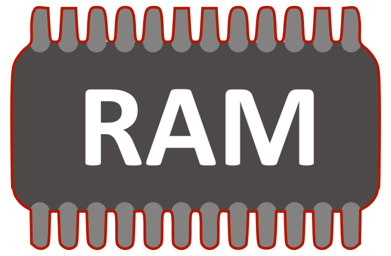
```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```

$: \uparrow C_{\text{Unit}}$

Type system based on adjoint logic rules

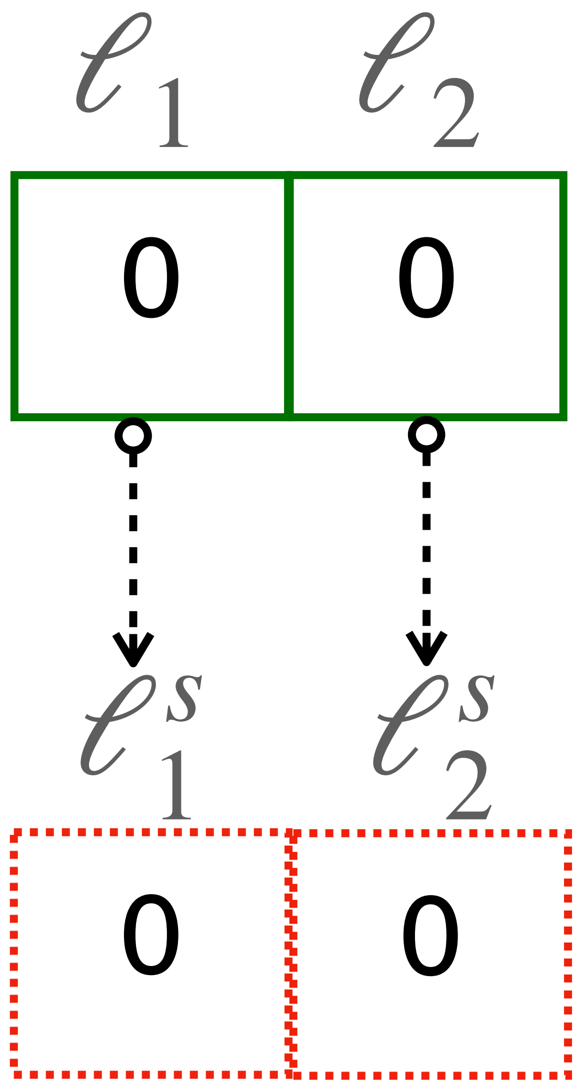
Nonvolatile memory

Stable values $\uparrow$Int



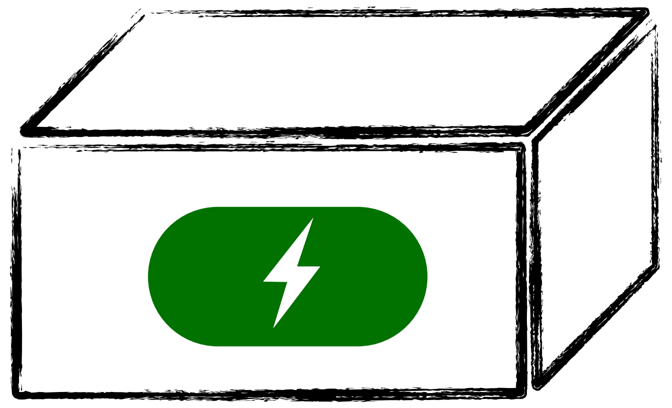
Volatile memory

Unstable values $\downarrow\uparrow$Int

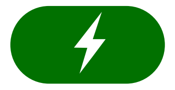


pc → Checkpoint

let w:=2 in
L1:= w+L1
L2:= L2+L1



Typing rule:

 $\Omega ; \Sigma \vdash$ let w:=2 in L1:= w+L1 L2:= L2+L1 $: C_{\text{Unit}}$

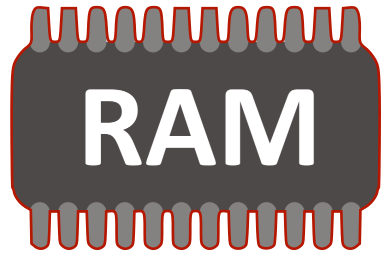
Ckpt

$\Omega \vdash$ Check point [let w:=2 in L1:= w+L1 L2:= L2+L1] $: \uparrow C_{\text{Unit}}$

Type system based on adjoint logic rules

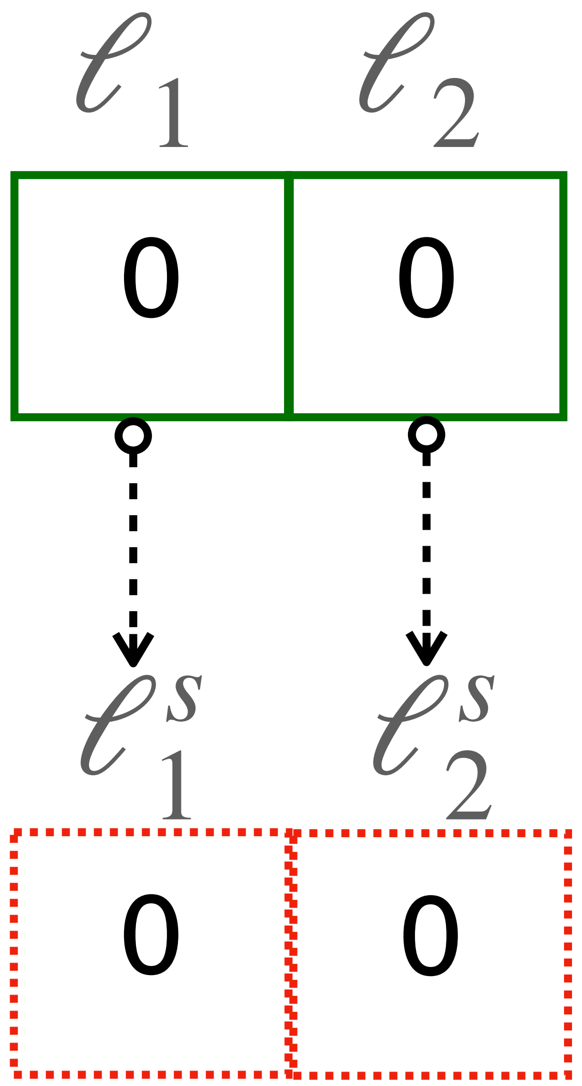
Nonvolatile memory

Stable values $\uparrow$Int



Volatile memory

Unstable values $\downarrow\uparrow$Int



pc → Checkpoint

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```

Adjoint logic:

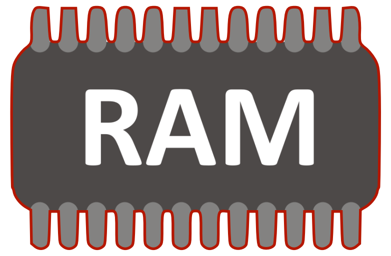
Typing rule:

$$\frac{\Omega ; \Sigma \vdash \begin{array}{l}\text{let w:=2 in}\\ \quad \text{L1:= w+L1}\\ \text{L2:= L2+L1}\end{array} : C_{\text{Unit}}}{\Omega \vdash \text{Check point}\begin{array}{|l|}\hline \text{let w:=2 in}\\ \quad \text{L1:= w+L1}\\ \text{L2:= L2+L1}\\ \hline\end{array} : \uparrow C_{\text{Unit}}}\ \text{Ckpt}$$

Type system based on adjoint logic rules

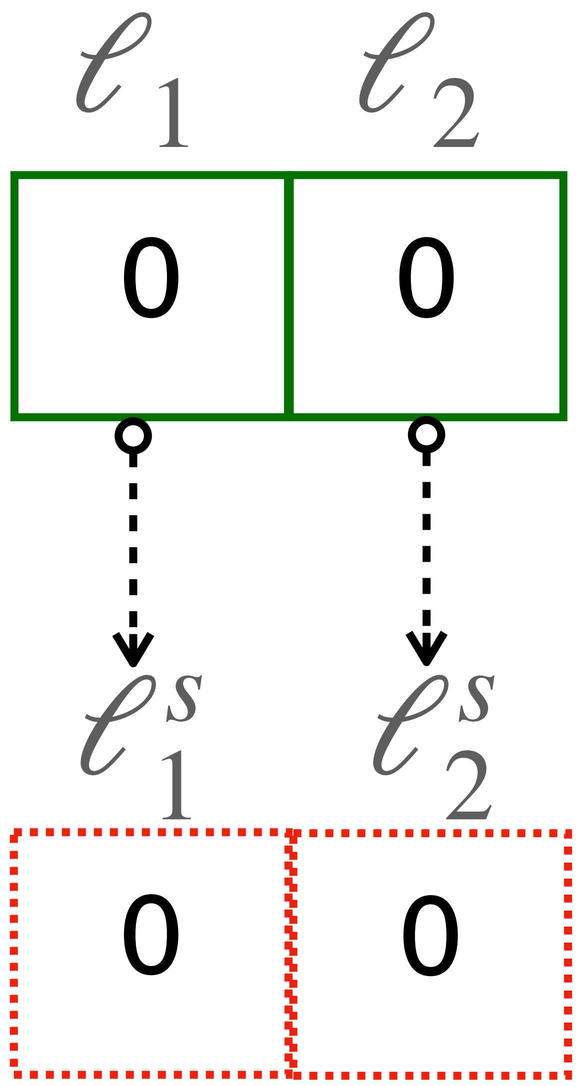
Nonvolatile memory

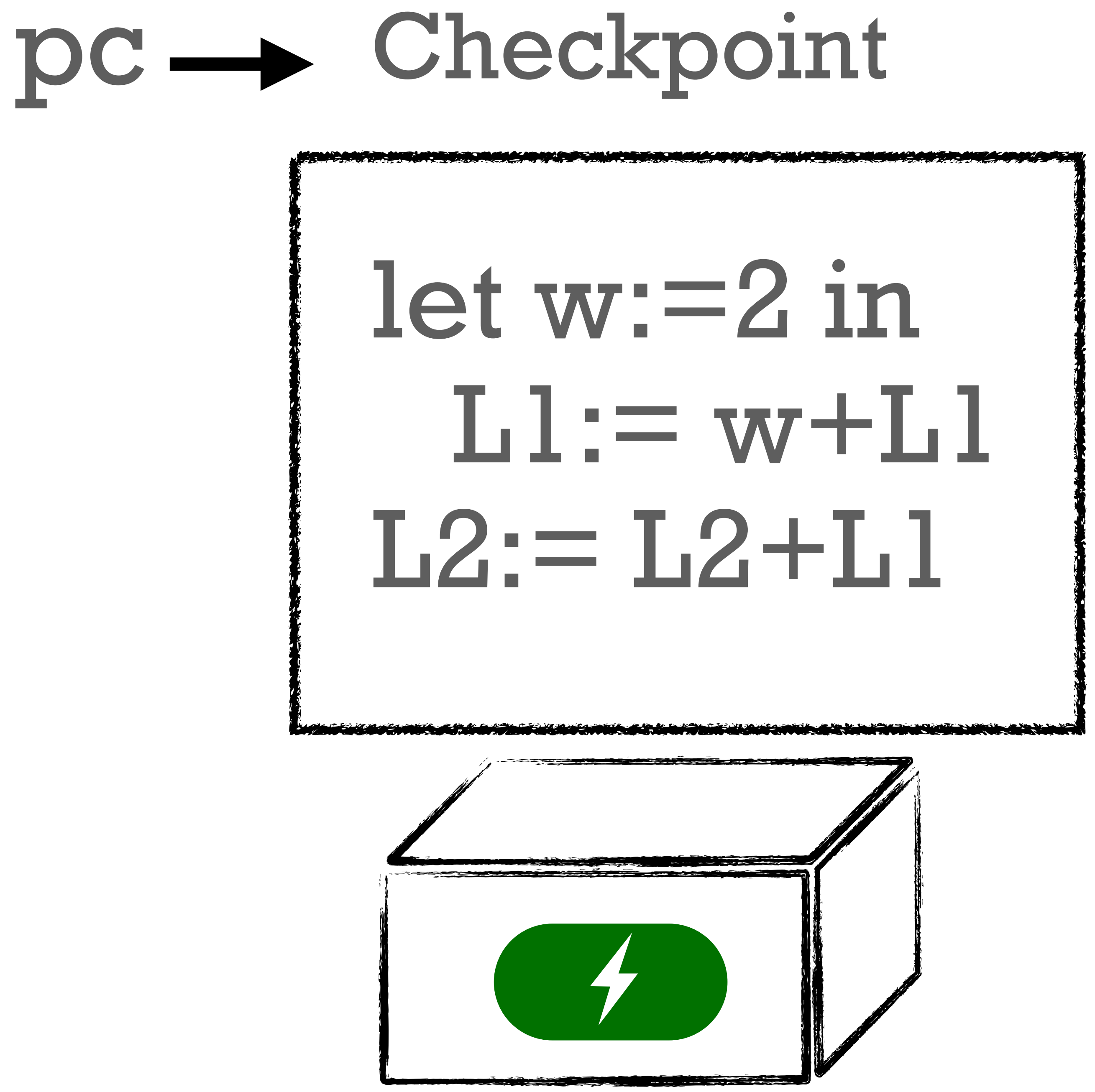
Stable values $\uparrow$Int



Volatile memory

Unstable values $\downarrow\uparrow$Int





Typing rule:

$$\Omega\,;\,\Sigma \vdash \begin{array}{l}\text{let w:=2 in}\\ \quad\text{L1:= w+L1}\\ \text{L2:= L2+L1}\end{array} : \mathrm{C}_{\mathrm{Unit}}$$

Ckpt

Check point

$$\Omega \vdash \boxed{\begin{array}{l}\text{let w:=2 in}\\ \quad\text{L1:= w+L1}\\ \text{L2:= L2+L1}\end{array}} : \uparrow\mathrm{C}_{\mathrm{Unit}}$$

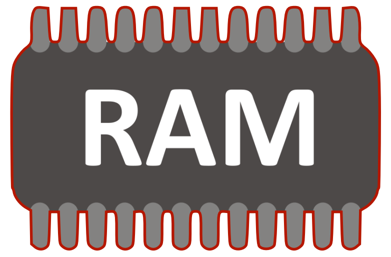
Adjoint logic:

$$\frac{\Omega\,;\,\bullet \vdash \tau}{\Omega \vdash \uparrow\tau}\ \uparrow\mathrm{R}$$

Type system based on adjoint logic rules

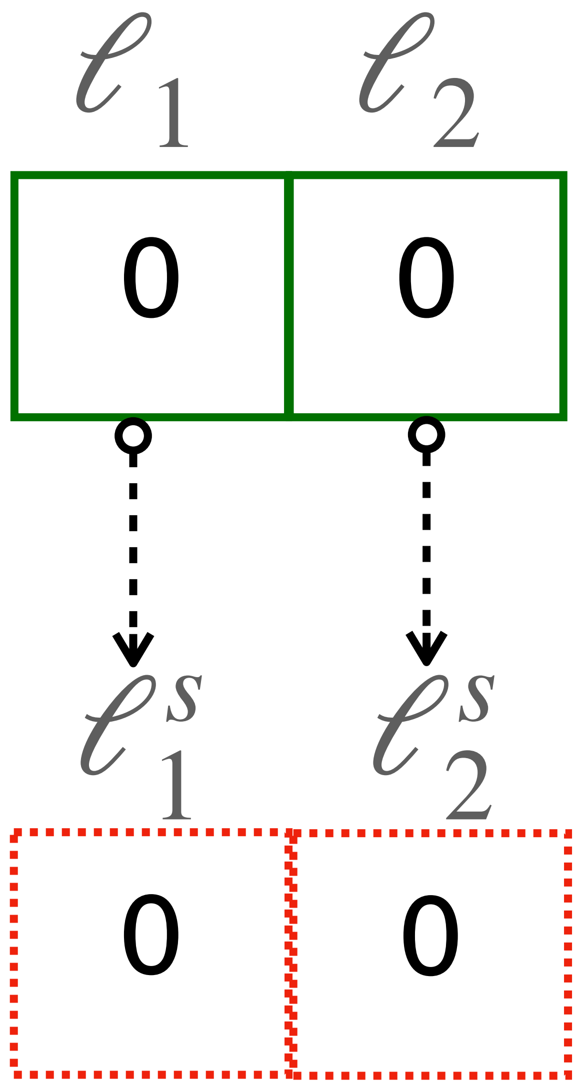
Nonvolatile memory

Stable values $\uparrow\mathrm{Int}$



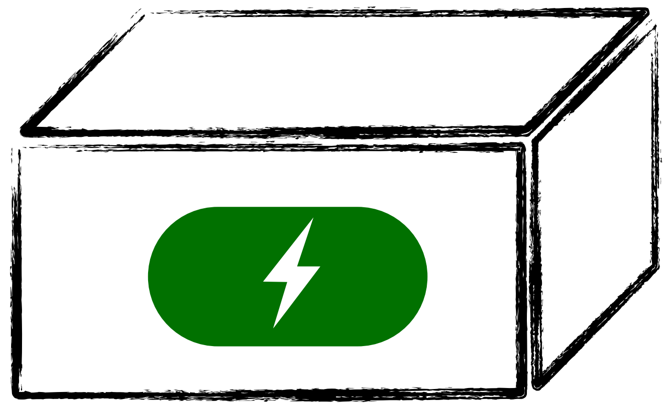
Volatile memory

Unstable values $\downarrow\uparrow\mathrm{Int}$



pc → Checkpoint

```
let w:=2 in
  L1:= w+L1
L2:= L2+L1
```



Typing rule:

$$\frac{\Omega\,;\,\Sigma \vdash \begin{array}{l}\text{let w:=2 in}\\ \quad\text{L1:= w+L1}\\ \text{L2:= L2+L1}\end{array} : \mathrm{C}_{\mathrm{Unit}}}{\Omega \vdash \boxed{\begin{array}{l}\text{let w:=2 in}\\ \quad\text{L1:= w+L1}\\ \text{L2:= L2+L1}\end{array}} : \uparrow\mathrm{C}_{\mathrm{Unit}}}\ \text{Ckpt}$$

Check point

Adjoint logic:

$$\frac{\Omega, \uparrow A\,;\, \downarrow\uparrow A, \Sigma \vdash \tau}{\Omega, \uparrow A\,;\, \Sigma \vdash \tau}\ \uparrow L^*$$

$$\frac{\Omega\,;\, \bullet \vdash \tau}{\Omega \vdash \uparrow\tau}\ \uparrow R$$

Outline

- ☑ A type system based on adjoint modalities and independence principle
- ☐ Correctness as a logical relation
- ☐ Conclusion

Correctness of programs

Every intermittent execution of the program can be simulated by a continuous execution of it.

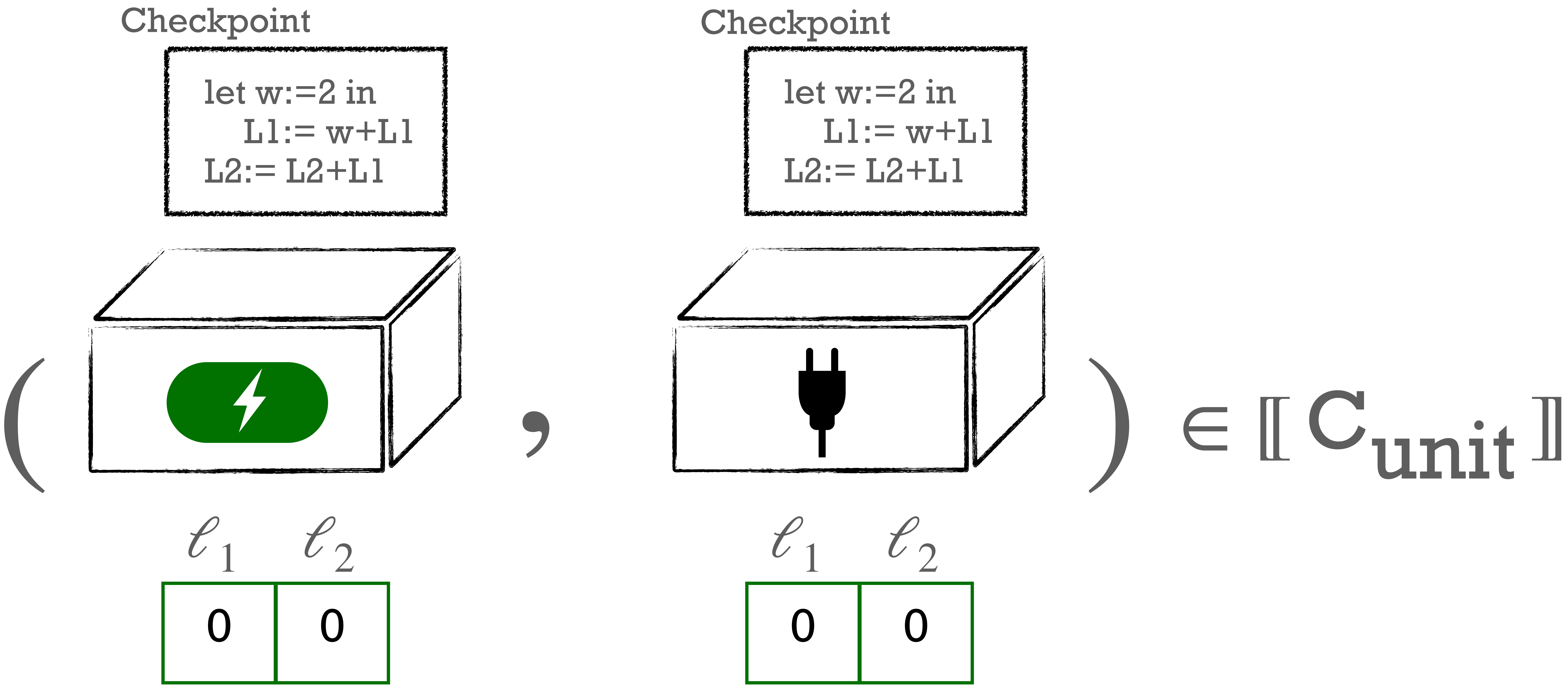
Correctness of programs

Every intermittent execution of the program can be simulated by a continuous execution of it.

We define correctness based on the meaning of crash types annotated with $\uparrow$ and $\downarrow$.

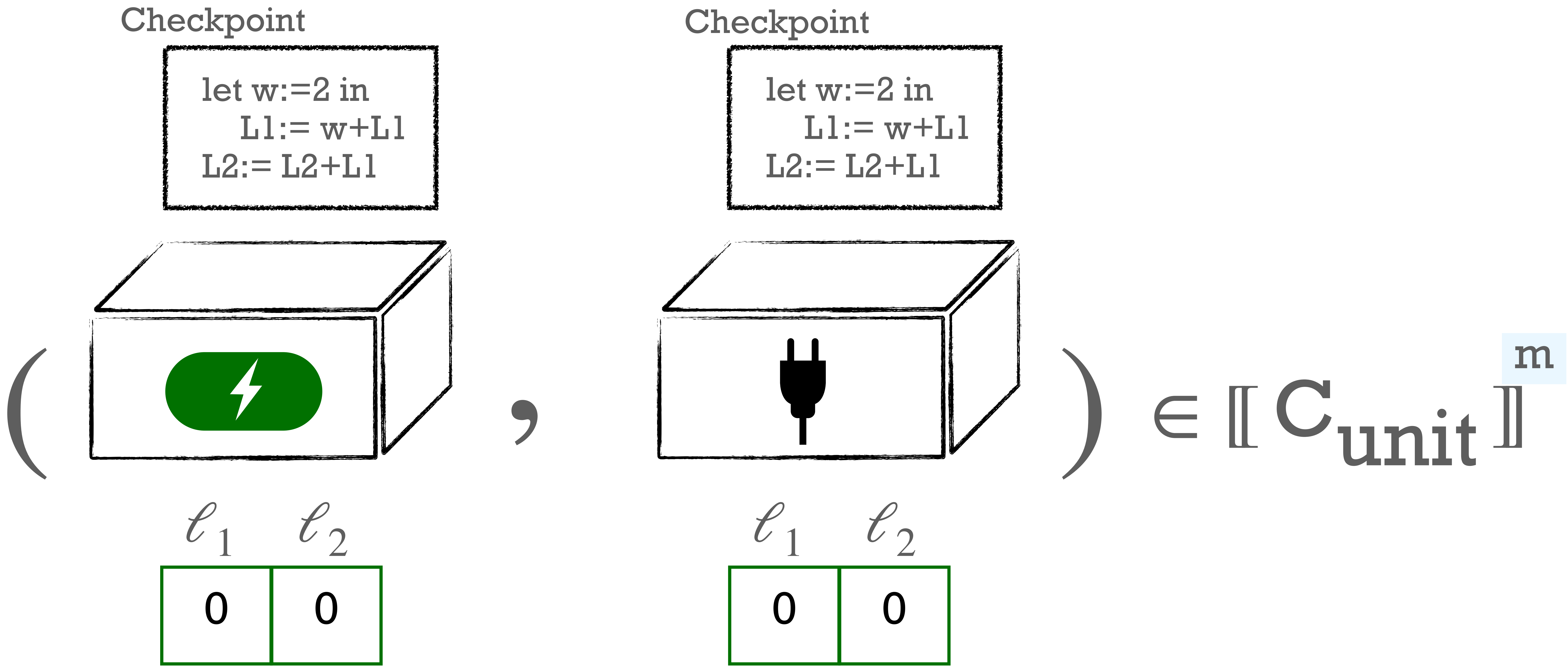
Logical relation for correctness

An intermittent execution of a well-typed program can be simulated by its continuous execution.



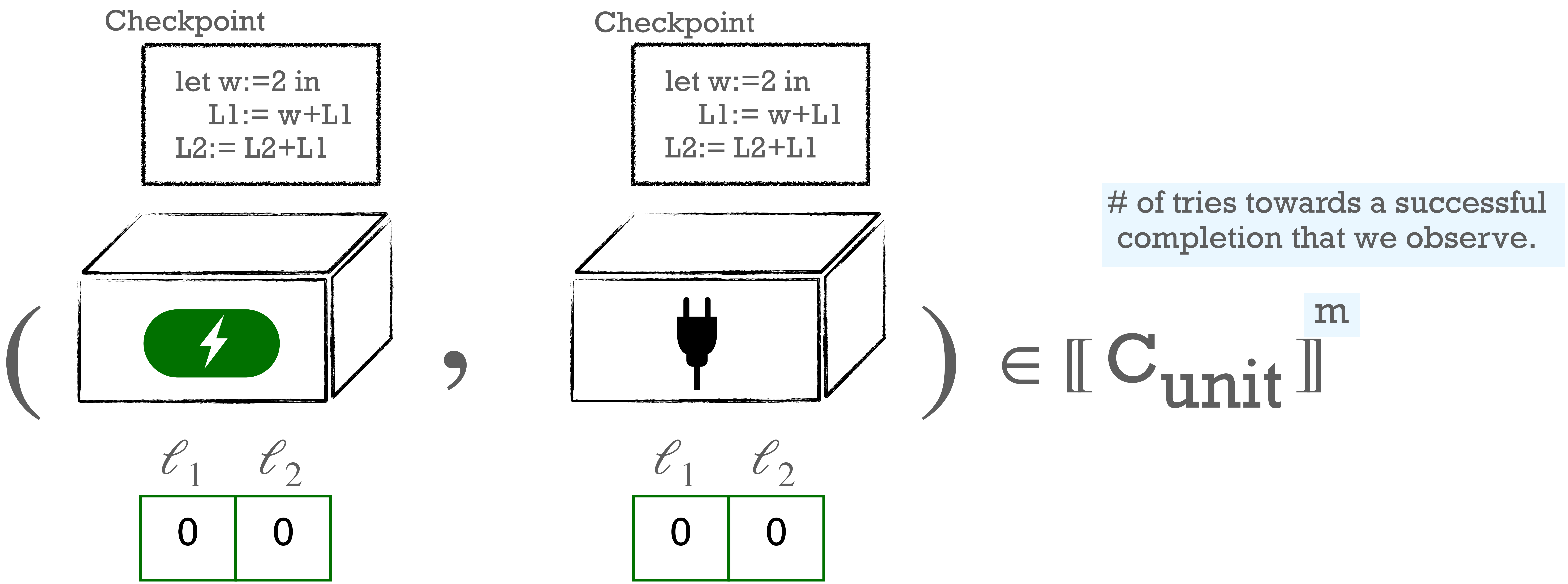
Logical relation for correctness

An intermittent execution of a well-typed program can be simulated by its continuous execution.



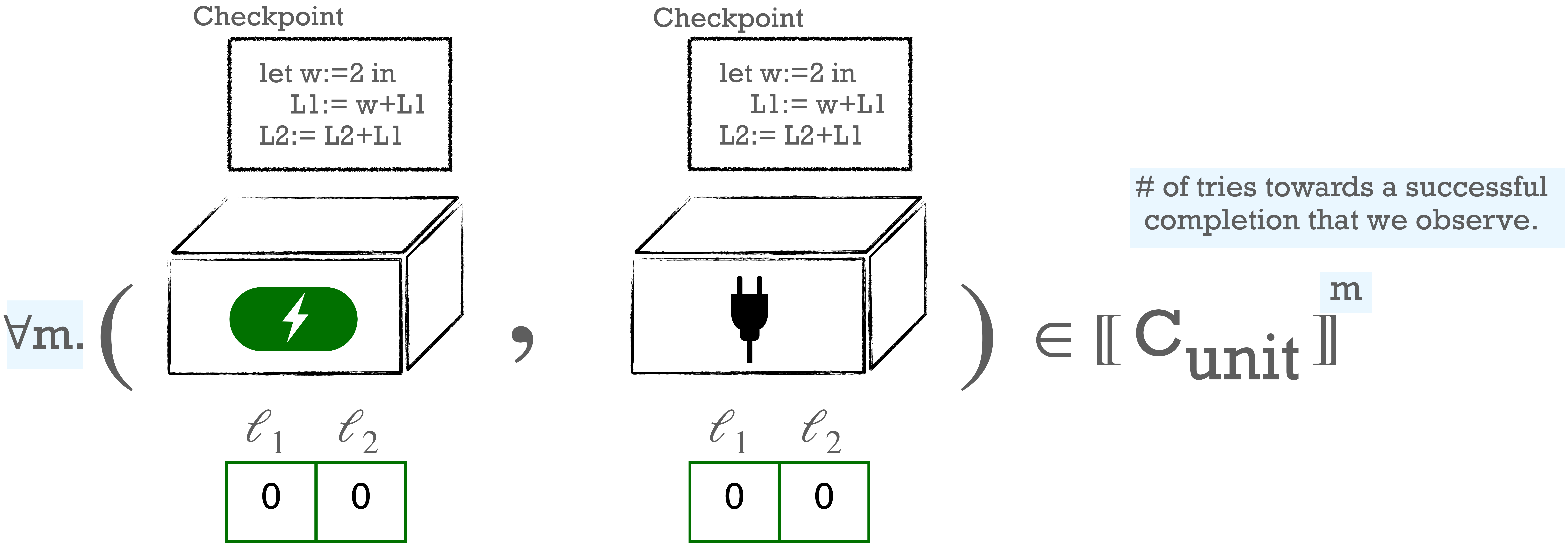
Logical relation for correctness

An intermittent execution of a well-typed program can be simulated by its continuous execution.

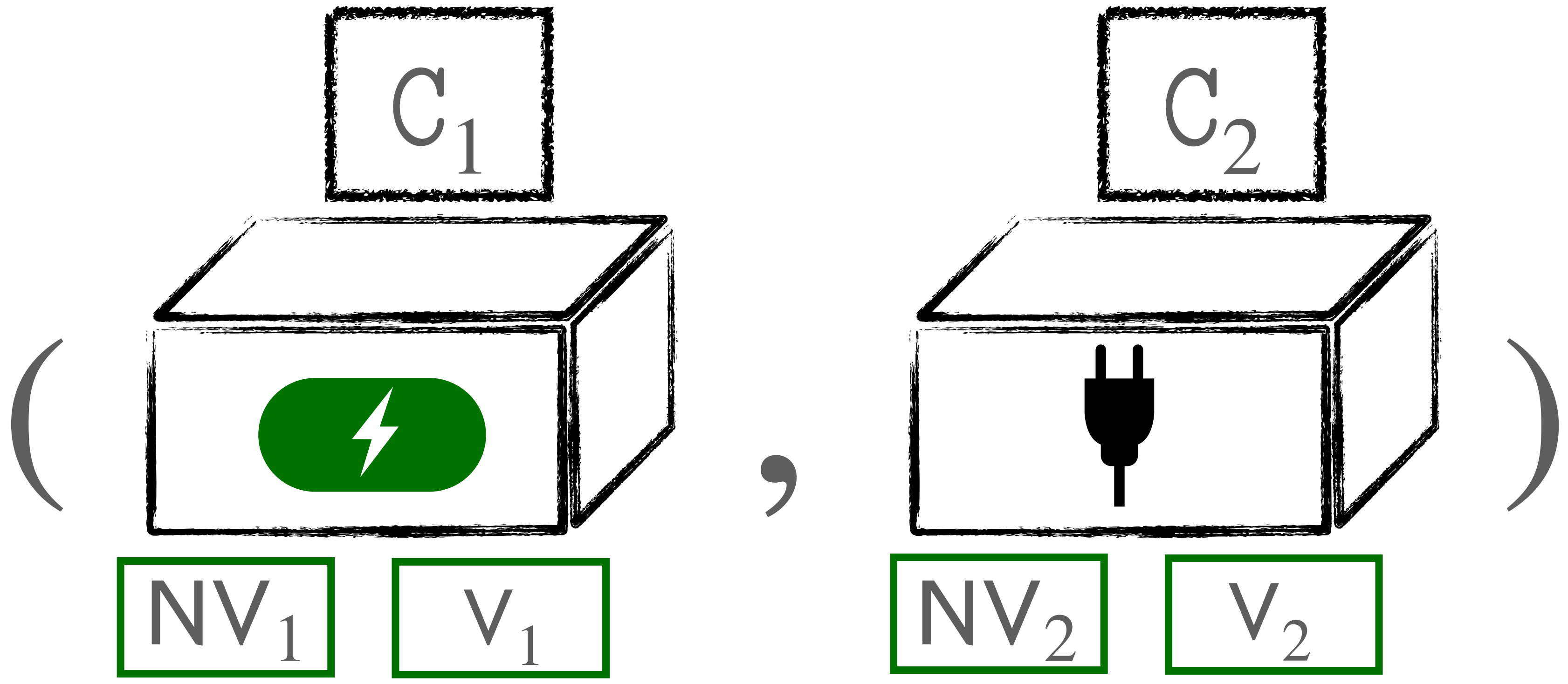


Logical relation for correctness

An intermittent execution of a well-typed program can be simulated by its continuous execution.

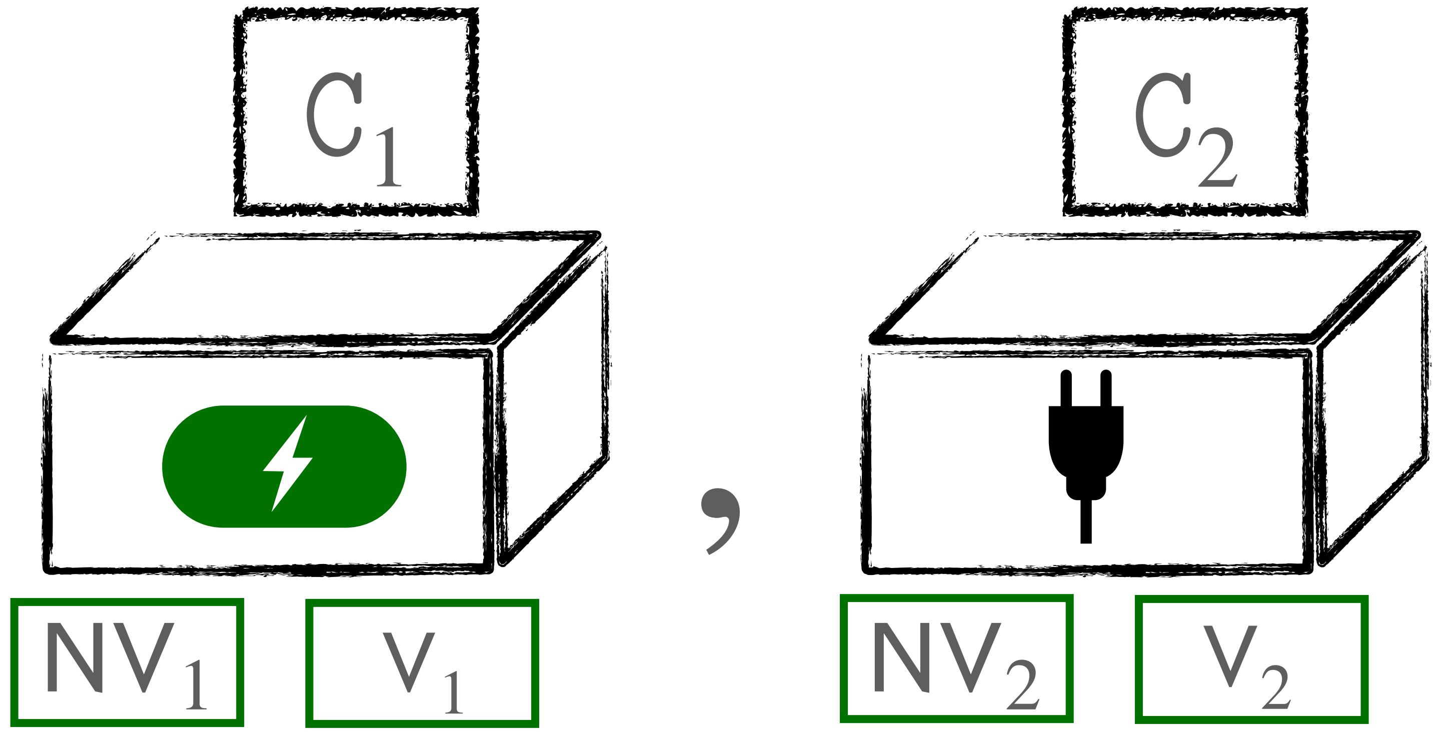


Term relation



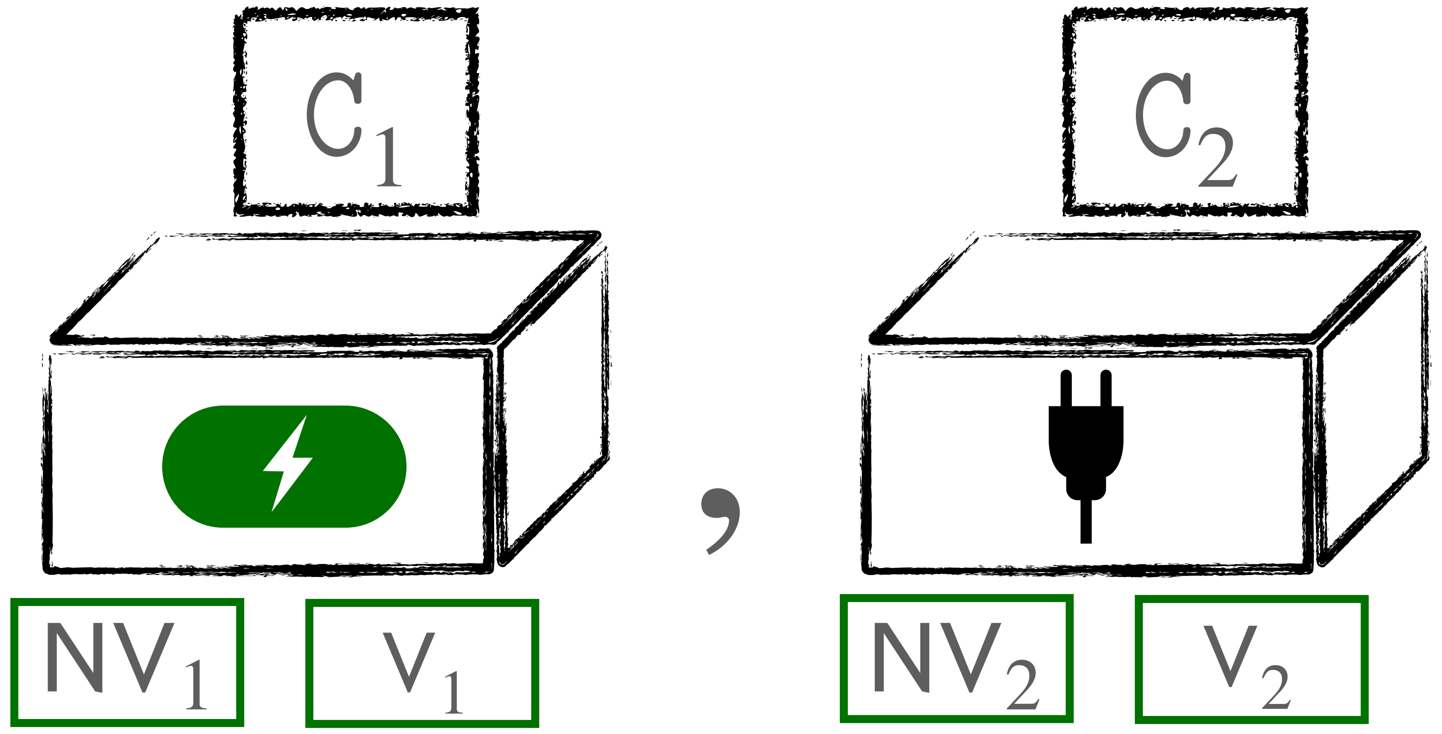
$) \in \mathscr{E}[\![\, \mathbf{C_{unit}} \,]\!]$

Term relation - base case : we observe nothing

(

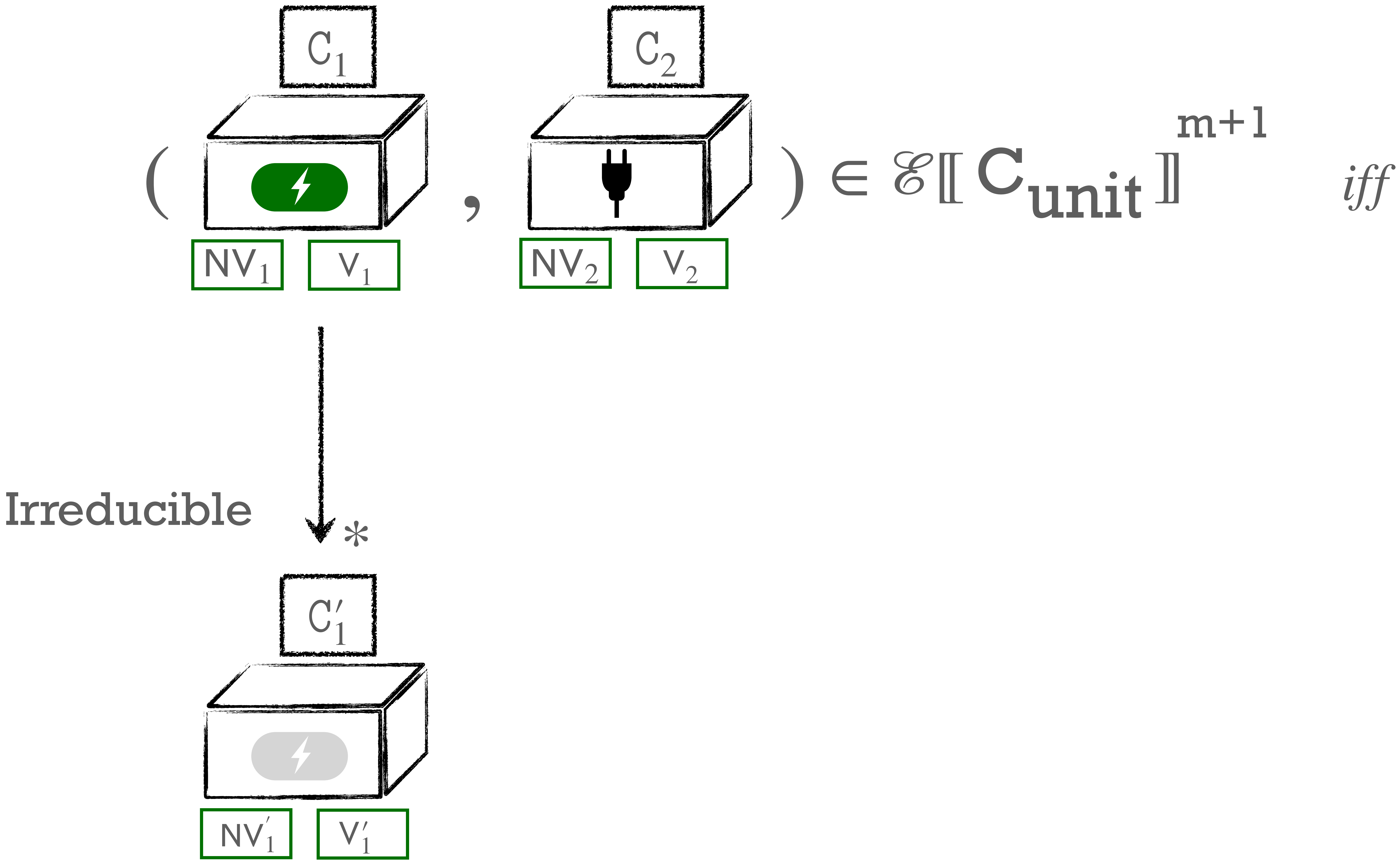
) $\in \mathcal{E}[\![\mathbf{C_{unit}}]\!]^{0}$ *always true!*

Term relation - inductive case

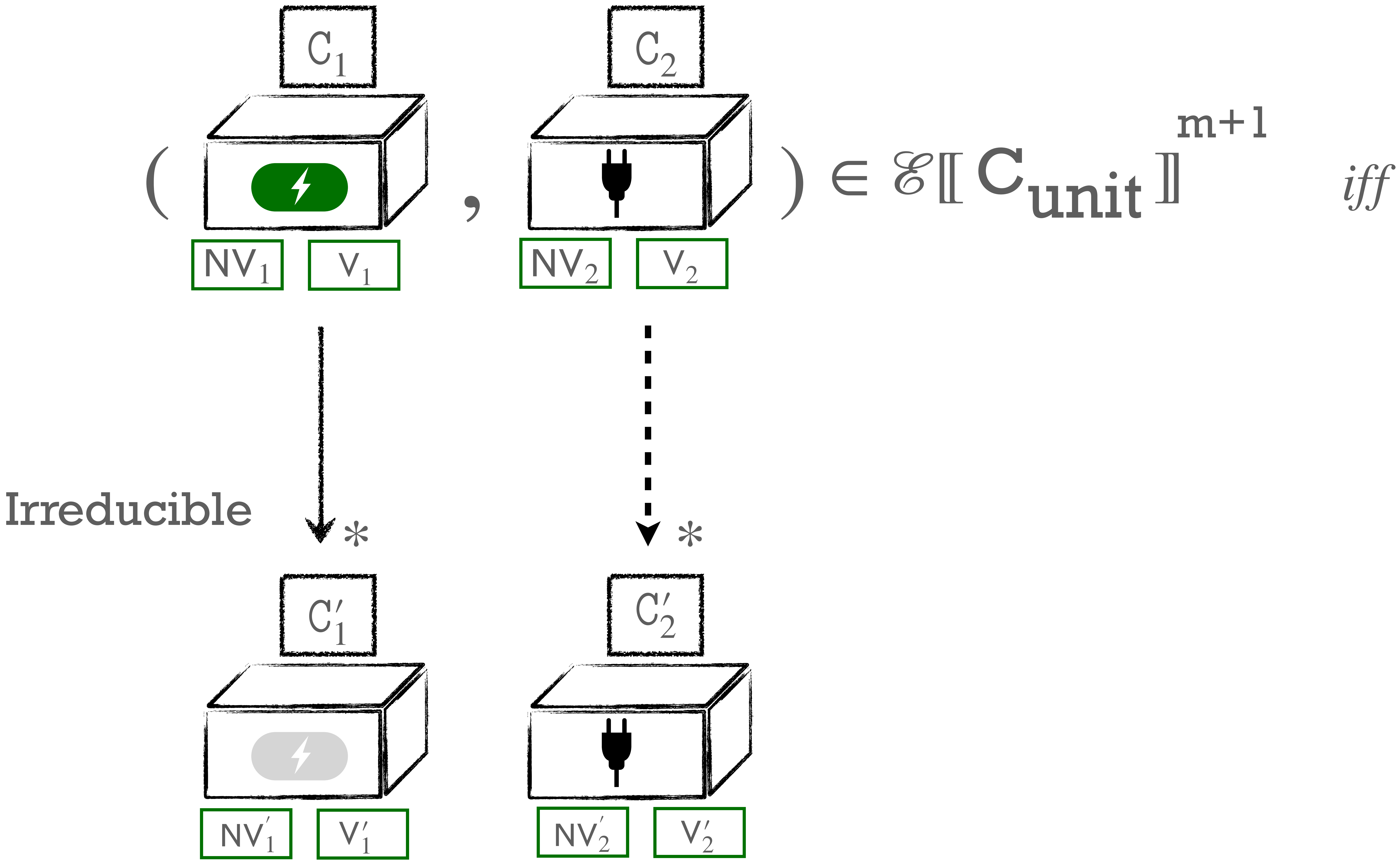


$$(\quad , \quad) \in \mathcal{E}[\![\mathbf{C}_{\mathbf{unit}}]\!]^{m+1} \qquad \textit{iff}$$

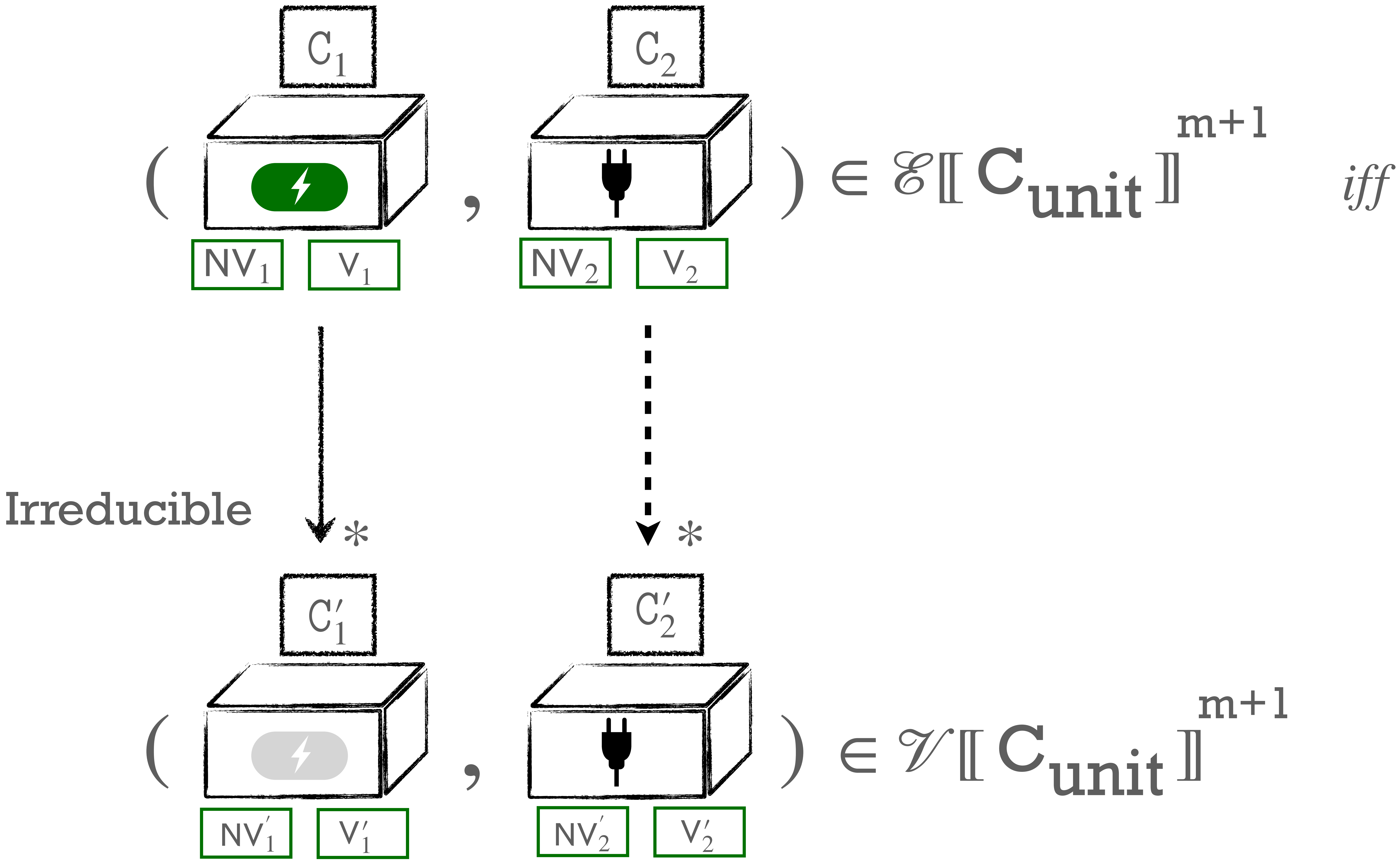
Term relation - inductive case



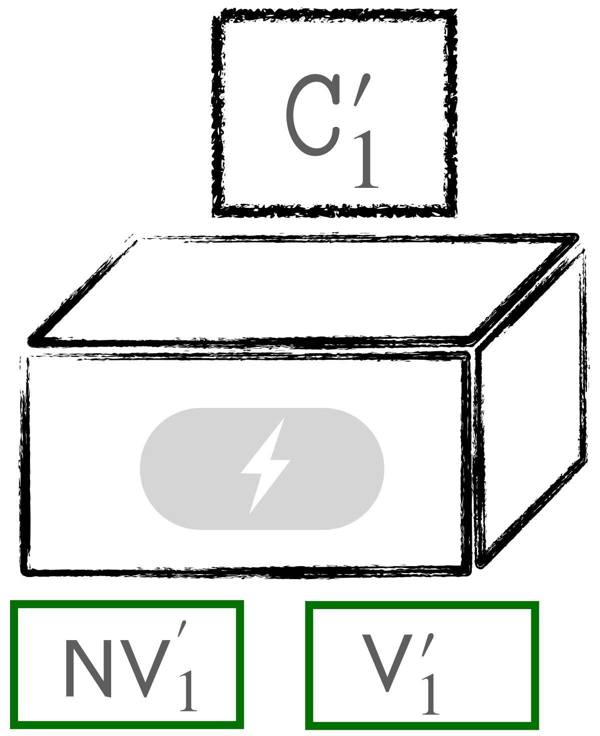
Term relation - inductive case

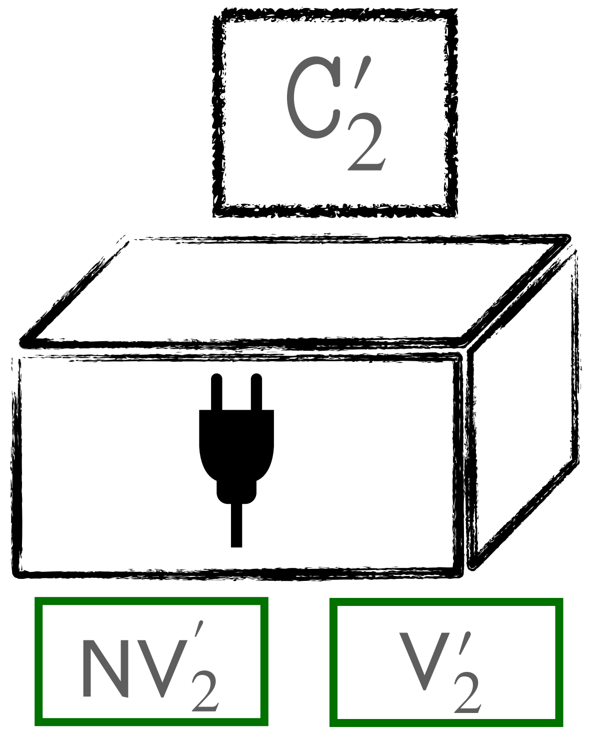


Term relation - inductive case



Value relation

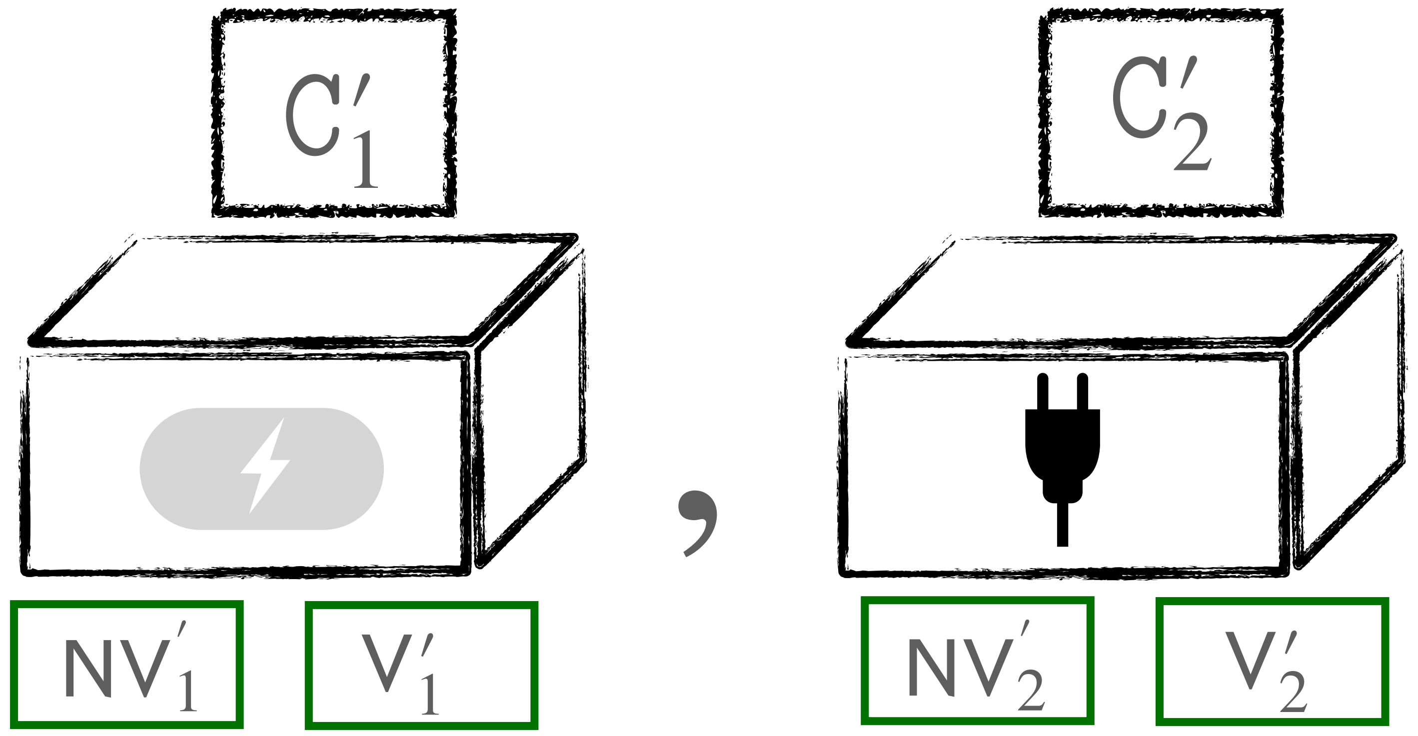
(

, 

) $\in \mathscr{V}[\![\, \mathbf{C_{unit}} \,]\!]^{\mathbf{m+1}}$ *iff*

Value relation

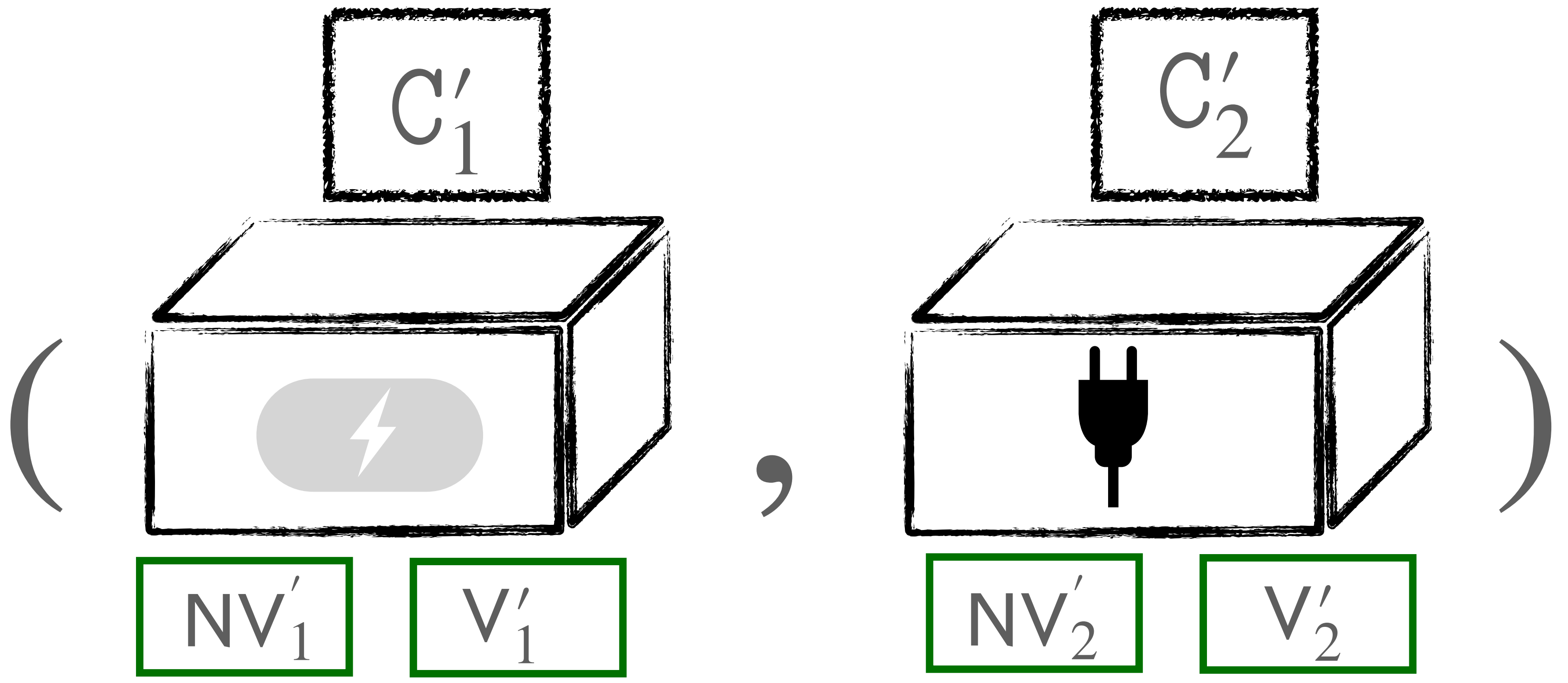
$$C_{unit} = \downarrow\uparrow \text{ unit} \vee \downarrow(\text{nat} \rightsquigarrow \uparrow C_{unit})$$



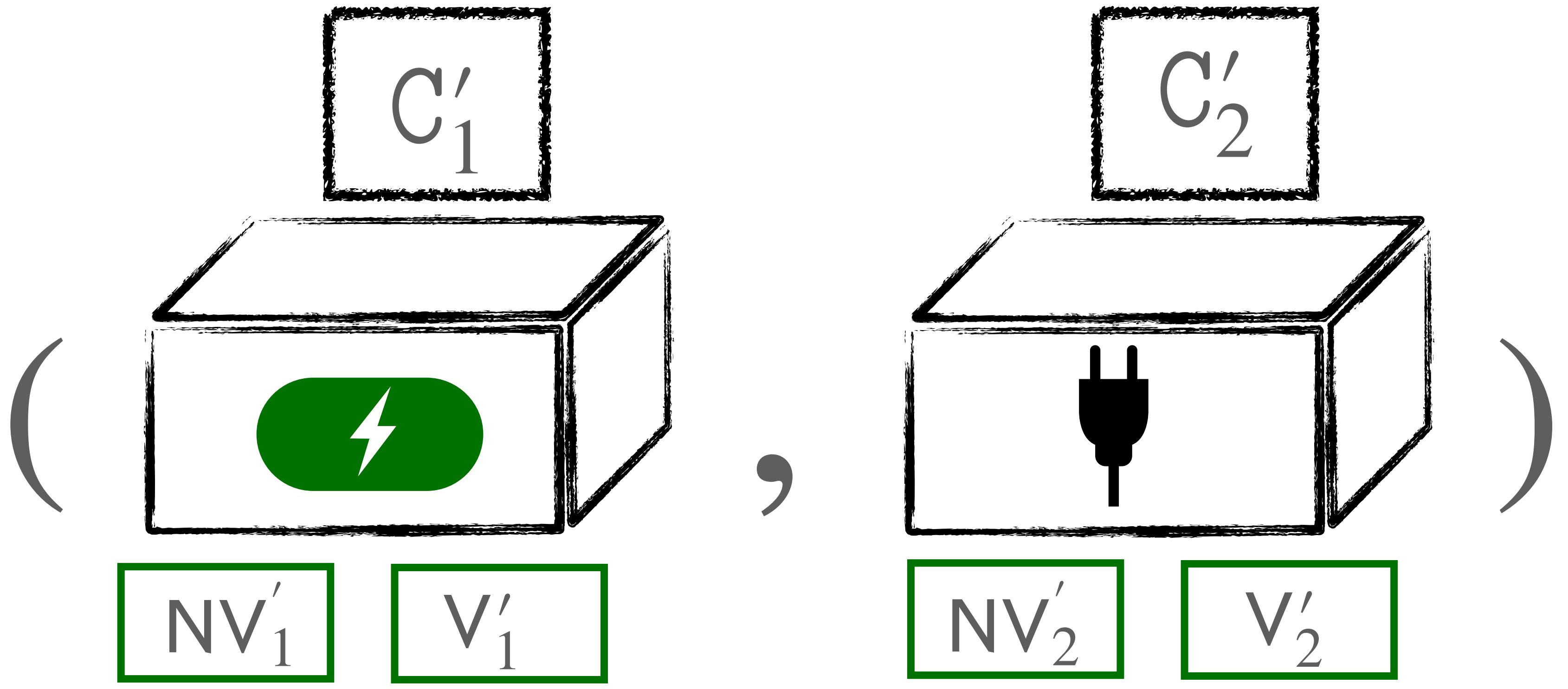
$$(\;\; , \;\;) \in \mathscr{V}[\![C_{unit}]\!]^{m+1} \quad \textit{iff}$$

Value relation

$$C_{unit} = \downarrow\uparrow \text{unit} \vee \downarrow(\text{nat} \rightsquigarrow \uparrow C_{unit})$$



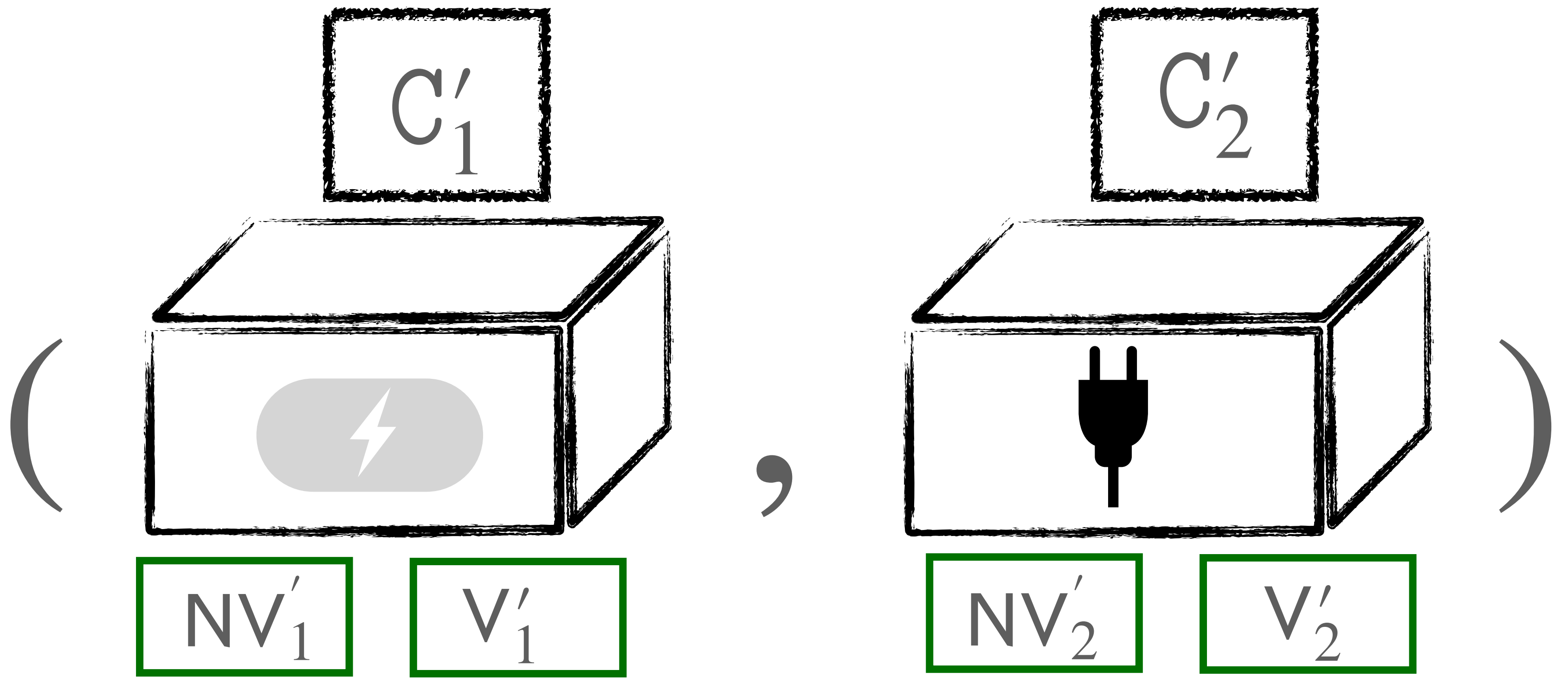
$$\in \mathscr{V}[\![C_{unit}]\!]^{m+1} \quad \textit{iff}$$



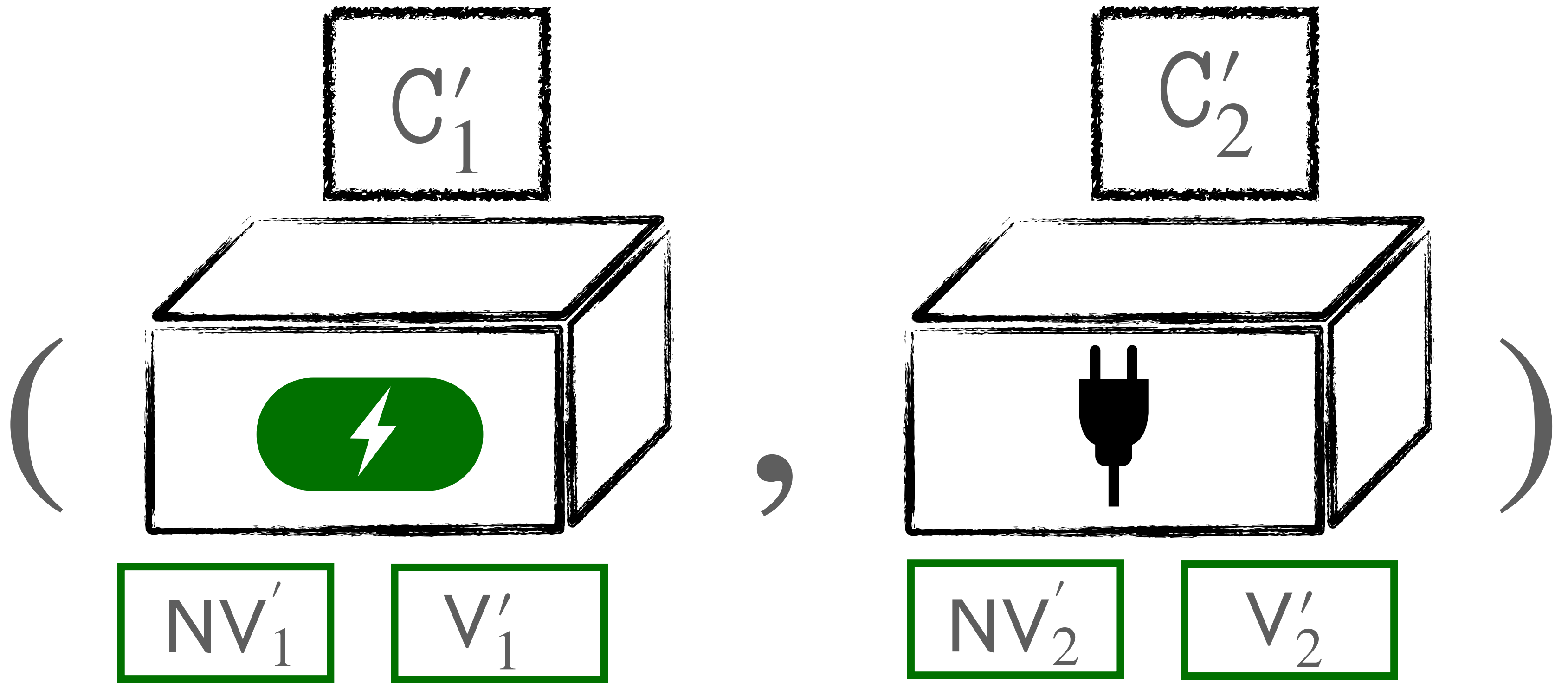
$$\in \mathscr{V}[\![\downarrow\uparrow \text{unit}]\!]^{m}$$

Value relation

$$C_{unit} = \downarrow\uparrow \text{ unit} \vee \downarrow(\text{nat} \rightsquigarrow \uparrow C_{unit})$$



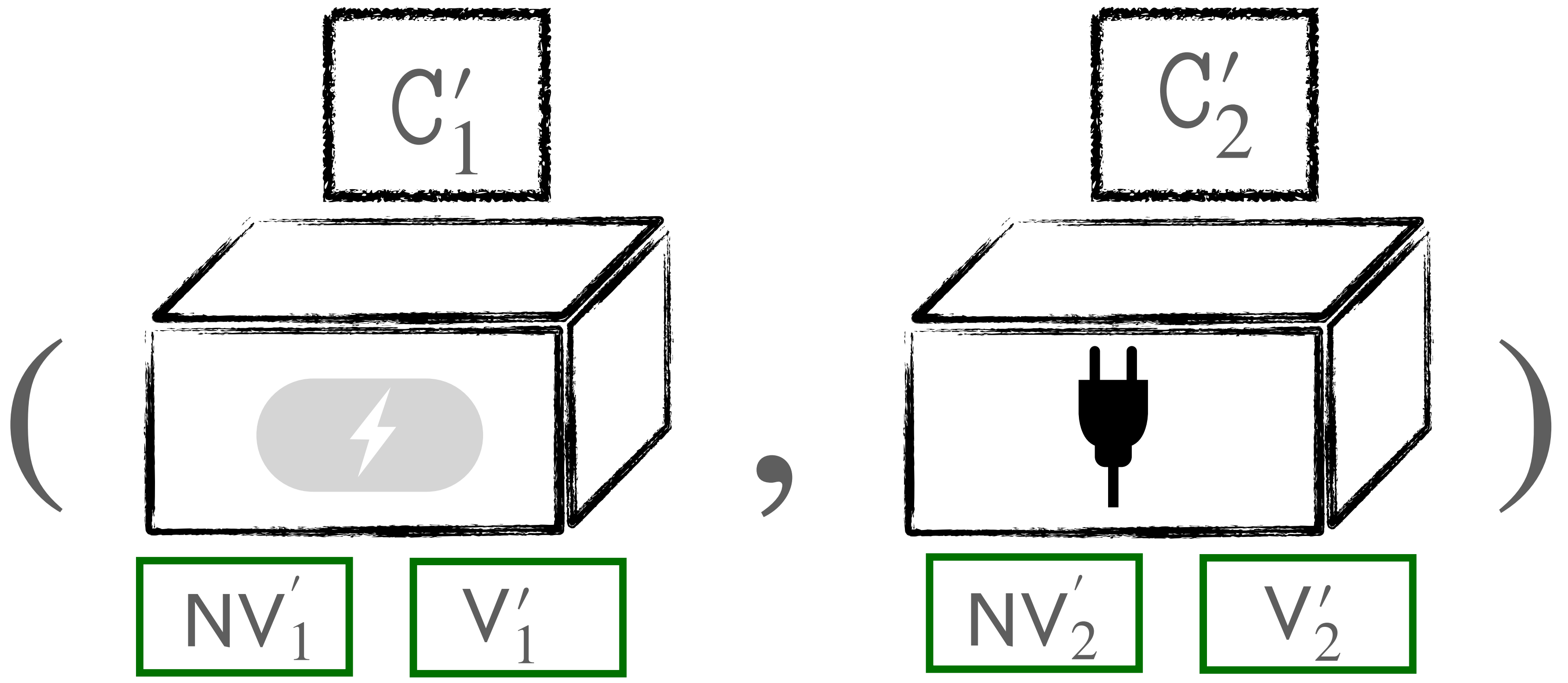
$$\in \mathscr{V}[\![C_{unit}]\!]^{m+1} \quad \textit{iff}$$



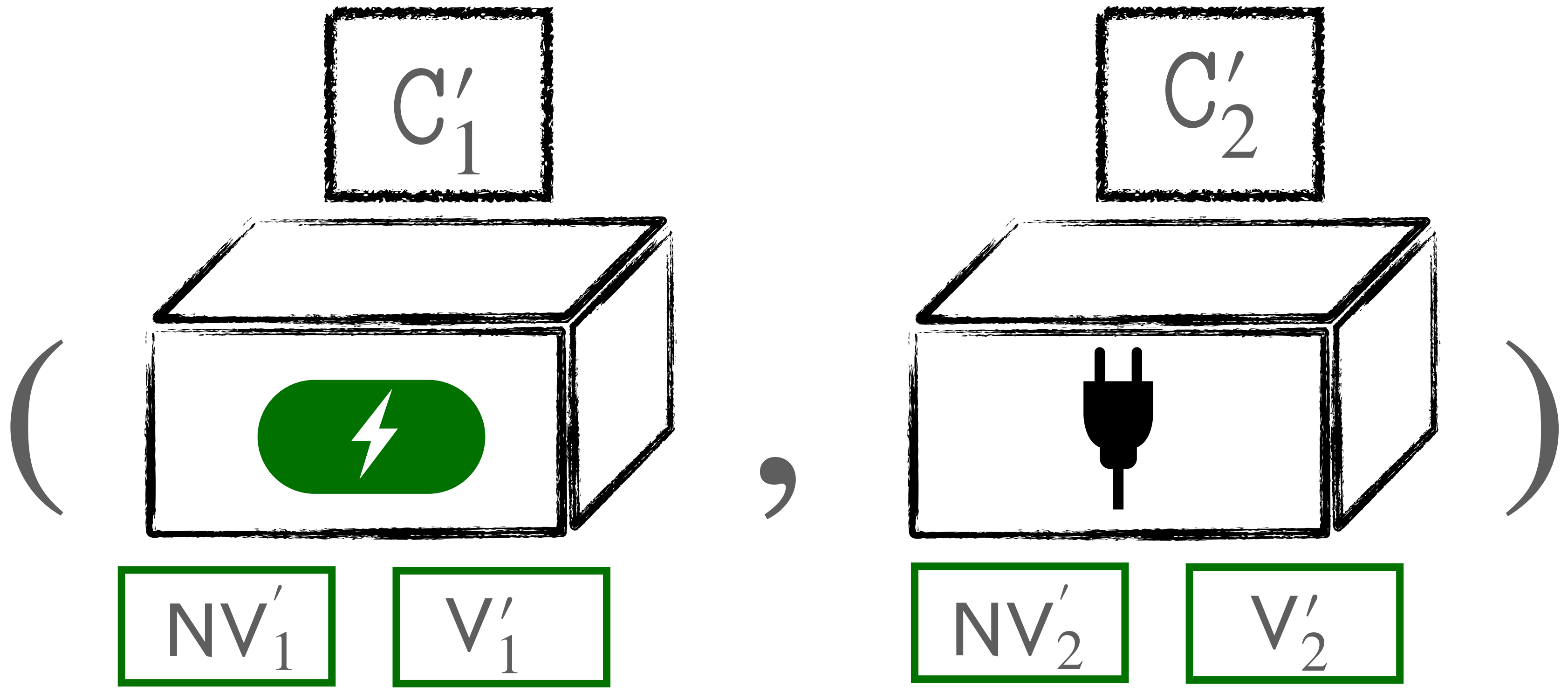
$$\in \mathscr{V}[\![\downarrow \uparrow \text{unit}]\!]^{m} \quad \textit{or}$$

Value relation

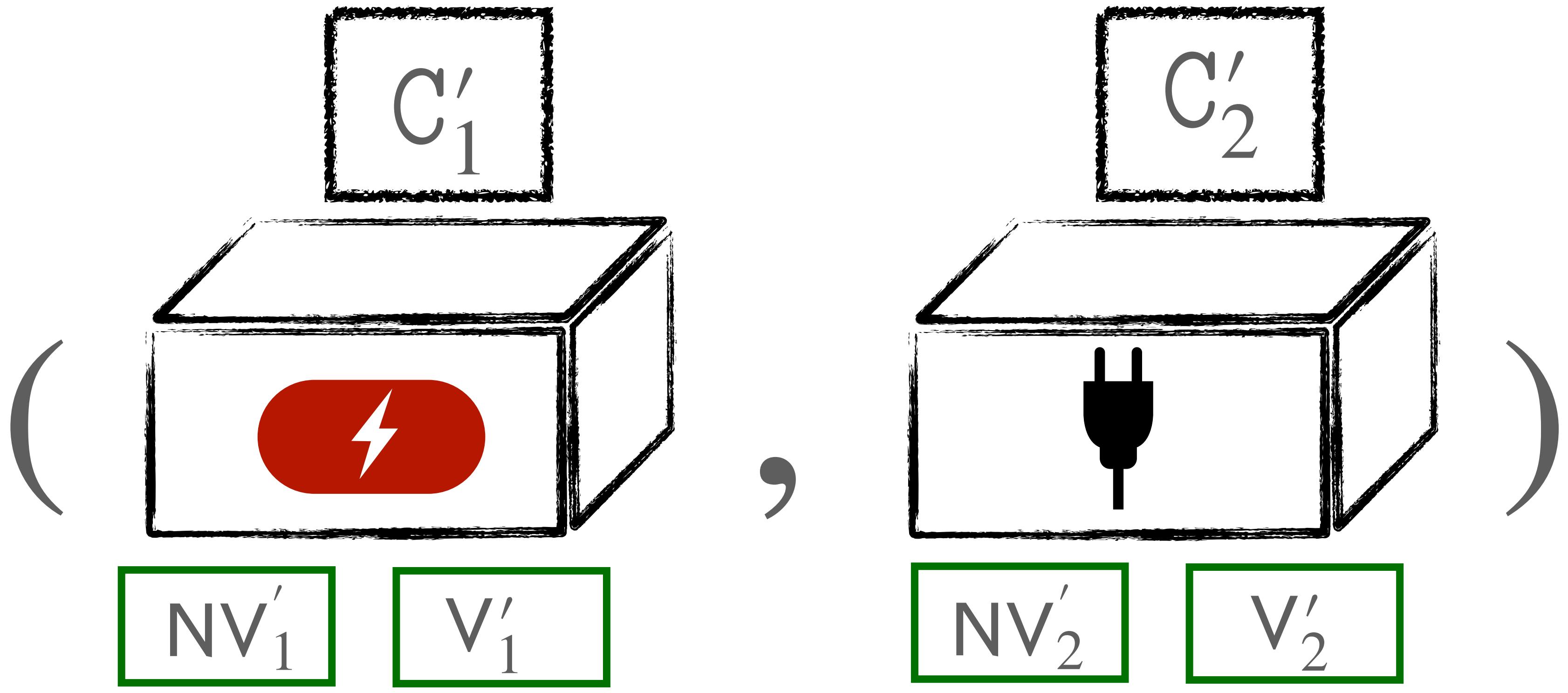
$C_{\text{unit}} = \downarrow\uparrow \text{unit} \vee \downarrow(\text{nat} \rightsquigarrow \uparrow C_{\text{unit}})$



$\in \mathcal{V}[\![C_{\text{unit}}]\!]^{m+1}$ *iff*



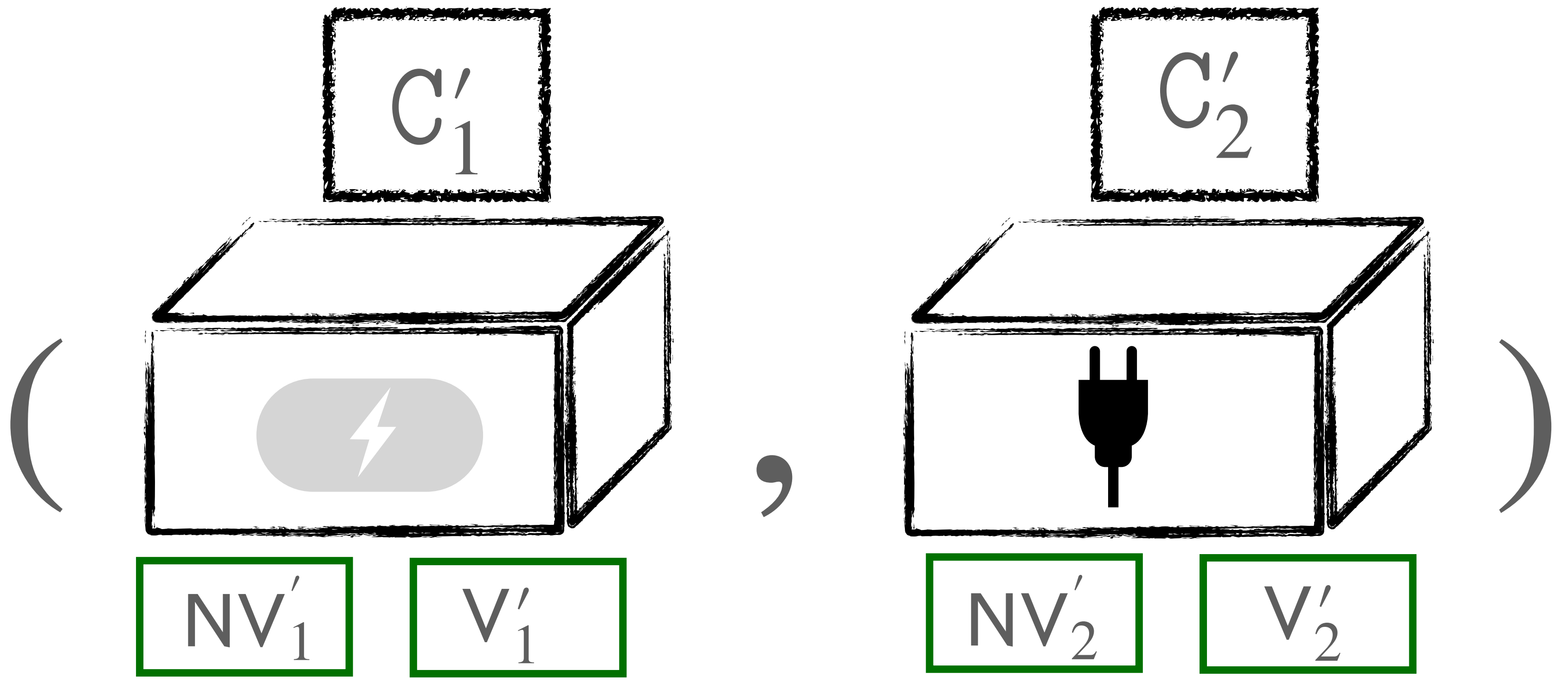
$\in \mathcal{V}[\![\downarrow \uparrow \text{unit}]\!]^{m}$ *or*



$\in \mathcal{V}[\![\downarrow (\text{nat} \rightsquigarrow \uparrow C_{\text{unit}})]\!]^{m}$

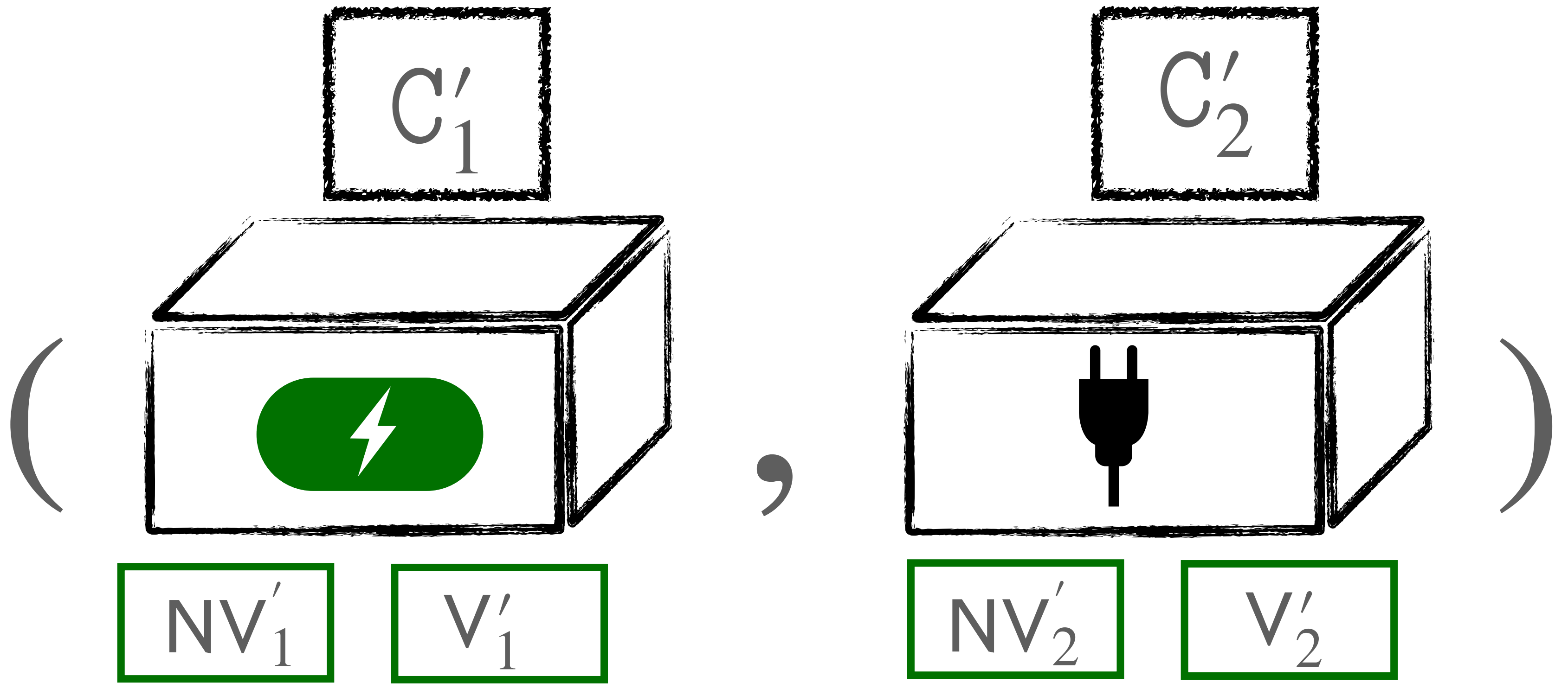
Value relation

$C_{\mathbf{unit}} = \downarrow\uparrow \mathbf{unit} \vee \downarrow(\mathbf{nat} \rightsquigarrow \uparrow C_{\mathbf{unit}})$

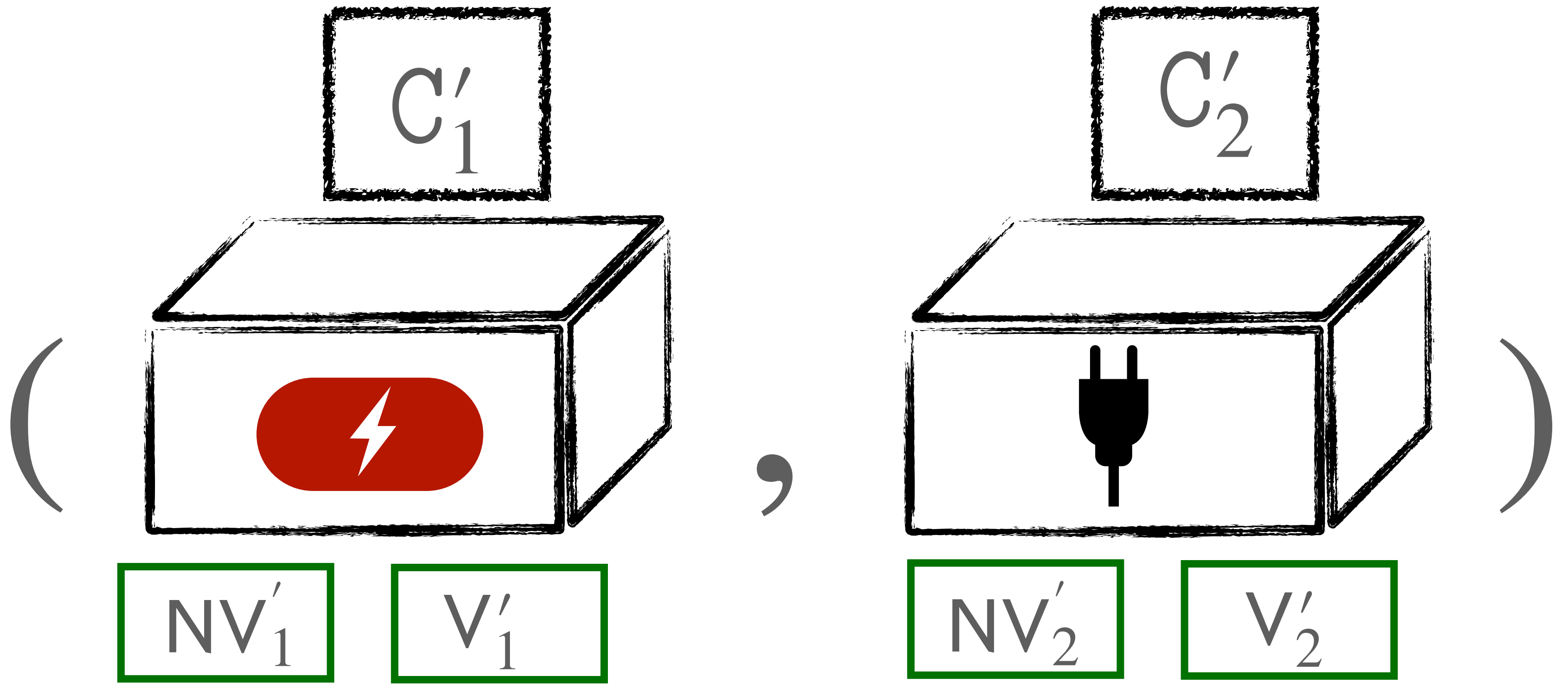


$\in \mathscr{V}[\![C_{\mathbf{unit}}]\!]^{m+1}$ *iff*

Finalize



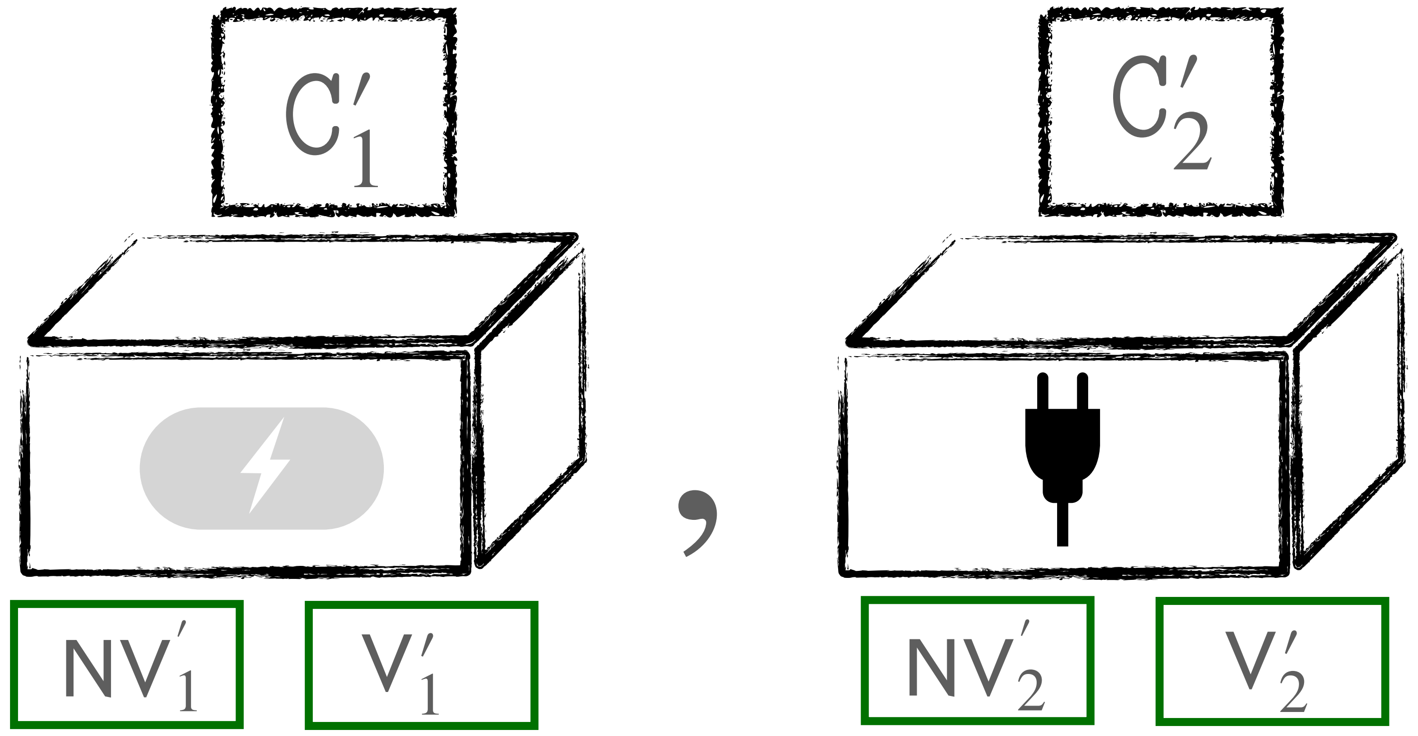
$\in \mathscr{V}[\![\downarrow\uparrow \mathbf{unit}]\!]^{m}$ *or*



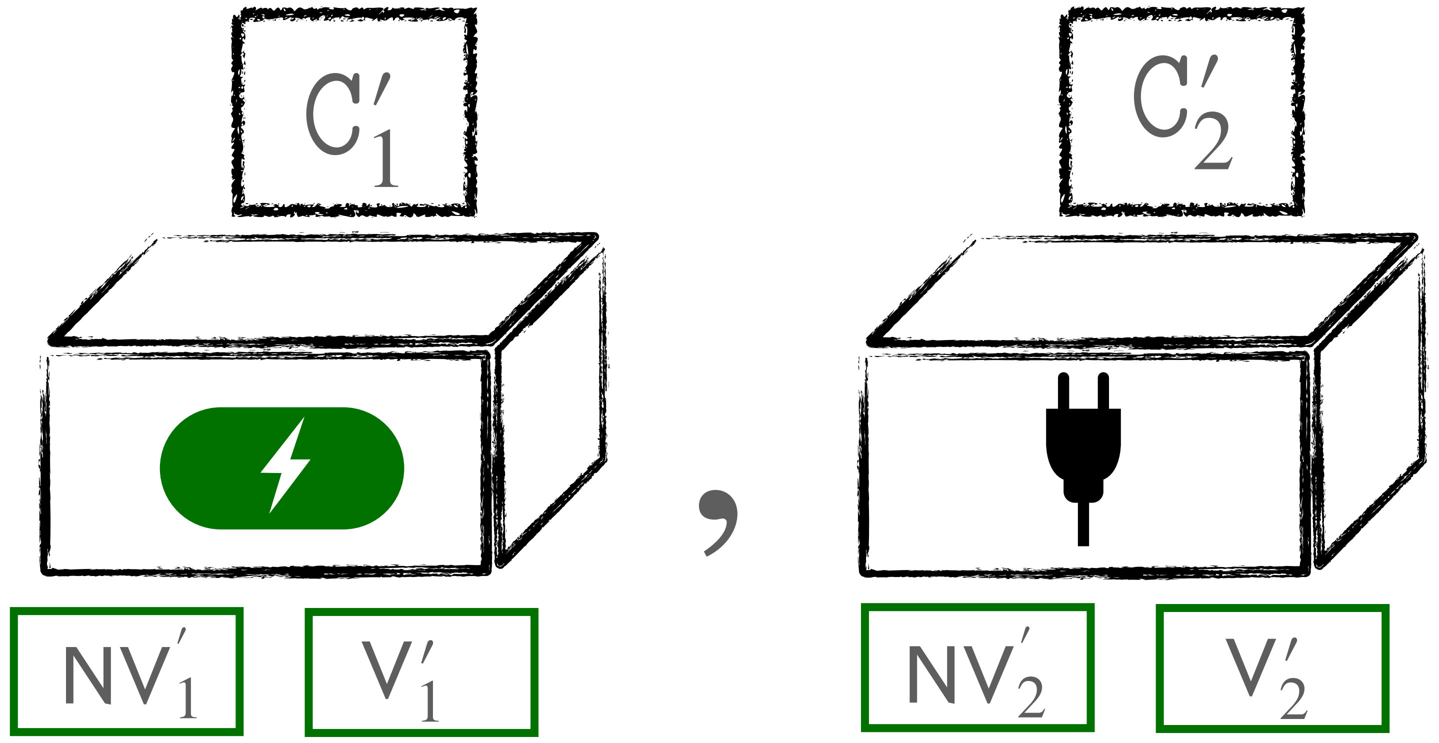
$\in \mathscr{V}[\![\downarrow(\mathbf{nat} \rightsquigarrow \uparrow C_{\mathbf{unit}})]\!]^{m}$

Value relation

$C_{unit} = \downarrow\uparrow \text{unit} \vee \downarrow(\text{nat} \rightsquigarrow \uparrow C_{unit})$



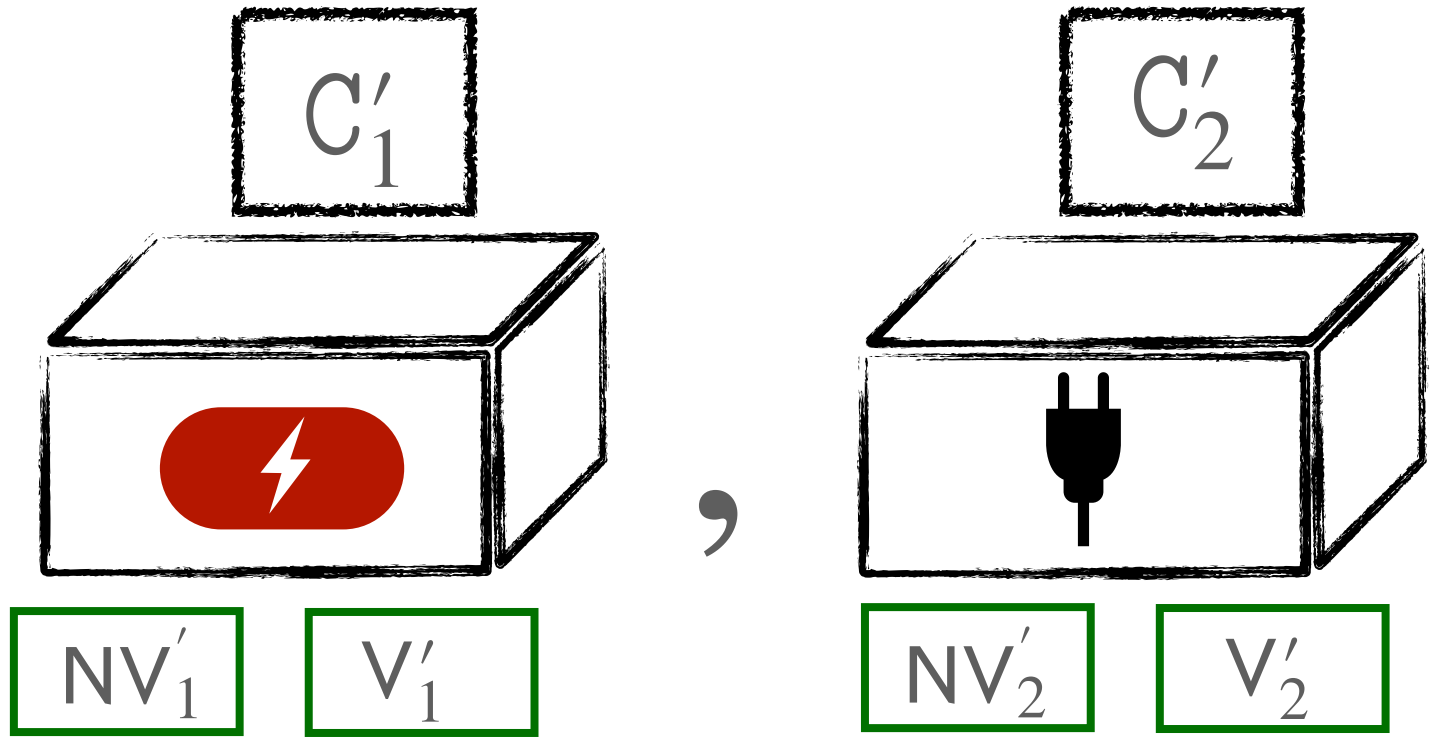
$(\ \cdot\ ,\ \cdot\ ) \in \mathscr{V}[\![C_{unit}]\!]^{m+1}$ *iff*



$(\ \cdot\ ,\ \cdot\ ) \in \mathscr{V}[\![\downarrow \uparrow \text{unit}]\!]^{m}$ *or*

Finalize

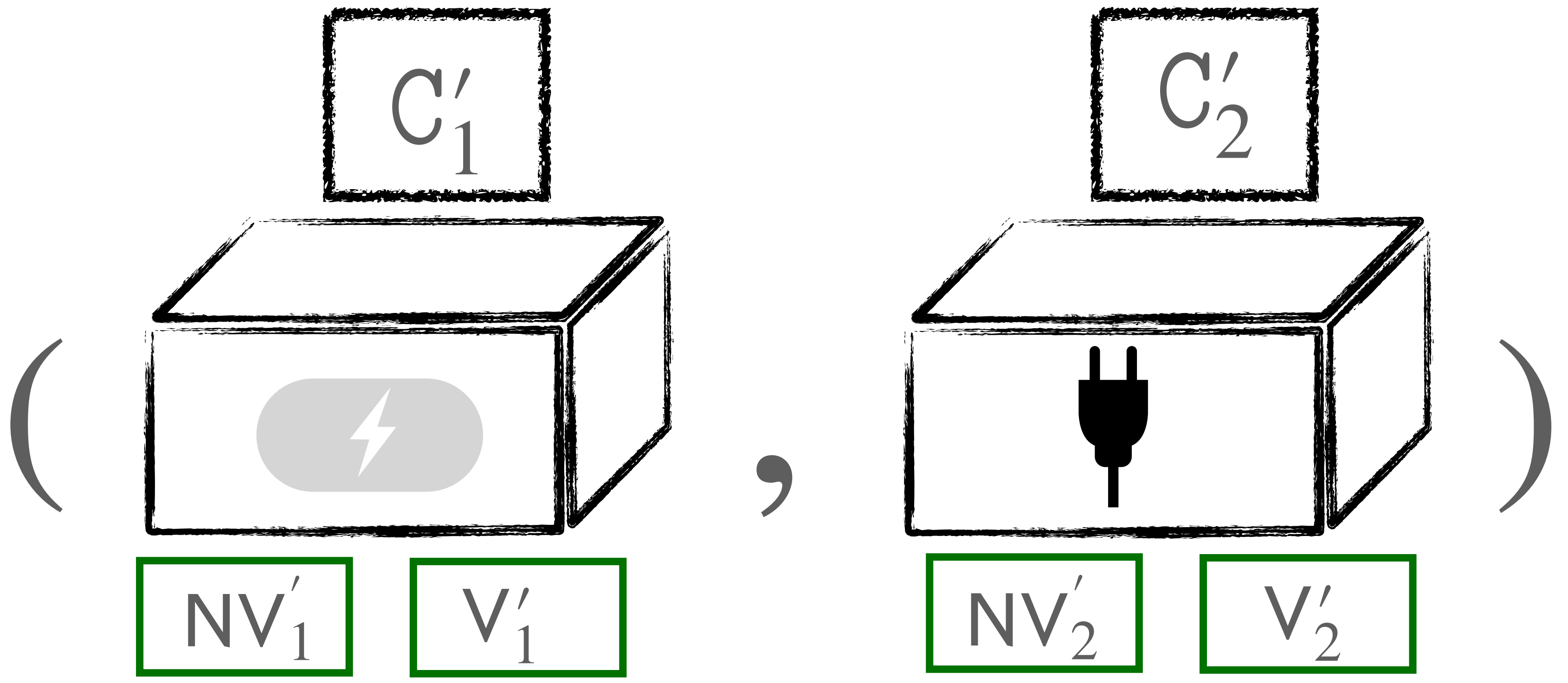
Equal final states



$(\ \cdot\ ,\ \cdot\ ) \in \mathscr{V}[\![\downarrow(\text{nat} \rightsquigarrow \uparrow C_{unit})]\!]^{m}$

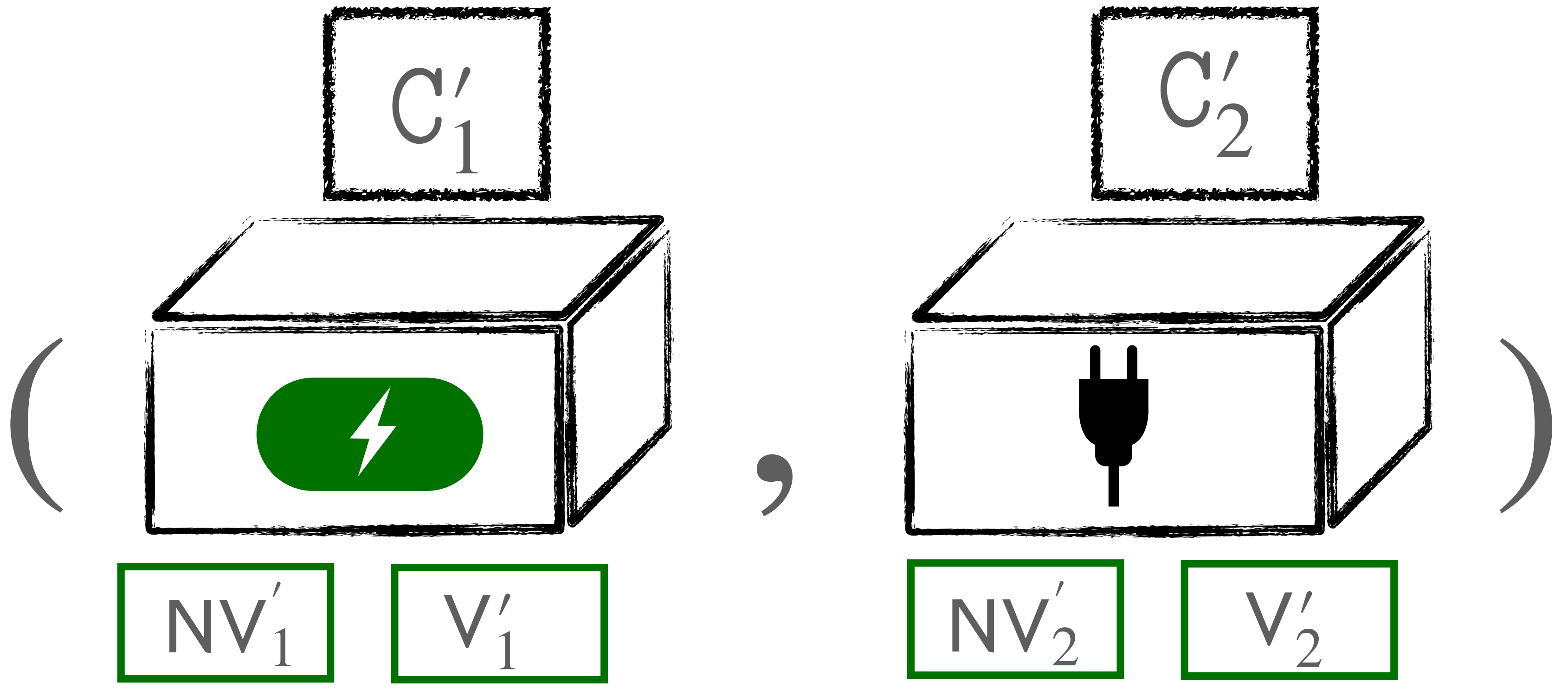
Value relation

$$\mathsf{C_{unit}} = \downarrow\uparrow \mathbf{unit} \vee \downarrow(\mathbf{nat} \rightsquigarrow \uparrow \mathsf{C_{unit}})$$



$$\in \mathscr{V}[\![\, \mathsf{C_{unit}} \,]\!]^{m+1} \quad \textit{iff}$$

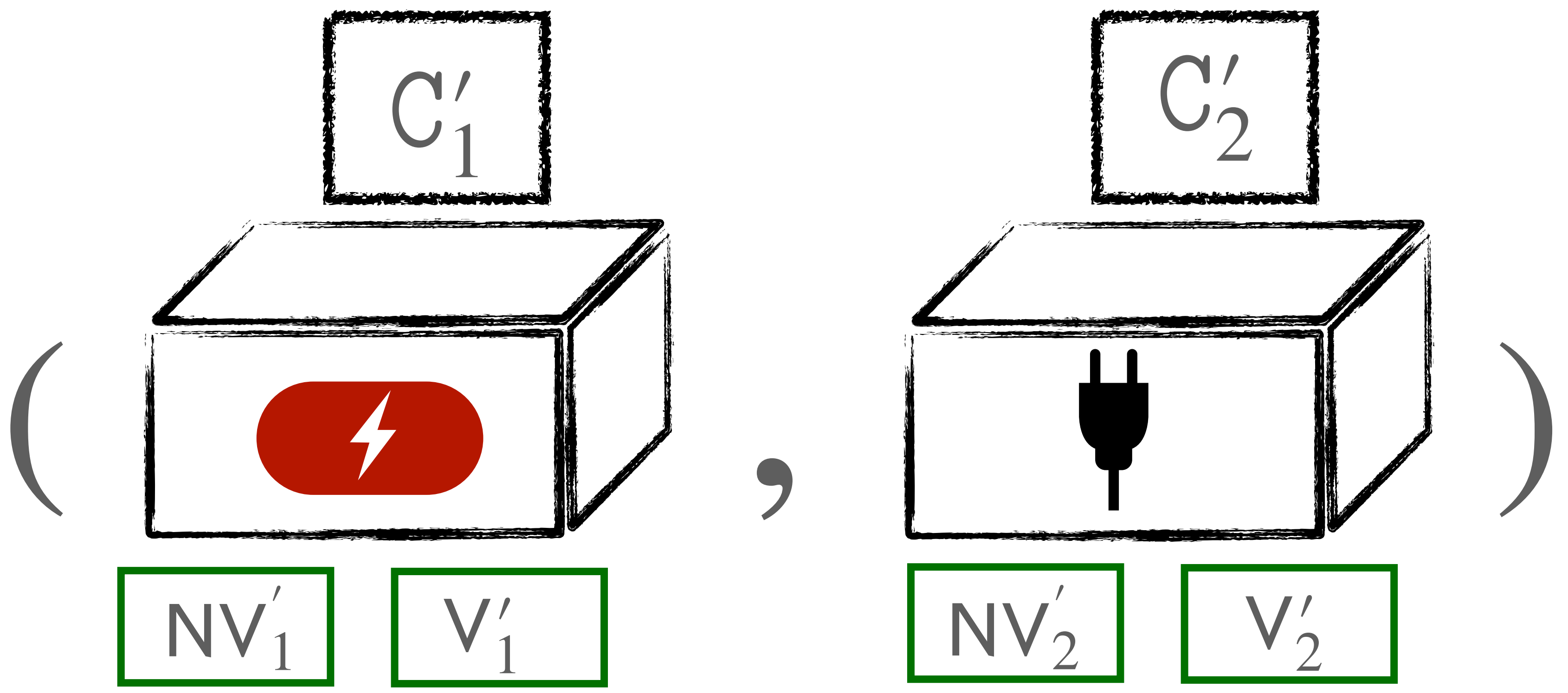
Finalize



$$\in \mathscr{V}[\![\, \downarrow \uparrow \mathbf{unit} \,]\!]^{m} \quad \textit{or}$$

Equal final states

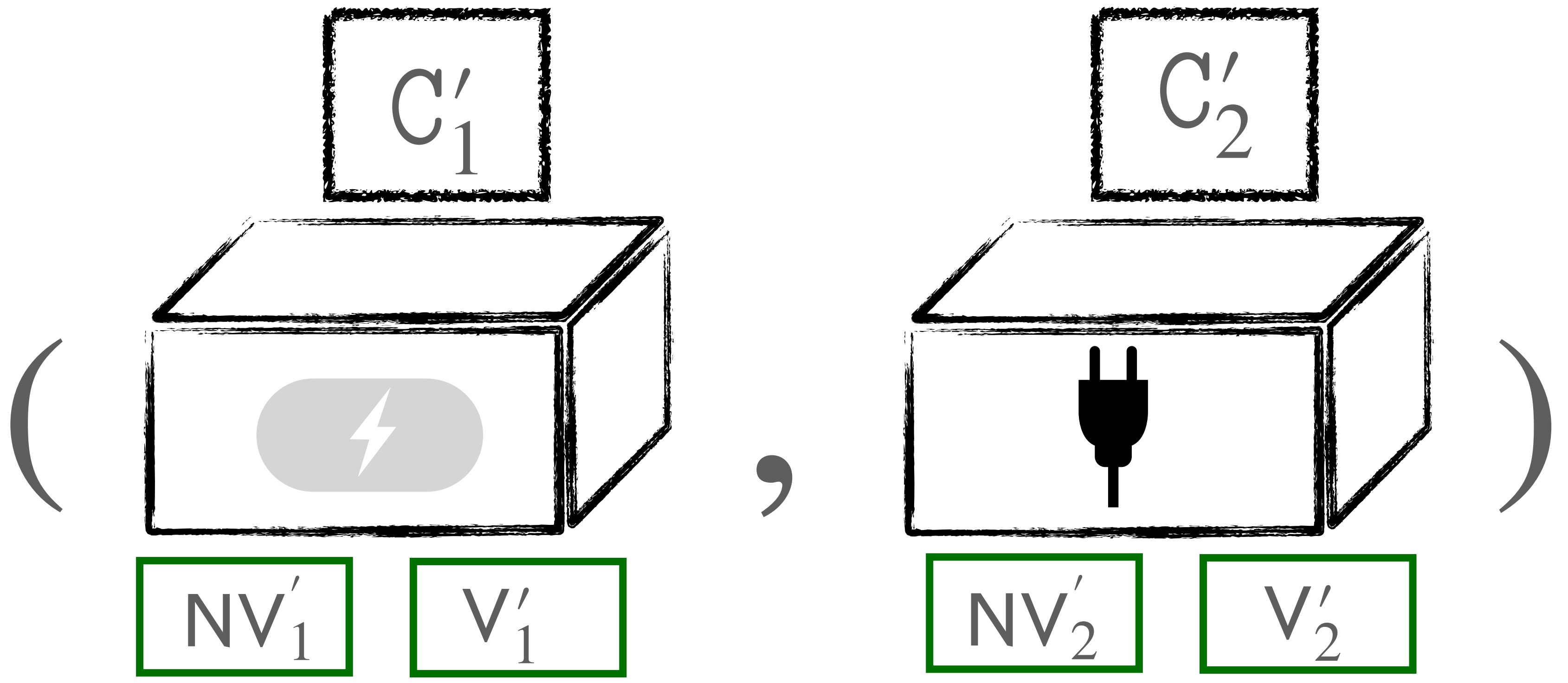
Wipe out



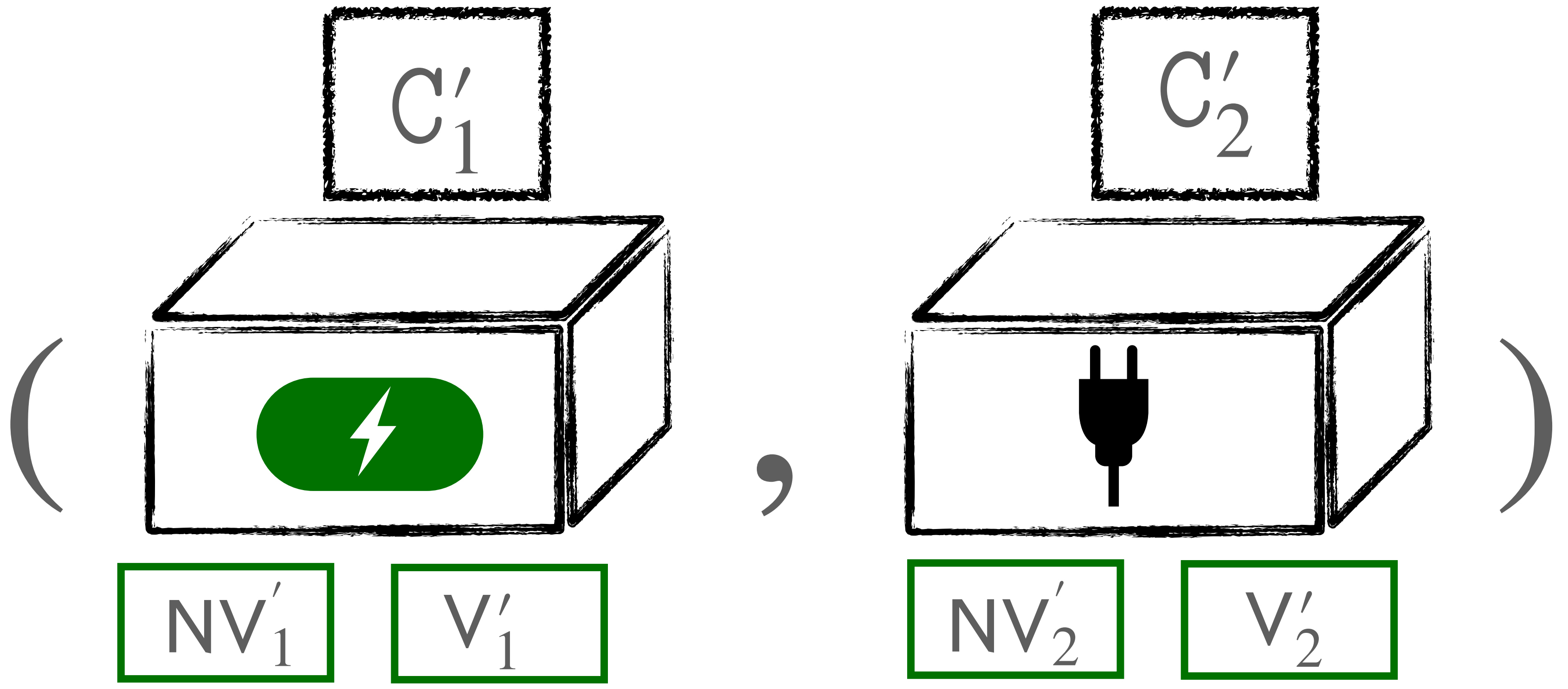
$$\in \mathscr{V}[\![\, \downarrow (\mathbf{nat} \rightsquigarrow \uparrow \mathsf{C_{unit}}) \,]\!]^{m}$$

Value relation

$C_{\text{unit}} = \downarrow\uparrow \text{unit} \vee \downarrow(\text{nat} \rightsquigarrow \uparrow C_{\text{unit}})$



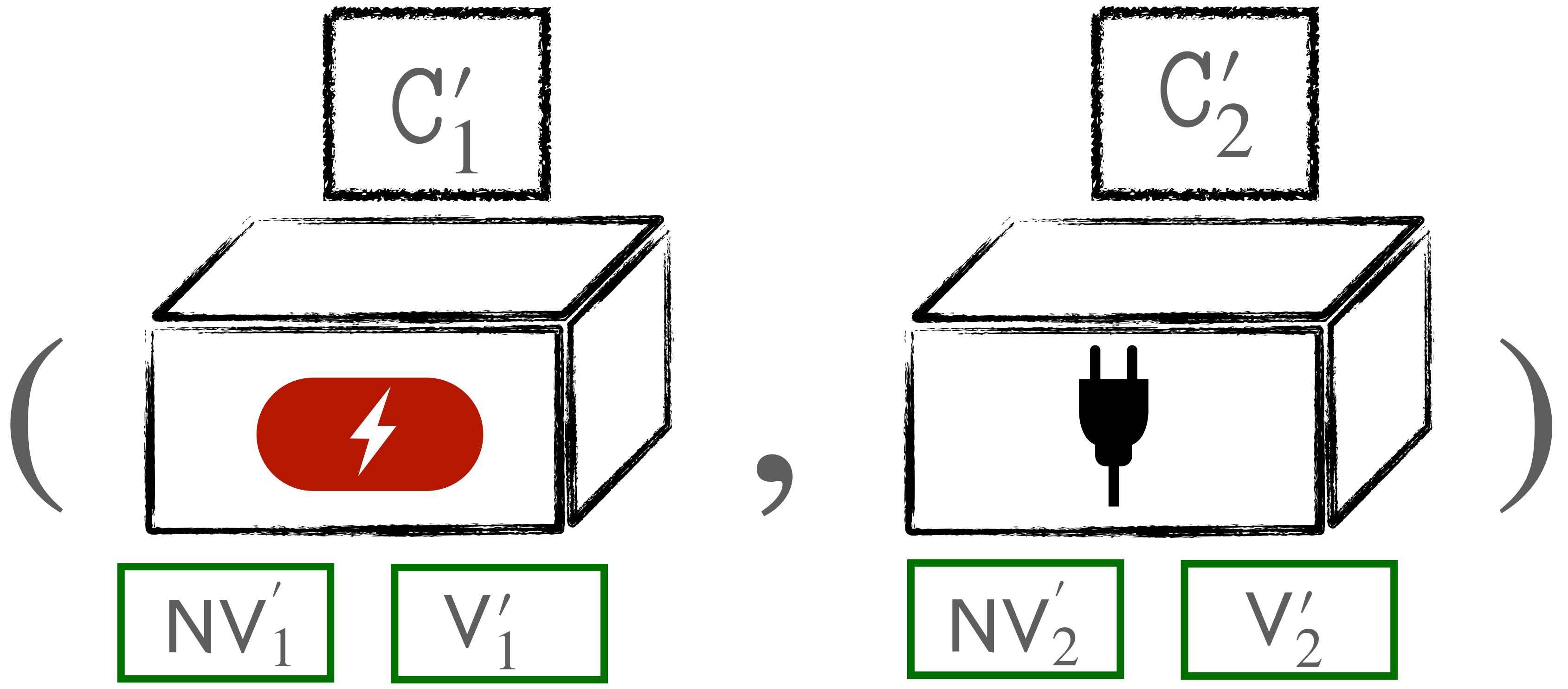
$\in \mathscr{V}[\![C_{\text{unit}}]\!]^{m+1}$ *iff*



Finalize

$\in \mathscr{V}[\![\downarrow \uparrow \text{unit}]\!]^{m}$ *or*

Equal final states



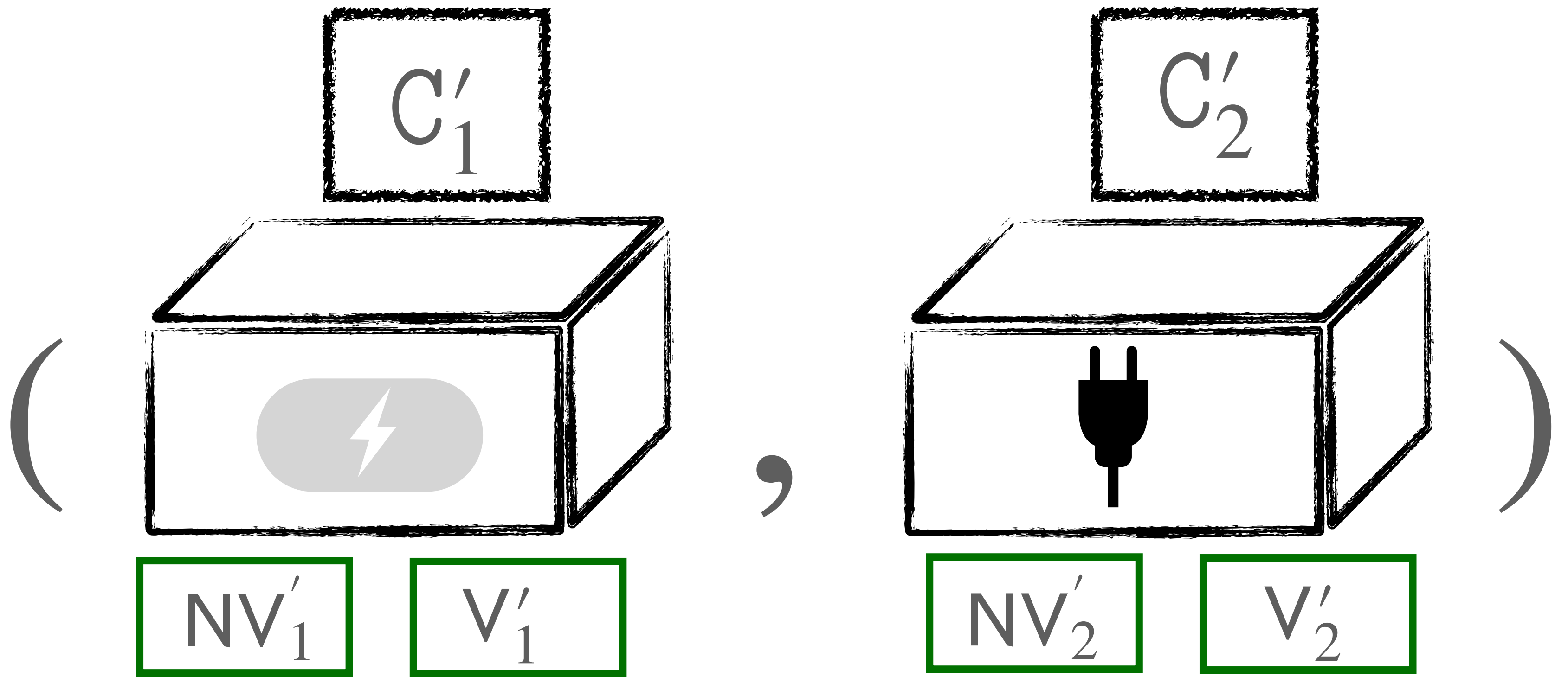
Wipe out

$\in \mathscr{V}[\![\downarrow (\text{nat} \rightsquigarrow \uparrow C_{\text{unit}})]\!]^{m}$

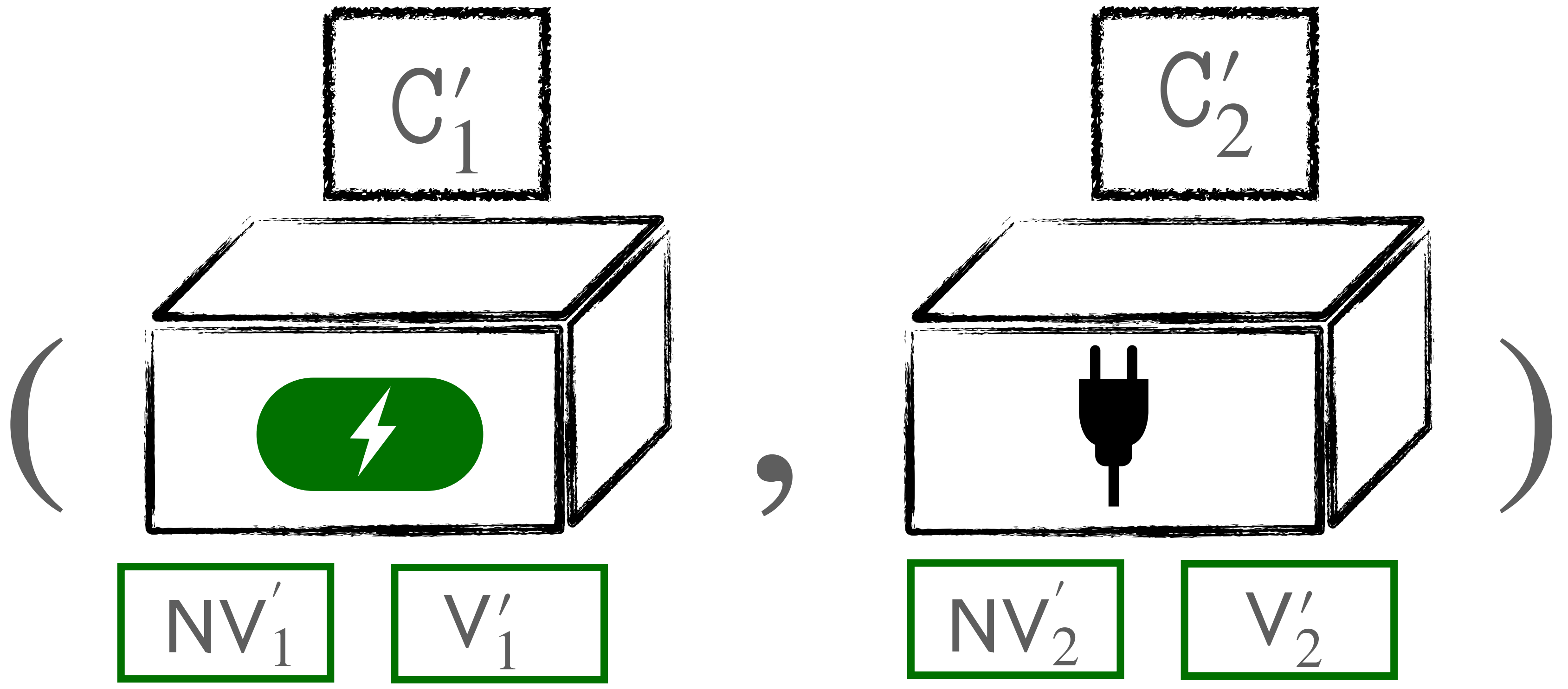
Energy input

Value relation

$$C_{unit} = \downarrow\uparrow \text{unit} \vee \downarrow(\text{nat} \rightsquigarrow \uparrow C_{unit})$$



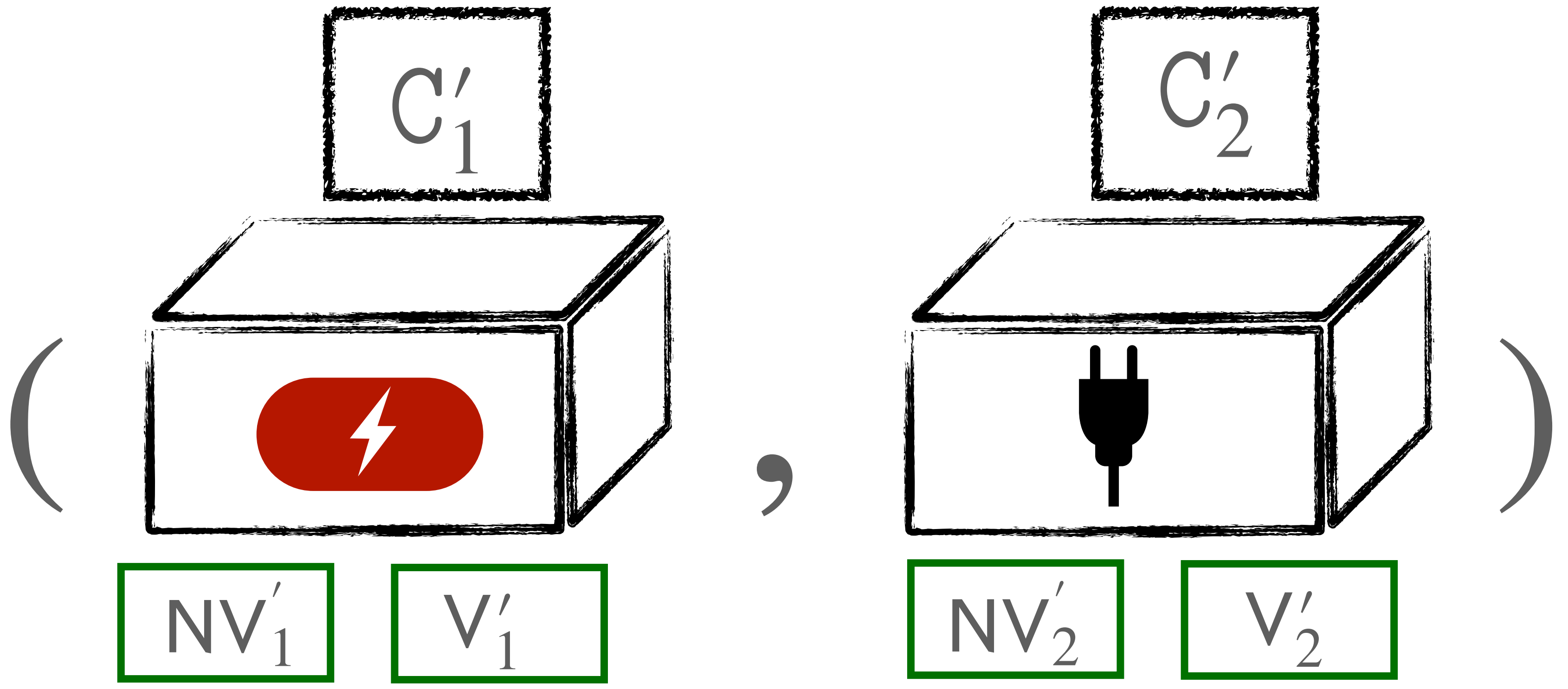
$$\in \mathscr{V}[\![C_{unit}]\!]^{m+1} \quad \textit{iff}$$



Finalize

$$\in \mathscr{V}[\![\downarrow \uparrow \text{unit}]\!]^{m} \quad \textit{or}$$

Equal final states



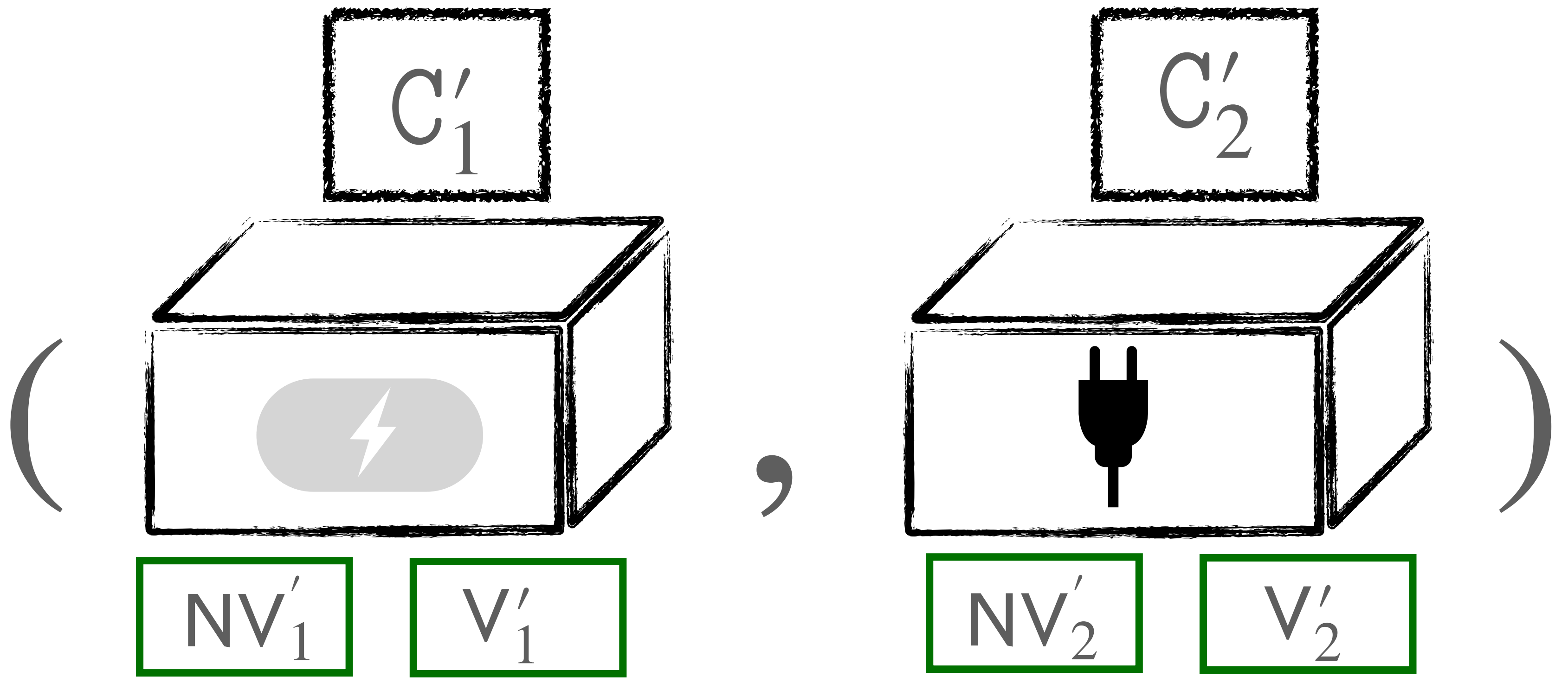
Wipe out Restore

$$\in \mathscr{V}[\![\downarrow(\text{nat} \rightsquigarrow \uparrow C_{unit})]\!]^{m}$$

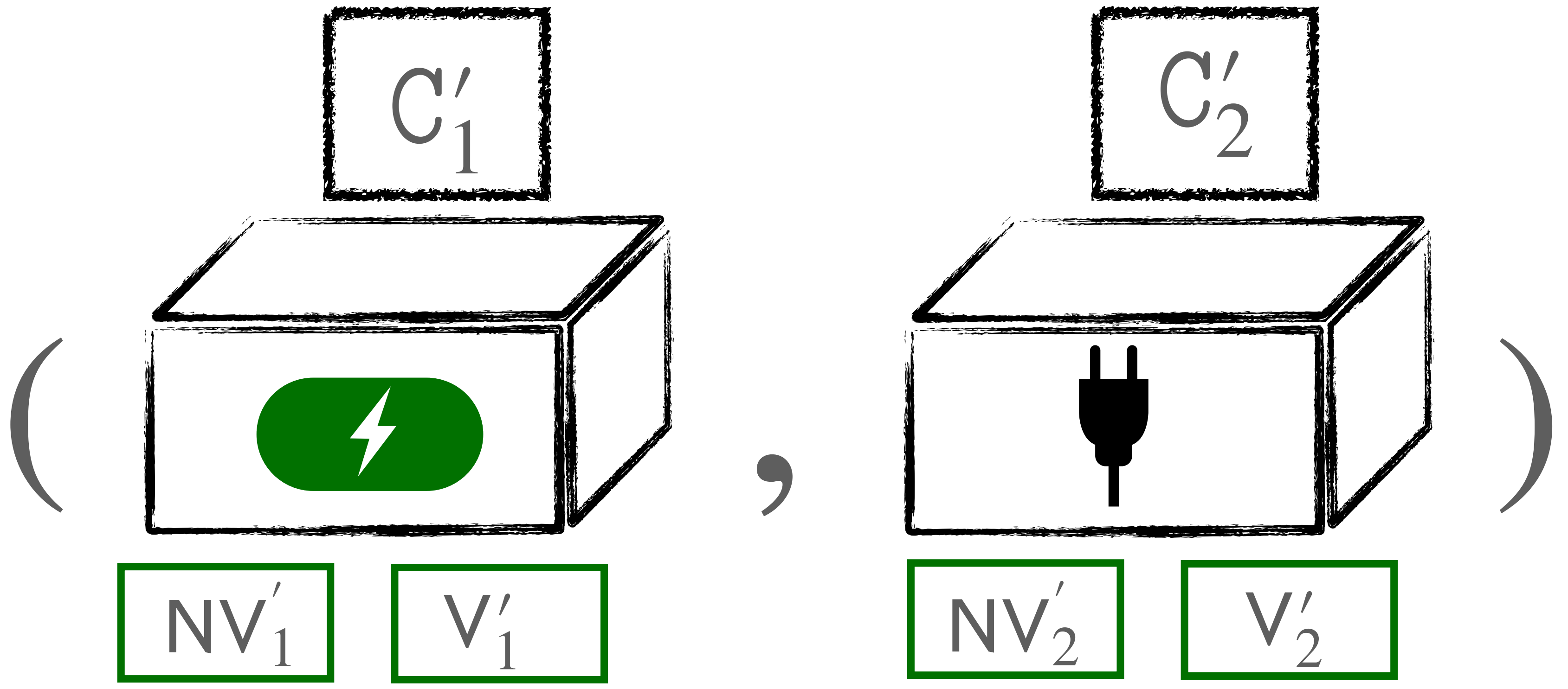
Energy input

Value relation

$$C_{unit} = \downarrow\uparrow \text{unit} \vee \downarrow(\text{nat} \rightsquigarrow \uparrow C_{unit})$$



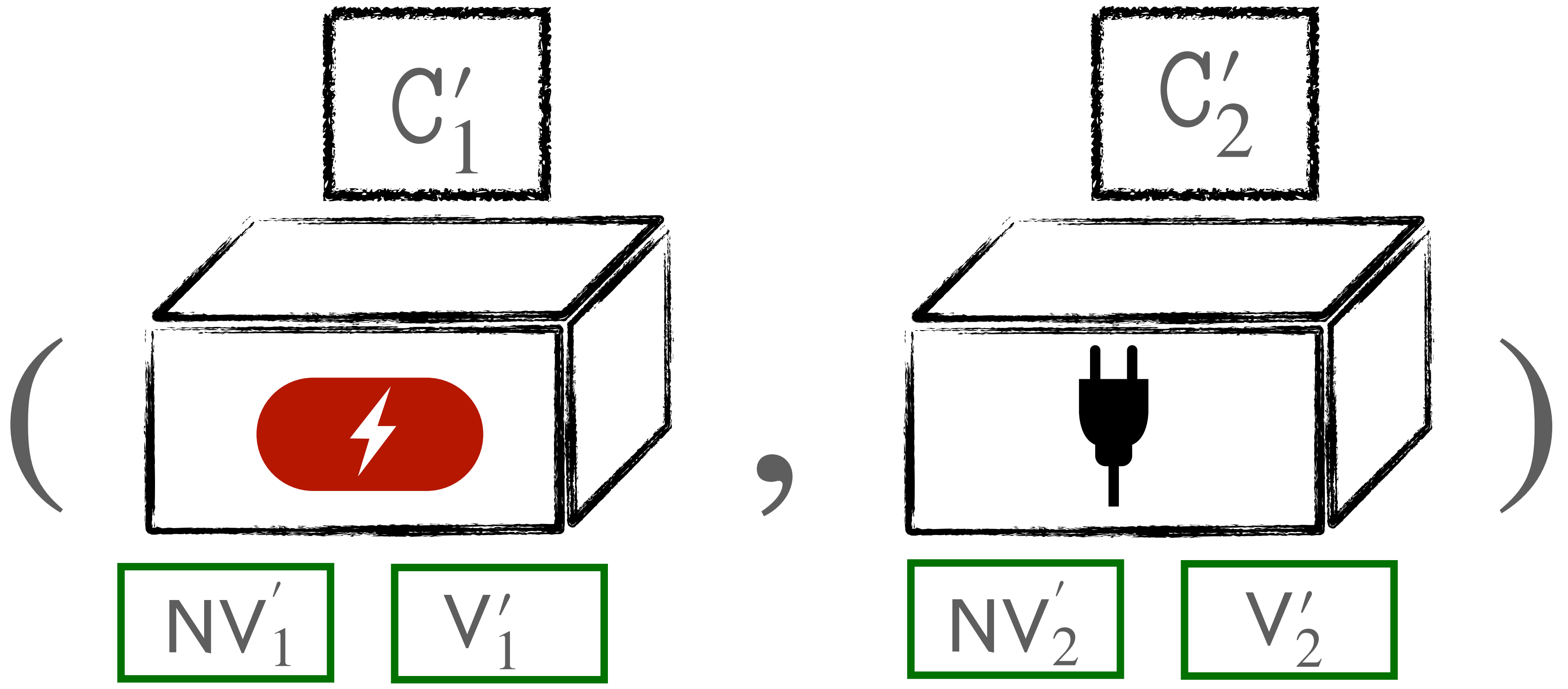
$$\in \mathscr{V}[\![C_{unit}]\!]^{m+1} \quad \textit{iff}$$



Finalize

$$\in \mathscr{V}[\![\downarrow \uparrow \text{unit}]\!]^{m} \quad \textit{or}$$

Equal final states



Wipe out Restore

$$\in \mathscr{V}[\![\downarrow(\text{nat} \rightsquigarrow \uparrow C_{unit})]\!]^{m}$$

Energy input Call term relation (index m)

Main results

- Fundamental theorem: Syntactically well-typed programs are semantically well-typed.
- Adequacy theorem: Semantically well-typed programs are correct.

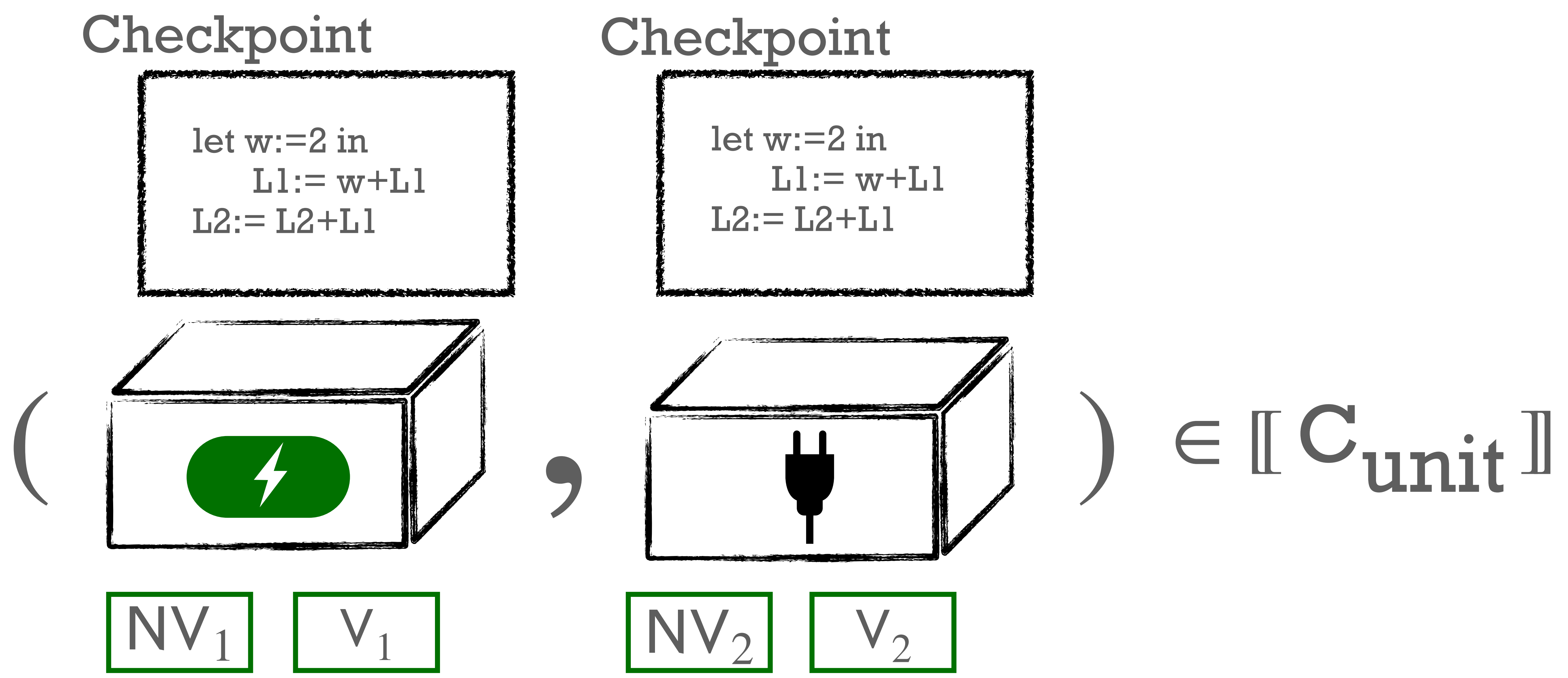
Every intermittent execution of well-typed programs can be simulated by a continuous execution of them.

Summary

- A logical interpretation of intermittent execution.
- Crash types to specify how stable and unstable portions interact.
- A core calculus for Crash types.
- A logical relation for correctness of intermittent executions.

Future work

- Not all variables need to be checkpointed
- Shared memory concurrency



Extras

$$\mathtt{Md} \mid b \geq 0 : \mathbf{nat} \mid \Omega \mid \Sigma \Vdash c_1 \leq c_2 : \mathtt{C}_{\mathtt{unit}}$$

$$\text{iff } \forall n, m \geq 0.\ \forall \gamma, \mathsf{NV}, \mathsf{V}.s.t.\ \mathsf{NV} \mid \mathsf{V} \Vdash \gamma :: \Omega \mid \Sigma.$$

$$(\gamma \mid \mathtt{Md} \mid n \mid \mathsf{NV} \mid \mathsf{V} \mid c_1, \gamma \mid \mathtt{Md} \mid \infty \mid \mathsf{NV} \mid \mathsf{V} \mid c_2) \in \mathcal{E}[\![\mathtt{C}_{\mathtt{unit}}]\!]^m$$

Term Relation

$$\begin{aligned}
\mathcal{E}[\![\mathtt{C}_{\mathtt{unit}}]\!]^{m+1} = \{ & (\gamma_1 \mid \mathtt{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid c_1, \gamma_2 \mid \mathtt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2)\ s.t. \\
& \exists.(\gamma_1' \mid \mathtt{Md}' \mid n_1' \mid \mathsf{NV}_1' \mid \mathsf{V}_1' \mid c_1')\ s.t. \\
& \gamma_1 \mid \mathtt{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid c_1 \to^*_{irred} \gamma_1' \mid \mathtt{Md}' \mid n_1' \mid \mathsf{NV}_1' \mid \mathsf{V}_1' \mid c_1' \wedge \\
& \exists.(\gamma_2' \mid \mathtt{Md}' \mid \infty \mid \mathsf{NV}_2' \mid \mathsf{V}_2' \mid c_2')\ s.t. \\
& \quad \gamma_2 \mid \mathtt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2 \to^* \gamma_2' \mid \mathtt{Md}' \mid \infty \mid \mathsf{NV}_2' \mid \mathsf{V}_2' \mid c_2' \wedge \\
& (\gamma_1' \mid \mathtt{Md}' \mid n_1' \mid \mathsf{NV}_1' \mid \mathsf{V}_1' \mid c_1', \gamma_2' \mid \mathtt{Md}' \mid \infty \mid \mathsf{NV}_2' \mid \mathsf{V}_2' \mid c_2') \in \mathcal{V}[\![\mathtt{C}_{\mathtt{unit}}]\!]^{m+1} \} \\
\mathcal{E}[\![\mathtt{C}_{\mathtt{unit}}]\!]^{0} = \{ & (\gamma_1 \mid \mathtt{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid c_1, \gamma_2 \mid \mathtt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2) \}
\end{aligned}$$

$$\texttt{Md} \mid b \geq 0 : \textbf{nat} \mid \Omega \mid \Sigma \Vdash c_1 \leq c_2 : \texttt{C}_{\texttt{unit}}$$

$$\text{iff } \forall n, m \geq 0.\ \forall \gamma, \mathsf{NV}, \mathsf{V}.s.t.\ \mathsf{NV} \mid \mathsf{V} \Vdash \gamma :: \Omega \mid \Sigma.$$

$$(\gamma \mid \texttt{Md} \mid n \mid \mathsf{NV} \mid \mathsf{V} \mid c_1, \gamma \mid \texttt{Md} \mid \infty \mid \mathsf{NV} \mid \mathsf{V} \mid c_2) \in \mathcal{E}[\![\texttt{C}_{\texttt{unit}}]\!]^m$$

Term Relation

$$\begin{aligned}
\mathcal{E}[\![\texttt{C}_{\texttt{unit}}]\!]^{m+1} = \{ & (\gamma_1 \mid \texttt{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid c_1, \gamma_2 \mid \texttt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2)\ s.t. \\
& \exists.(\gamma_1' \mid \texttt{Md}' \mid n_1' \mid \mathsf{NV}_1' \mid \mathsf{V}_1' \mid c_1')\ s.t. \\
& \gamma_1 \mid \texttt{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid c_1 \to^*_{irred} \gamma_1' \mid \texttt{Md}' \mid n_1' \mid \mathsf{NV}_1' \mid \mathsf{V}_1' \mid c_1' \wedge \\
& \exists.(\gamma_2' \mid \texttt{Md}' \mid \infty \mid \mathsf{NV}_2' \mid \mathsf{V}_2' \mid c_2')\ s.t. \\
& \quad \gamma_2 \mid \texttt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2 \to^* \gamma_2' \mid \texttt{Md}' \mid \infty \mid \mathsf{NV}_2' \mid \mathsf{V}_2' \mid c_2' \wedge \\
& (\gamma_1' \mid \texttt{Md}' \mid n_1' \mid \mathsf{NV}_1' \mid \mathsf{V}_1' \mid c_1', \gamma_2' \mid \texttt{Md}' \mid \infty \mid \mathsf{NV}_2' \mid \mathsf{V}_2' \mid c_2') \in \mathcal{V}[\![\texttt{C}_{\texttt{unit}}]\!]^{m+1} \} \\
\mathcal{E}[\![\texttt{C}_{\texttt{unit}}]\!]^{0} = \{ & (\gamma_1 \mid \texttt{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid c_1, \gamma_2 \mid \texttt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2)\}
\end{aligned}$$

$$\texttt{Md} \mid b \geq 0 : \textbf{nat} \mid \Omega \mid \Sigma \Vdash c_1 \leq c_2 : \texttt{C}_{\texttt{unit}}$$

$$\text{iff } \forall n, m \geq 0.\ \forall \gamma, \mathsf{NV}, \mathsf{V}. s.t.\ \mathsf{NV} \mid \mathsf{V} \Vdash \gamma :: \Omega \mid \Sigma.$$

$$(\gamma \mid \texttt{Md} \mid n \mid \mathsf{NV} \mid \mathsf{V} \mid c_1, \gamma \mid \texttt{Md} \mid \infty \mid \mathsf{NV} \mid \mathsf{V} \mid c_2) \in \mathcal{E}[\![\texttt{C}_{\texttt{unit}}]\!]^m$$

Term Relation

$$\begin{aligned}
\mathcal{E}[\![\texttt{C}_{\texttt{unit}}]\!]^{m+1} = \{ & (\gamma_1 \mid \texttt{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid c_1, \gamma_2 \mid \texttt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2)\ s.t. \\
& \exists.(\gamma_1' \mid \texttt{Md}' \mid n_1' \mid \mathsf{NV}_1' \mid \mathsf{V}_1' \mid c_1')\ s.t. \\
& \gamma_1 \mid \texttt{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid c_1 \to_{irred}^{*} \gamma_1' \mid \texttt{Md}' \mid n_1' \mid \mathsf{NV}_1' \mid \mathsf{V}_1' \mid c_1' \wedge \\
& \exists.(\gamma_2' \mid \texttt{Md}' \mid \infty \mid \mathsf{NV}_2' \mid \mathsf{V}_2' \mid c_2')\ s.t. \\
& \quad \gamma_2 \mid \texttt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2 \to^{*} \gamma_2' \mid \texttt{Md}' \mid \infty \mid \mathsf{NV}_2' \mid \mathsf{V}_2' \mid c_2' \wedge \\
& (\gamma_1' \mid \texttt{Md}' \mid n_1' \mid \mathsf{NV}_1' \mid \mathsf{V}_1' \mid c_1', \gamma_2' \mid \texttt{Md}' \mid \infty \mid \mathsf{NV}_2' \mid \mathsf{V}_2' \mid c_2') \in \mathcal{V}[\![\texttt{C}_{\texttt{unit}}]\!]^{m+1} \} \\
\mathcal{E}[\![\texttt{C}_{\texttt{unit}}]\!]^{0} = \{ & (\gamma_1 \mid \texttt{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid c_1, \gamma_2 \mid \texttt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2) \}
\end{aligned}$$

$$\mathtt{Md} \mid b \geq 0 : \mathbf{nat} \mid \Omega \mid \Sigma \Vdash c_1 \leq c_2 : \mathtt{C}_{\mathtt{unit}}$$
$$\text{iff } \forall n, m \geq 0.\ \forall \gamma, \mathsf{NV}, \mathsf{V}. s.t.\ \mathsf{NV} \mid \mathsf{V} \Vdash \gamma :: \Omega \mid \Sigma.$$
$$(\gamma \mid \mathtt{Md} \mid n \mid \mathsf{NV} \mid \mathsf{V} \mid c_1, \gamma \mid \mathtt{Md} \mid \infty \mid \mathsf{NV} \mid \mathsf{V} \mid c_2) \in \mathcal{E}[\![\mathtt{C}_{\mathtt{unit}}]\!]^m$$

Term Relation

$$\begin{aligned}
\mathcal{E}[\![\mathtt{C}_{\mathtt{unit}}]\!]^{m+1} = \{ & (\gamma_1 \mid \mathtt{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid c_1, \gamma_2 \mid \mathtt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2)\ s.t. \\
& \exists.(\gamma_1' \mid \mathtt{Md}' \mid n_1' \mid \mathsf{NV}_1' \mid \mathsf{V}_1' \mid c_1')\ s.t. \\
& \gamma_1 \mid \mathtt{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid c_1 \rightarrow^*_{irred} \gamma_1' \mid \mathtt{Md}' \mid n_1' \mid \mathsf{NV}_1' \mid \mathsf{V}_1' \mid c_1' \wedge \\
& \exists.(\gamma_2' \mid \mathtt{Md}' \mid \infty \mid \mathsf{NV}_2' \mid \mathsf{V}_2' \mid c_2')\ s.t. \\
& \quad \gamma_2 \mid \mathtt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2 \rightarrow^* \gamma_2' \mid \mathtt{Md}' \mid \infty \mid \mathsf{NV}_2' \mid \mathsf{V}_2' \mid c_2' \wedge \\
& (\gamma_1' \mid \mathtt{Md}' \mid n_1' \mid \mathsf{NV}_1' \mid \mathsf{V}_1' \mid c_1', \gamma_2' \mid \mathtt{Md}' \mid \infty \mid \mathsf{NV}_2' \mid \mathsf{V}_2' \mid c_2') \in \mathcal{V}[\![\mathtt{C}_{\mathtt{unit}}]\!]^{m+1} \} \\
\mathcal{E}[\![\mathtt{C}_{\mathtt{unit}}]\!]^{0} = \{ & (\gamma_1 \mid \mathtt{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid c_1, \gamma_2 \mid \mathtt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2) \}
\end{aligned}$$

$$\mathtt{Md} \mid b \geq 0 : \mathbf{nat} \mid \Omega \mid \Sigma \Vdash c_1 \leq c_2 : \mathtt{C}_{\mathtt{unit}}$$

$$\text{iff } \forall n, m \geq 0.\ \forall \gamma, \mathsf{NV}, \mathsf{V}. s.t.\ \mathsf{NV} \mid \mathsf{V} \Vdash \gamma :: \Omega \mid \Sigma.$$

$$(\gamma \mid \mathtt{Md} \mid n \mid \mathsf{NV} \mid \mathsf{V} \mid c_1, \gamma \mid \mathtt{Md} \mid \infty \mid \mathsf{NV} \mid \mathsf{V} \mid c_2) \in \mathcal{E}[\![\mathtt{C}_{\mathtt{unit}}]\!]^m$$

Term Relation

$$\begin{aligned}
\mathcal{E}[\![\mathtt{C}_{\mathtt{unit}}]\!]^{m+1} = \{ & (\gamma_1 \mid \mathtt{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid c_1, \gamma_2 \mid \mathtt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2)\ s.t. \\
& \exists.(\gamma_1' \mid \mathtt{Md}' \mid n_1' \mid \mathsf{NV}_1' \mid \mathsf{V}_1' \mid c_1')\ s.t. \\
& \gamma_1 \mid \mathtt{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid c_1 \to^*_{irred} \gamma_1' \mid \mathtt{Md}' \mid n_1' \mid \mathsf{NV}_1' \mid \mathsf{V}_1' \mid c_1' \land \\
& \exists.(\gamma_2' \mid \mathtt{Md}' \mid \infty \mid \mathsf{NV}_2' \mid \mathsf{V}_2' \mid c_2')\ s.t. \\
& \quad \gamma_2 \mid \mathtt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2 \to^* \gamma_2' \mid \mathtt{Md}' \mid \infty \mid \mathsf{NV}_2' \mid \mathsf{V}_2' \mid c_2' \land \\
& (\gamma_1' \mid \mathtt{Md}' \mid n_1' \mid \mathsf{NV}_1' \mid \mathsf{V}_1' \mid c_1', \gamma_2' \mid \mathtt{Md}' \mid \infty \mid \mathsf{NV}_2' \mid \mathsf{V}_2' \mid c_2') \in \mathcal{V}[\![\mathtt{C}_{\mathtt{unit}}]\!]^{m+1} \} \\
\mathcal{E}[\![\mathtt{C}_{\mathtt{unit}}]\!]^{0} = \{ & (\gamma_1 \mid \mathtt{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid c_1, \gamma_2 \mid \mathtt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2)\}
\end{aligned}$$

$$\texttt{Md} \mid b \geq 0 : \textbf{nat} \mid \Omega \mid \Sigma \Vdash c_1 \leq c_2 : \texttt{C}_{\texttt{unit}}$$

$$\text{iff } \forall n, m \geq 0.\ \forall \gamma, \mathsf{NV}, \mathsf{V}.s.t.\ \mathsf{NV} \mid \mathsf{V} \Vdash \gamma :: \Omega \mid \Sigma.$$

$$(\gamma \mid \texttt{Md} \mid n \mid \mathsf{NV} \mid \mathsf{V} \mid c_1, \gamma \mid \texttt{Md} \mid \infty \mid \mathsf{NV} \mid \mathsf{V} \mid c_2) \in \mathcal{E}[\![\texttt{C}_{\texttt{unit}}]\!]^m$$

Term Relation

$$\begin{aligned}
\mathcal{E}[\![\texttt{C}_{\texttt{unit}}]\!]^{m+1} = \{ & (\gamma_1 \mid \texttt{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid c_1, \gamma_2 \mid \texttt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2)\ s.t. \\
& \exists.(\gamma_1' \mid \texttt{Md}' \mid n_1' \mid \mathsf{NV}_1' \mid \mathsf{V}_1' \mid c_1')\ s.t. \\
& \gamma_1 \mid \texttt{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid c_1 \rightarrow^*_{irred} \gamma_1' \mid \texttt{Md}' \mid n_1' \mid \mathsf{NV}_1' \mid \mathsf{V}_1' \mid c_1' \wedge \\
& \exists.(\gamma_2' \mid \texttt{Md}' \mid \infty \mid \mathsf{NV}_2' \mid \mathsf{V}_2' \mid c_2')\ s.t. \\
& \quad \gamma_2 \mid \texttt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2 \rightarrow^* \gamma_2' \mid \texttt{Md}' \mid \infty \mid \mathsf{NV}_2' \mid \mathsf{V}_2' \mid c_2' \wedge \\
& (\gamma_1' \mid \texttt{Md}' \mid n_1' \mid \mathsf{NV}_1' \mid \mathsf{V}_1' \mid c_1', \gamma_2' \mid \texttt{Md}' \mid \infty \mid \mathsf{NV}_2' \mid \mathsf{V}_2' \mid c_2') \in \mathcal{V}[\![\texttt{C}_{\texttt{unit}}]\!]^{m+1} \} \\
\mathcal{E}[\![\texttt{C}_{\texttt{unit}}]\!]^{0} = \{ & (\gamma_1 \mid \texttt{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid c_1, \gamma_2 \mid \texttt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2) \}
\end{aligned}$$

Value Relation

$$\mathcal{V}[\![\uparrow\mathtt{unit}]\!]^m = \{(\gamma \mid \mathsf{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{skip}, \gamma \mid \mathsf{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{skip}) \text{ s.t. } \mathsf{NV}_1 = \mathsf{NV}_2\}$$

$$\mathcal{V}[\![\downarrow\uparrow\mathtt{unit}]\!]^m = \{(\gamma_1 \mid \mathsf{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid \mathsf{skip}, \gamma_2 \mid \mathsf{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid \mathsf{skip}) \text{ s.t. } \mathtt{Commit}(\gamma_i \mid \mathsf{Md} \mid \mathsf{NV}_i \mid \mathsf{V}_i) = \gamma_1' \mid \mathsf{NV}_i' \wedge (\gamma_1' \mid \mathsf{Md} \mid n_1 \mid \mathsf{NV}_1' \mid \mathsf{skip}, \gamma_2 \mid \mathsf{Md} \mid \infty \mid \mathsf{NV}_2' \mid \mathsf{skip}) \in \mathcal{V}[\![\uparrow\mathtt{unit}]\!]^m\}$$

$$\mathcal{V}[\![\uparrow\mathtt{C}_{\mathtt{unit}}]\!]^m = \{(\gamma_1 \mid \mathsf{Md} \mid n \mid \mathsf{NV}_1 \mid \uparrow\kappa, \gamma_2 \mid \mathsf{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2) \text{ s.t. } \mathtt{restore}(\gamma_1, \mathsf{Md}, \mathsf{NV}_1, \kappa) = \mathsf{NV}_0 \mid \mathsf{V}_0 \mid c_0 \wedge (\gamma_1 \mid \mathsf{Md} \mid n \mid \mathsf{NV}_0 \mid \mathsf{V}_0 \mid c_0, \gamma_2 \mid \mathsf{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2) \in \mathcal{E}[\![\mathtt{C}_{\mathtt{unit}}]\!]^m\}$$

$$\mathcal{V}[\![\mathtt{nat} \rightsquigarrow \uparrow\mathtt{C}_{\mathtt{unit}}]\!]^m = \{(\gamma_1 \mid \mathsf{Md} \mid \cdot \mid \mathsf{NV}_1 \mid \varepsilon \;\#\; \mathsf{in}(n > 0, \uparrow\kappa), \gamma_2 \mid \mathsf{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2) \text{ s.t. } \forall n > 0. (\gamma_1 \mid \mathsf{Md} \mid n \mid \mathsf{NV}_1 \mid \uparrow\kappa, \gamma_2 \mid \mathsf{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2) \in \mathcal{V}[\![\uparrow \mathtt{C}_{\mathtt{unit}}]\!]^m\}$$

$$\mathcal{V}[\![\downarrow(\mathtt{nat} \rightsquigarrow \uparrow\mathtt{C}_{\mathtt{unit}})]\!]^m = \{(\gamma_1 \mid \mathsf{Md} \mid \cdot \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid \downarrow\varepsilon \;\#\; \mathsf{in}(n > 0, \uparrow\kappa), \gamma_2 \mid \mathsf{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2) \text{ s.t. } \mathtt{PwOff}(\gamma_1, \mathsf{Md}, \mathsf{NV}_1, \mathsf{V}_1) = \gamma_1' \mid \mathsf{V}' \wedge (\gamma_1' \mid \mathsf{Md} \mid \cdot \mid \mathsf{V}', \mathsf{NV}_1 \mid \varepsilon \;\#\; \mathsf{in}(n > 0, \uparrow\kappa), \gamma_2 \mid \mathsf{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2) \in \mathcal{V}[\![\mathtt{nat} \rightsquigarrow \uparrow\mathtt{C}_{\mathtt{unit}}]\!]^m\}$$

$$\mathcal{V}[\![\mathtt{C}_{\mathtt{unit}}]\!]^{m+1} = \{(\gamma_1 \mid \mathsf{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid c_1,\ \gamma_2 \mid \mathsf{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2) \text{ s.t. either } n_1 = 0 \wedge (\gamma_1 \mid \mathsf{Md} \mid \cdot \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid \downarrow\varepsilon \;\#\; \mathsf{in}(n_1 > 0, \uparrow c_1), \gamma_2 \mid \mathsf{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2) \in \mathcal{V}[\![\downarrow(\mathtt{nat} \rightsquigarrow\uparrow \mathtt{C}_{\mathtt{unit}})]\!]^m, \text{ or } n_1 > 0 \wedge (\gamma_1 \mid \mathsf{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid c_1, \gamma_2 \mid \mathsf{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2) \in \mathcal{V}[\![\downarrow\uparrow \mathtt{unit}]\!]^m\}$$

Value Relation

$$\mathcal{V}[\![\uparrow\texttt{unit}]\!]^m = \{(\gamma \,|\, \mathsf{Md} \,|\, n_1 \,|\, \mathsf{NV}_1 \,|\, \mathsf{skip}, \gamma \,|\, \mathsf{Md} \,|\, \infty \,|\, \mathsf{NV}_2 \,|\, \mathsf{skip}) \text{ s.t. } \mathsf{NV}_1 = \mathsf{NV}_2\}$$

$$\begin{aligned}\mathcal{V}[\![\downarrow\uparrow\texttt{unit}]\!]^m = \{&(\gamma_1 \,|\, \mathsf{Md} \,|\, n_1 \,|\, \mathsf{NV}_1 \,|\, \mathsf{V}_1 \,|\, \mathsf{skip}, \gamma_2 \,|\, \mathsf{Md} \,|\, \infty \,|\, \mathsf{NV}_2 \,|\, \mathsf{V}_2 \,|\, \mathsf{skip}) \text{ s.t.}\\ &\texttt{Commit}(\gamma_i \,|\, \mathsf{Md} \,|\, \mathsf{NV}_i \,|\, \mathsf{V}_i) = \gamma_1' \,|\, \mathsf{NV}_i' \wedge\\ &(\gamma_1' \,|\, \mathsf{Md} \,|\, n_1 \,|\, \mathsf{NV}_1' \,|\, \mathsf{skip}, \gamma_2 \,|\, \mathsf{Md} \,|\, \infty \,|\, \mathsf{NV}_2' \,|\, \mathsf{skip}) \in \mathcal{V}[\![\uparrow\texttt{unit}]\!]^m\}\end{aligned}$$

$$\begin{aligned}\mathcal{V}[\![\uparrow\mathsf{C}_{\texttt{unit}}]\!]^m = \{&(\gamma_1 \,|\, \mathsf{Md} \,|\, n \,|\, \mathsf{NV}_1 \,|\, \uparrow\kappa, \gamma_2 \,|\, \mathsf{Md} \,|\, \infty \,|\, \mathsf{NV}_2 \,|\, \mathsf{V}_2 \,|\, c_2) \text{ s.t.}\\ &\texttt{restore}(\gamma_1, \mathsf{Md}, \mathsf{NV}_1, \kappa) = \mathsf{NV}_0 \,|\, \mathsf{V}_0 \,|\, c_0 \wedge\\ &(\gamma_1 \,|\, \mathsf{Md} \,|\, n \,|\, \mathsf{NV}_0 \,|\, \mathsf{V}_0 \,|\, c_0, \gamma_2 \,|\, \mathsf{Md} \,|\, \infty \,|\, \mathsf{NV}_2 \,|\, \mathsf{V}_2 \,|\, c_2) \in \mathcal{E}[\![\mathsf{C}_{\texttt{unit}}]\!]^m\}\end{aligned}$$

$$\begin{aligned}\mathcal{V}[\![\texttt{nat}\rightsquigarrow\uparrow\mathsf{C}_{\texttt{unit}}]\!]^m = \{&(\gamma_1 \,|\, \mathsf{Md} \,|\, \cdot \,|\, \mathsf{NV}_1 \,|\, \varepsilon \;\#\; \mathsf{in}(n > 0, \uparrow\kappa), \gamma_2 \,|\, \mathsf{Md} \,|\, \infty \,|\, \mathsf{NV}_2 | \mathsf{V}_2 | c_2) \text{ s.t.}\\ &\forall n{>}0. (\gamma_1 \,|\, \mathsf{Md} \,|\, n \,|\mathsf{NV}_1|\, \uparrow\kappa, \gamma_2 \,|\, \mathsf{Md} \,|\, \infty \,|\mathsf{NV}_2|\mathsf{V}_2|\, c_2) \in \mathcal{V}[\![\uparrow \mathsf{C}_{\texttt{unit}}]\!]^m\}\end{aligned}$$

$$\begin{aligned}\mathcal{V}[\![\downarrow(\texttt{nat}\rightsquigarrow\uparrow\mathsf{C}_{\texttt{unit}})]\!]^m = \{&(\gamma_1 \,|\, \mathsf{Md} \,|\, \cdot \,|\mathsf{NV}_1|\mathsf{V}_1|\downarrow\varepsilon \;\#\; \mathsf{in}(n > 0, \uparrow\kappa), \gamma_2 \,|\, \mathsf{Md} \,|\, \infty \,|\mathsf{NV}_2|\mathsf{V}_2|\, c_2)\\ &\text{s.t. } \texttt{PwOff}(\gamma_1, \mathsf{Md}, \mathsf{NV}_1, \mathsf{V}_1) = \gamma_1' \,|\, \mathsf{V}' \wedge\\ &(\gamma_1' \,|\, \mathsf{Md} \,|\, \cdot \,|\, \mathsf{V}', \mathsf{NV}_1 \,|\, \varepsilon \;\#\; \mathsf{in}(n > 0, \uparrow\kappa), \gamma_2 \,|\, \mathsf{Md} \,|\, \infty \,|\, \mathsf{NV}_2 \,|\, \mathsf{V}_2 \,|\, c_2)\\ &\qquad \in \mathcal{V}[\![\texttt{nat} \rightsquigarrow \uparrow\mathsf{C}_{\texttt{unit}}]\!]^m\}\end{aligned}$$

$$\begin{aligned}\mathcal{V}[\![\mathsf{C}_{\texttt{unit}}]\!]^{m+1} = \{&(\gamma_1 \,|\, \mathsf{Md} \,|\, n_1 \,|\, \mathsf{NV}_1 \,|\, \mathsf{V}_1 \,|\, c_1,\ \gamma_2 \,|\, \mathsf{Md} \,|\, \infty \,|\, \mathsf{NV}_2 \,|\, \mathsf{V}_2 \,|\, c_2)\\ &\text{s.t. either}\\ &n_1 = 0 \wedge (\gamma_1 \,|\, \mathsf{Md} \,|\, \cdot \,|\, \mathsf{NV}_1 \,|\, \mathsf{V}_1 \,|\downarrow\varepsilon \;\#\; \mathsf{in}(n_1 > 0, \uparrow c_1),\\ &\qquad \gamma_2 \,|\, \mathsf{Md} \,|\, \infty \,|\, \mathsf{NV}_2 \,|\, \mathsf{V}_2 \,|\, c_2) \in \mathcal{V}[\![\downarrow(\texttt{nat} \rightsquigarrow\uparrow \mathsf{C}_{\texttt{unit}})]\!]^m, \text{ or}\\ &n_1 > 0 \wedge (\gamma_1 \,|\, \mathsf{Md} \,|\, n_1 \,|\, \mathsf{NV}_1 \,|\, \mathsf{V}_1 \,|\, c_1, \gamma_2 \,|\, \mathsf{Md} \,|\, \infty \,|\, \mathsf{NV}_2 \,|\, \mathsf{V}_2 \,|\, c_2)\\ &\qquad \in \mathcal{V}[\![\downarrow\uparrow \texttt{unit}]\!]^m\}\end{aligned}$$

Value Relation

$$\mathcal{V}[\![\uparrow\texttt{unit}]\!]^m = \{(\gamma \,|\, \mathsf{Md} \,|\, n_1 \,|\, \mathsf{NV}_1 \,|\, \mathsf{skip}, \gamma \,|\, \mathsf{Md} \,|\, \infty \,|\, \mathsf{NV}_2 \,|\, \mathsf{skip}) \text{ s.t. } \mathsf{NV}_1 = \mathsf{NV}_2\}$$

$$\begin{aligned}\mathcal{V}[\![\downarrow\uparrow\texttt{unit}]\!]^m = \{&(\gamma_1 \,|\, \mathsf{Md} \,|\, n_1 \,|\, \mathsf{NV}_1 \,|\, \mathsf{V}_1 \,|\, \mathsf{skip}, \gamma_2 \,|\, \mathsf{Md} \,|\, \infty \,|\, \mathsf{NV}_2 \,|\, \mathsf{V}_2 \,|\, \mathsf{skip}) \text{ s.t.}\\ &\texttt{Commit}(\gamma_i \,|\, \mathsf{Md} \,|\, \mathsf{NV}_i \,|\, \mathsf{V}_i) = \gamma_1' \,|\, \mathsf{NV}_i' \wedge\\ &(\gamma_1' \,|\, \mathsf{Md} \,|\, n_1 \,|\, \mathsf{NV}_1' \,|\, \mathsf{skip}, \gamma_2 \,|\, \mathsf{Md} \,|\, \infty \,|\, \mathsf{NV}_2' \,|\, \mathsf{skip}) \in \mathcal{V}[\![\uparrow\texttt{unit}]\!]^m\}\end{aligned}$$

$$\begin{aligned}\mathcal{V}[\![\uparrow\mathsf{C}_{\texttt{unit}}]\!]^m = \{&(\gamma_1 \,|\, \mathsf{Md} \,|\, n \,|\, \mathsf{NV}_1 \,|\, \uparrow\kappa, \gamma_2 \,|\, \mathsf{Md} \,|\, \infty \,|\, \mathsf{NV}_2 \,|\, \mathsf{V}_2 \,|\, c_2) \text{ s.t.}\\ &\texttt{restore}(\gamma_1, \mathsf{Md}, \mathsf{NV}_1, \kappa) = \mathsf{NV}_0 \,|\, \mathsf{V}_0 \,|\, c_0 \wedge\\ &(\gamma_1 \,|\, \mathsf{Md} \,|\, n \,|\, \mathsf{NV}_0 \,|\, \mathsf{V}_0 \,|\, c_0, \gamma_2 \,|\, \mathsf{Md} \,|\, \infty \,|\, \mathsf{NV}_2 \,|\, \mathsf{V}_2 \,|\, c_2) \in \mathcal{E}[\![\mathsf{C}_{\texttt{unit}}]\!]^m\}\end{aligned}$$

$$\begin{aligned}\mathcal{V}[\![\texttt{nat} \rightsquigarrow \uparrow\mathsf{C}_{\texttt{unit}}]\!]^m = \{&(\gamma_1 \,|\, \mathsf{Md} \,|\, \cdot \,|\, \mathsf{NV}_1 \,|\, \varepsilon \,\#\, \mathsf{in}(n > 0, \uparrow\kappa), \gamma_2 \,|\, \mathsf{Md} \,|\, \infty \,|\, \mathsf{NV}_2 | \mathsf{V}_2 | c_2) \text{ s.t.}\\ &\forall n{>}0. (\gamma_1 \,|\, \mathsf{Md} \,|\, n \,|\mathsf{NV}_1 |\, \uparrow\kappa, \gamma_2 \,|\, \mathsf{Md} \,|\, \infty \,|\mathsf{NV}_2 | \mathsf{V}_2 |\, c_2) \in \mathcal{V}[\![\uparrow \mathsf{C}_{\texttt{unit}}]\!]^m\}\end{aligned}$$

$$\begin{aligned}\mathcal{V}[\![\downarrow(\texttt{nat} \rightsquigarrow \uparrow\mathsf{C}_{\texttt{unit}})]\!]^m = \{&(\gamma_1 \,|\, \mathsf{Md} \,|\, \cdot \,| \mathsf{NV}_1 | \mathsf{V}_1 | \downarrow\varepsilon \,\#\, \mathsf{in}(n > 0, \uparrow\kappa), \gamma_2 \,|\, \mathsf{Md} \,|\, \infty \,| \mathsf{NV}_2 | \mathsf{V}_2 |\, c_2)\\ &\text{s.t. } \texttt{PwOff}(\gamma_1, \mathsf{Md}, \mathsf{NV}_1, \mathsf{V}_1) = \gamma_1' \,|\, \mathsf{V}' \wedge\\ &(\gamma_1' \,|\, \mathsf{Md} \,|\, \cdot \,|\, \mathsf{V}', \mathsf{NV}_1 \,|\, \varepsilon \,\#\, \mathsf{in}(n > 0, \uparrow\kappa), \gamma_2 \,|\, \mathsf{Md} \,|\, \infty \,|\, \mathsf{NV}_2 \,|\, \mathsf{V}_2 \,|\, c_2)\\ &\qquad \in \mathcal{V}[\![\texttt{nat} \rightsquigarrow \uparrow\mathsf{C}_{\texttt{unit}}]\!]^m\}\end{aligned}$$

$$\begin{aligned}\mathcal{V}[\![\mathsf{C}_{\texttt{unit}}]\!]^{m+1} = \{&(\gamma_1 \,|\, \mathsf{Md} \,|\, n_1 \,|\, \mathsf{NV}_1 \,|\, \mathsf{V}_1 \,|\, c_1,\ \gamma_2 \,|\, \mathsf{Md} \,|\, \infty \,|\, \mathsf{NV}_2 \,|\, \mathsf{V}_2 \,|\, c_2)\\ &\text{s.t. either}\\ &n_1 = 0 \wedge (\gamma_1 \,|\, \mathsf{Md} \,|\, \cdot \,|\, \mathsf{NV}_1 \,|\, \mathsf{V}_1 \,|\, \downarrow\varepsilon \,\#\, \mathsf{in}(n_1 > 0, \uparrow c_1),\\ &\qquad \gamma_2 \,|\, \mathsf{Md} \,|\, \infty \,|\, \mathsf{NV}_2 \,|\, \mathsf{V}_2 \,|\, c_2) \in \mathcal{V}[\![\downarrow(\texttt{nat} \rightsquigarrow \uparrow \mathsf{C}_{\texttt{unit}})]\!]^m, \text{ or}\\ &n_1 > 0 \wedge (\gamma_1 \,|\, \mathsf{Md} \,|\, n_1 \,|\, \mathsf{NV}_1 \,|\, \mathsf{V}_1 \,|\, c_1, \gamma_2 \,|\, \mathsf{Md} \,|\, \infty \,|\, \mathsf{NV}_2 \,|\, \mathsf{V}_2 \,|\, c_2)\\ &\qquad \in \mathcal{V}[\![\downarrow\uparrow \texttt{unit}]\!]^m\}\end{aligned}$$

$$\mathtt{Md} \mid b \geq 0 : \mathbf{nat} \mid \Omega \mid \Sigma \Vdash c_1 \leq c_2 : \mathtt{C_{unit}}$$
iff $\forall n, m \geq 0.\ \forall \gamma, \mathsf{NV}, \mathsf{V}. s.t.\ \mathsf{NV} \mid \mathsf{V} \Vdash \gamma :: \Omega \mid \Sigma.$
$$(\gamma \mid \mathtt{Md} \mid n \mid \mathsf{NV} \mid \mathsf{V} \mid c_1, \gamma \mid \mathtt{Md} \mid \infty \mid \mathsf{NV} \mid \mathsf{V} \mid c_2) \in \mathcal{E}[\![\mathtt{C_{unit}}]\!]^m$$

Term Relation

$$\begin{aligned}
\mathcal{E}[\![\mathtt{C_{unit}}]\!]^{m+1} = \{ & (\gamma_1 \mid \mathtt{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid c_1, \gamma_2 \mid \mathtt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2)\ s.t. \\
& \exists.(\gamma_1' \mid \mathtt{Md}' \mid n_1' \mid \mathsf{NV}_1' \mid \mathsf{V}_1' \mid c_1')\ s.t. \\
& \gamma_1 \mid \mathtt{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid c_1 \rightarrow^*_{irred} \gamma_1' \mid \mathtt{Md}' \mid n_1' \mid \mathsf{NV}_1' \mid \mathsf{V}_1' \mid c_1' \wedge \\
& \exists.(\gamma_2' \mid \mathtt{Md}' \mid \infty \mid \mathsf{NV}_2' \mid \mathsf{V}_2' \mid c_2')\ s.t. \\
& \quad \gamma_2 \mid \mathtt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2 \rightarrow^* \gamma_2' \mid \mathtt{Md}' \mid \infty \mid \mathsf{NV}_2' \mid \mathsf{V}_2' \mid c_2' \wedge \\
& (\gamma_1' \mid \mathtt{Md}' \mid n_1' \mid \mathsf{NV}_1' \mid \mathsf{V}_1' \mid c_1', \gamma_2' \mid \mathtt{Md}' \mid \infty \mid \mathsf{NV}_2' \mid \mathsf{V}_2' \mid c_2') \in \mathcal{V}[\![\mathtt{C_{unit}}]\!]^{m+1} \} \\
\mathcal{E}[\![\mathtt{C_{unit}}]\!]^{0} = \{ & (\gamma_1 \mid \mathtt{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid c_1, \gamma_2 \mid \mathtt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2) \}
\end{aligned}$$

Value Relation

$$\begin{aligned}
\mathcal{V}[\![\uparrow\mathtt{unit}]\!]^m = \{ & (\gamma \mid \mathtt{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{skip}, \gamma \mid \mathtt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{skip})\ \mathtt{s.t.} \mathsf{NV}_1 = \mathsf{NV}_2 \} \\
\mathcal{V}[\![\downarrow\uparrow\mathtt{unit}]\!]^m = \{ & (\gamma_1 \mid \mathtt{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid \mathsf{skip}, \gamma_2 \mid \mathtt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid \mathsf{skip})\ \mathtt{s.t.} \\
& \mathtt{Commit}(\gamma_i \mid \mathtt{Md} \mid \mathsf{NV}_i \mid \mathsf{V}_i) = \gamma_1' \mid \mathsf{NV}_i' \wedge \\
& (\gamma_1' \mid \mathtt{Md} \mid n_1 \mid \mathsf{NV}_1' \mid \mathsf{skip}, \gamma_2 \mid \mathtt{Md} \mid \infty \mid \mathsf{NV}_2' \mid \mathsf{skip}) \in \mathcal{V}[\![\uparrow\mathtt{unit}]\!]^m \} \\
\mathcal{V}[\![\uparrow\mathtt{C_{unit}}]\!]^m = \{ & (\gamma_1 \mid \mathtt{Md} \mid n \mid \mathsf{NV}_1 \mid \uparrow\kappa, \gamma_2 \mid \mathtt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2)\ \mathtt{s.t.} \\
& \mathtt{restore}(\gamma_1, \mathtt{Md}, \mathsf{NV}_1, \kappa) = \mathsf{NV}_0 \mid \mathsf{V}_0 \mid c_0 \wedge \\
& (\gamma_1 \mid \mathtt{Md} \mid n \mid \mathsf{NV}_0 \mid \mathsf{V}_0 \mid c_0, \gamma_2 \mid \mathtt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2) \in \mathcal{E}[\![\mathtt{C_{unit}}]\!]^m \} \\
\mathcal{V}[\![\mathtt{nat} \rightsquigarrow \uparrow\mathtt{C_{unit}}]\!]^m = \{ & (\gamma_1 \mid \mathtt{Md} \mid \cdot \mid \mathsf{NV}_1 \mid \varepsilon\ \#\ \mathsf{in}(n > 0, \uparrow\kappa), \gamma_2 \mid \mathtt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2)\ \mathtt{s.t.} \\
& \forall n{>}0.(\gamma_1 \mid \mathtt{Md} \mid n \mid \mathsf{NV}_1 \mid \uparrow\kappa, \gamma_2 \mid \mathtt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2) \in \mathcal{V}[\![\uparrow \mathtt{C_{unit}}]\!]^m \} \\
\mathcal{V}[\![\downarrow(\mathtt{nat} \rightsquigarrow \uparrow\mathtt{C_{unit}})]\!]^m = \{ & (\gamma_1 \mid \mathtt{Md} \mid \cdot \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid \downarrow\varepsilon\ \#\ \mathsf{in}(n > 0, \uparrow\kappa), \gamma_2 \mid \mathtt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2) \\
& \mathtt{s.t.}\ \mathtt{PwOff}(\gamma_1, \mathtt{Md}, \mathsf{NV}_1, \mathsf{V}_1) = \gamma_1' \mid \mathsf{V}' \wedge \\
& (\gamma_1' \mid \mathtt{Md} \mid \cdot \mid \mathsf{V}', \mathsf{NV}_1 \mid \varepsilon\ \#\ \mathsf{in}(n > 0, \uparrow\kappa), \gamma_2 \mid \mathtt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2) \\
& \qquad \in \mathcal{V}[\![\mathtt{nat} \rightsquigarrow \uparrow\mathtt{C_{unit}}]\!]^m \} \\
\mathcal{V}[\![\mathtt{C_{unit}}]\!]^{m+1} = \{ & (\gamma_1 \mid \mathtt{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid c_1,\ \gamma_2 \mid \mathtt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2) \\
& \mathtt{s.t.\ either} \\
& n_1 = 0 \wedge (\gamma_1 \mid \mathtt{Md} \mid \cdot \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid \downarrow\varepsilon\ \#\ \mathsf{in}(n_1 > 0, \uparrow c_1), \\
& \qquad \gamma_2 \mid \mathtt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2) \in \mathcal{V}[\![\downarrow(\mathtt{nat} \rightsquigarrow\uparrow \mathtt{C_{unit}})]\!]^m, \mathtt{or} \\
& n_1 > 0 \wedge (\gamma_1 \mid \mathtt{Md} \mid n_1 \mid \mathsf{NV}_1 \mid \mathsf{V}_1 \mid c_1, \gamma_2 \mid \mathtt{Md} \mid \infty \mid \mathsf{NV}_2 \mid \mathsf{V}_2 \mid c_2) \\
& \qquad \in \mathcal{V}[\![\downarrow\uparrow\ \mathtt{unit}]\!]^m \}
\end{aligned}$$

Semantic typing

$$\frac{\mathsf{jit} \mid b \geq 0 : \mathtt{nat} \mid \Omega; \cdot \Vdash c \leq c : \mathtt{C}_{\mathtt{unit}} \quad b : \mathtt{nat} \mid \Omega \Vdash p : \uparrow\mathtt{C}_{\mathtt{unit}}}{b : \mathtt{nat} \mid \Omega \Vdash c; p : \uparrow\mathtt{C}_{\mathtt{unit}}} \text{ (P-SEQ-SEMANTIC)}$$

$$\frac{\begin{array}{c}\Omega_0 \mid \Sigma_0 = \mathsf{InitWorld}_t(\Omega; \rho) \\ \mathsf{aID}(c_0) \mid b \geq 0 : \mathtt{nat} \mid \Omega_0; \Sigma_0 \Vdash c_0 \leq c_0 : \mathtt{C}_{\mathtt{unit}} \quad b : \mathtt{nat} \mid \Omega \Vdash p : \uparrow\mathtt{C}_{\mathtt{unit}}\end{array}}{b : \mathtt{nat} \mid \Omega \Vdash \mathsf{Ckpt}[\mathsf{aID}, \rho](c_0); p : \uparrow\mathtt{C}_{\mathtt{unit}}} \text{ (P-CKPT-SEMANTIC)}$$