

CS 511: Formal Methods in Security and Privacy

Non-interference

Marco Gaboardi
gaboardi@bu.edu

Alley Stoughton
stough@bu.edu

Rules of Hoare Logic:

$$\frac{}{\vdash \text{skip} : P \Rightarrow P}$$

$$\frac{}{\vdash \text{abort} : P \Rightarrow Q}$$

$$\frac{}{\vdash x := e : P[e/x] \Rightarrow P}$$

$$\frac{\vdash c : P \Rightarrow R \quad \vdash c' : R \Rightarrow Q}{\vdash c ; c' : P \Rightarrow Q}$$

$$\frac{P \Rightarrow S \quad \vdash c : S \Rightarrow R \quad R \Rightarrow Q}{\vdash c : P \Rightarrow Q}$$

$$\frac{\vdash c_1 : e \wedge P \Rightarrow Q \quad \vdash c_2 : \neg e \wedge P \Rightarrow Q}{\vdash \text{if } e \text{ then } c_1 \text{ else } c_2 : P \Rightarrow Q}$$

$$\frac{\vdash c : e \wedge P \Rightarrow P}{\vdash \text{while } e \text{ do } c : P \Rightarrow P \wedge \neg e}$$

Security as information flow
control

Some Examples of Security Properties

- Access Control
- Encryption
- Malicious Behavior Detection
- Information Filtering
- Information Flow Control

Some Examples of Security Properties

- Access Control
- Encryption
- Malicious Behavior Detection
- Information Filtering
- Information Flow Control

Private vs Public

We want to distinguish **confidential information** that need to be kept secret from **nonconfidential information** that can be accessed by everyone.

We assume that every variable is tagged with one either **public** or **private**.

`x:public`

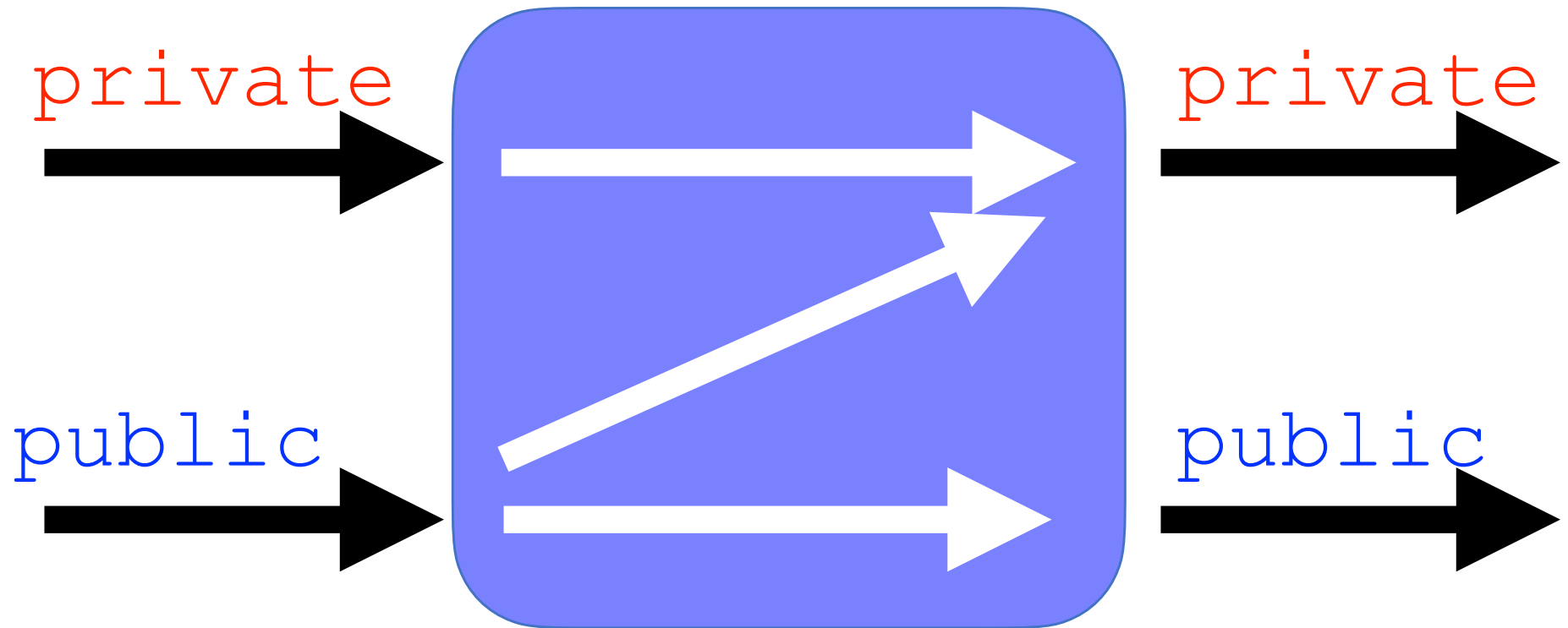
`x:private`

Information Flow Control

We want to guarantee that **confidential**
information do not flow in what is considered
nonconfidential.

Information Flow Control

We want to guarantee that **confidential information** do not flow in what is considered **nonconfidential**.



Is this program secure?

```
x:private  
y:public  
  
x:=y
```

Is this program secure?

```
x:private  
y:public  
  
x:=y
```

Secure

Is this program secure?

```
x:private  
y:public  
  
y:=x
```

Is this program secure?

```
x:private  
y:public  
  
y:=x
```

Insecure

Is this program secure?

```
x:private  
y:public  
  
y:=x;  
y:=5
```

Is this program secure?

```
x:private  
y:public  
  
y:=x;  
y:=5
```

Secure

Is this program secure?

```
x:private  
y:public
```

```
if y mod 3 = 0 then  
  x:=1  
else  
  x:=0
```

Is this program secure?

```
x:private  
y:public
```

```
if y mod 3 = 0 then  
  x:=1  
else  
  x:=0
```

Secure

Is this program secure?

```
x:private  
y:public
```

```
if x mod 3 = 0 then  
  y:=1  
else  
  y:=0
```

Is this program secure?

```
x:private  
y:public
```

```
if x mod 3 = 0 then  
  y:=1  
else  
  y:=0
```

Insecure

How can we formulate a policy that forbids flows from private to public?

Low equivalence

Two memories m_1 and m_2 are low equivalent if and only if they coincide in the value that they assign to public variables.

In symbols: $m_1 \sim_{\text{low}} m_2$

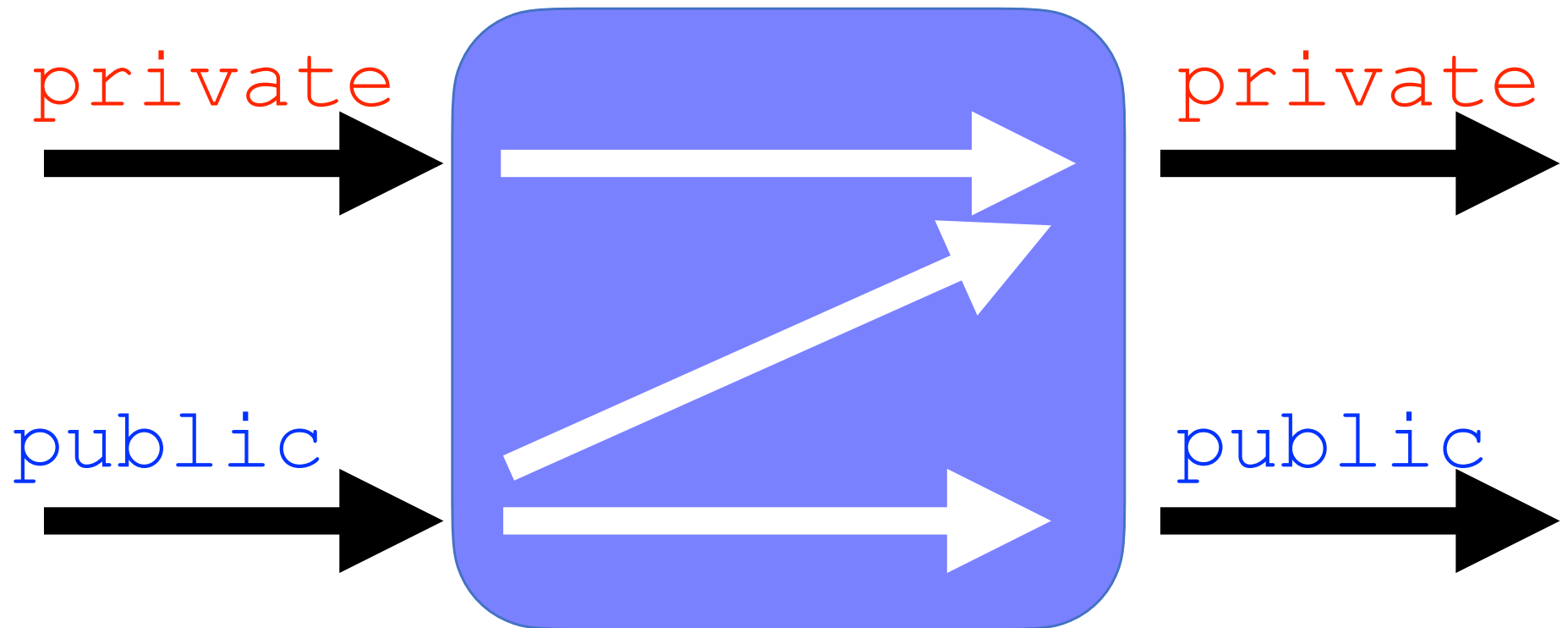
Noninterference

A program `prog` is **noninterferent** if and only if, whenever we run it on two memories m_1 and m_2 that are **low equivalent**, we obtain two memories m_1' and m_2' which are also **low equivalent**.

Noninterference

In symbols

$m_1 \sim_{\text{low}} m_2$ and $\{c\}_{m_1} = m_1'$ and $m_2' \{c\}_{m_2} = m_2'$
implies $m_1' \sim_{\text{low}} m_2'$



Does this program satisfy noninterference?

```
x:private  
y:public  
  
x:=y
```

Does this program satisfy noninterference?

```
x:private  
y:public  
  
x := y
```

Yes

Does this program satisfy noninterference?

```
x:private  
y:public  
  
x := y
```

Yes

$m_1^{\text{in}} = [x=n_1, y=k]$

Does this program satisfy noninterference?

```
x:private  
y:public  
  
x := y
```

Yes

$m_1^{in} = [x=n_1, y=k]$

$m_2^{in} = [x=n_2, y=k]$

Does this program satisfy noninterference?

```
x:private  
y:public  
  
x := y
```

Yes

$m_1^{\text{in}} = [x=n_1, y=k]$

$m_1^{\text{out}} = [x=k, y=k]$

$m_2^{\text{in}} = [x=n_2, y=k]$

$m_2^{\text{out}} = [x=k, y=k]$

Does this program satisfy noninterference?

```
x:private  
y:public  
  
y:=x
```

Does this program satisfy noninterference?

```
x:private  
y:public  
  
y:=x
```

No

Does this program satisfy noninterference?

```
x:private  
y:public  
  
y:=x
```

No

$m_1^{\text{in}} = [x=n_1, y=k]$

Does this program satisfy noninterference?

```
x:private  
y:public  
  
y:=x
```

No

$m_1^{\text{in}} = [x=n_1, y=k]$

$m_2^{\text{in}} = [x=n_2, y=k]$

Does this program satisfy noninterference?

```
x:private  
y:public  
  
y := x
```

No

$m_1^{\text{in}} = [x=n_1, y=k]$

$m_1^{\text{out}} = [x=n_1, y=n_1]$

$m_2^{\text{in}} = [x=n_2, y=k]$

$m_2^{\text{out}} = [x=n_2, y=n_2]$

Does this program satisfy noninterference?

```
x:private  
y:public  
  
y:=x  
y:=5
```

Does this program satisfy noninterference?

```
x:private  
y:public
```

```
y:=x  
y:=5
```

Yes

Does this program satisfy noninterference?

```
x:private  
y:public
```

```
y:=x  
y:=5
```

Yes

$m_1^{\text{in}} = [x=n_1, y=k]$

Does this program satisfy noninterference?

```
x:private  
y:public
```

```
y:=x  
y:=5
```

Yes

$m_1^{\text{in}} = [x=n_1, y=k]$

$m_2^{\text{in}} = [x=n_2, y=k]$

Does this program satisfy noninterference?

```
x:private  
y:public
```

```
y:=x  
y:=5
```

Yes

$m_1^{\text{in}} = [x=n_1, y=k]$

$m_1^{\text{out}} = [x=n_1, y=5]$

$m_2^{\text{in}} = [x=n_2, y=k]$

$m_2^{\text{out}} = [x=n_2, y=5]$

Does this program satisfy noninterference?

```
x:private  
y:public  
if y mod 3 = 0 then  
  x:=1  
else  
  x:=0
```

Does this program satisfy noninterference?

```
x:private  
y:public  
if y mod 3 = 0 then  
  x:=1  
else  
  x:=0
```

Yes

Does this program satisfy noninterference?

```
x:private  
y:public  
if y mod 3 = 0 then  
  x:=1  
else  
  x:=0
```

Yes

$m_1^{in} = [x=n_1, y=6]$

Does this program satisfy noninterference?

```
x:private  
y:public  
if y mod 3 = 0 then  
  x:=1  
else  
  x:=0
```

Yes

$m^{in}_1 = [x=n_1, y=6]$

$m^{in}_2 = [x=n_2, y=6]$

Does this program satisfy noninterference?

```
x:private  
y:public  
if y mod 3 = 0 then  
  x:=1  
else  
  x:=0
```

Yes

$m^{\text{in}}_1 = [x=n_1, y=6]$

$m^{\text{in}}_2 = [x=n_2, y=6]$

$m^{\text{out}}_1 = [x=1, y=6]$

$m^{\text{out}}_2 = [x=1, y=6]$

Does this program satisfy noninterference?

```
x:private  
y:public  
if x mod 3 = 0 then  
  y:=1  
else  
  y:=0
```

Does this program satisfy noninterference?

```
x:private  
y:public  
if x mod 3 = 0 then  
  y:=1  
else  
  y:=0
```

No

Does this program satisfy noninterference?

```
x:private  
y:public  
if x mod 3 = 0 then  
  y:=1  
else  
  y:=0
```

No

$m_1^{in} = [x=6, y=k]$

Does this program satisfy noninterference?

```
x:private  
y:public  
if x mod 3 = 0 then  
  y:=1  
else  
  y:=0
```

No

$m_1^{in} = [x=6, y=k]$

$m_2^{in} = [x=5, y=k]$

Does this program satisfy noninterference?

```
x:private  
y:public  
if x mod 3 = 0 then  
  y:=1  
else  
  y:=0
```

No

$m_1^{\text{in}} = [x=6, y=k]$

$m_2^{\text{in}} = [x=5, y=k]$

$m_1^{\text{out}} = [x=6, y=1]$

$m_2^{\text{out}} = [x=5, y=0]$

Does this program satisfy noninterference?

```
s1:public
s2:private
r:private
i:public

proc Compare (s1:list[n] bool,s2:list[n] bool)
i:=0;
r:=0;
while i<n /\ r=0 do
  if not(s1[i]=s2[i]) then
    r:=1
  i:=i+1
```

Does this program satisfy noninterference?

```
s1:public
s2:private
r:private
i:public

proc Compare (s1:list[n] bool,s2:list[n] bool)
i:=0;
r:=0;
while i<n /\ r=0 do
  if not(s1[i]=s2[i]) then
    r:=1
  i:=i+1
```

No

How can we prove our
programs noninterferent?

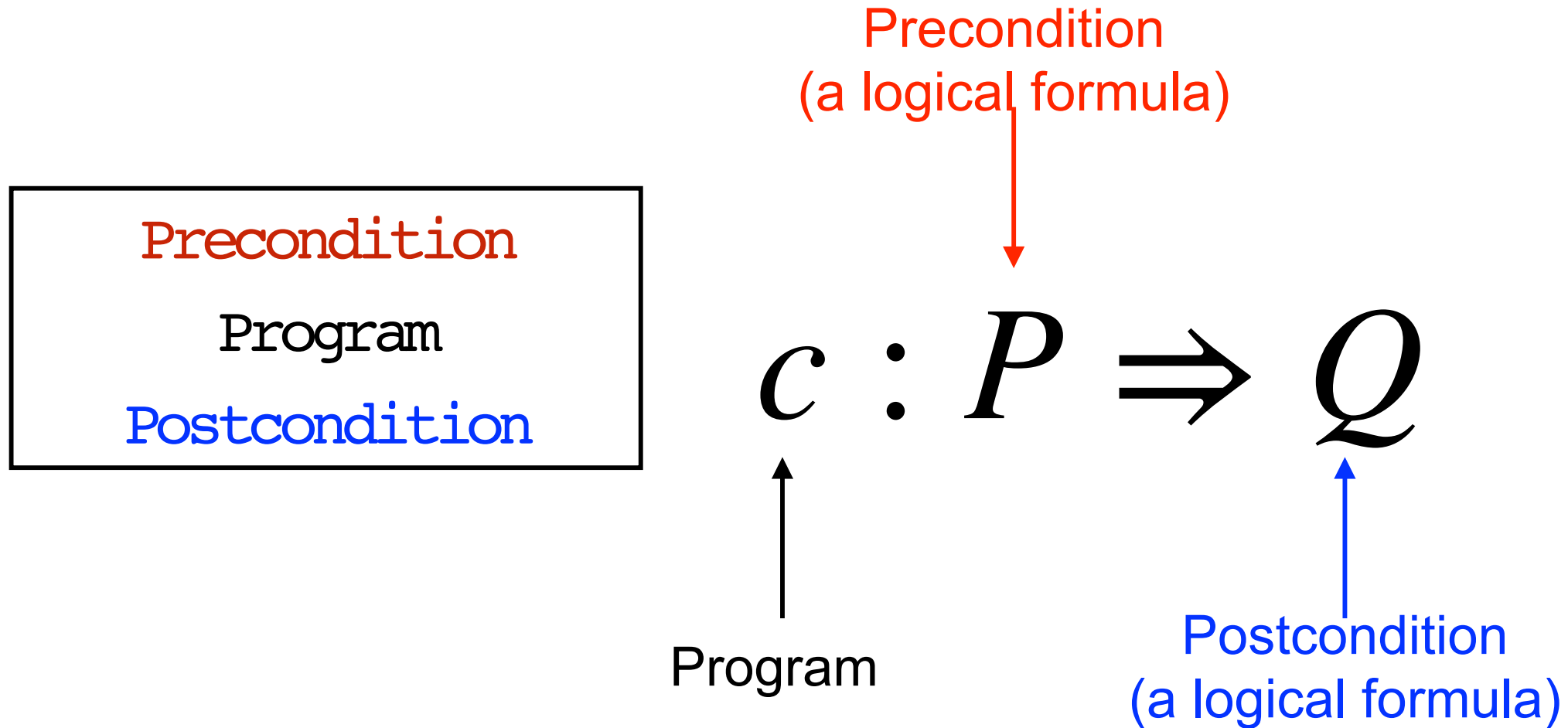
Noninterference

In symbols, c is **noninterferent** if and only if for every $m_1 \sim_{\text{low}} m_2$:

- 1) $\{c\}_{m_1} = \perp$ iff $\{c\}_{m_2} = \perp$
- 2) $\{c\}_{m_1} = m_1'$ and $\{c\}_{m_2} = m_2'$ implies $m_1' \sim_{\text{low}} m_2'$

Is this condition easy to check?

Can we use the tool we studied so far?



Validity of Hoare triple

We say that the triple $c : P \Rightarrow Q$ is valid
if and only if

for every memory m such that $P(m)$
and memory m' such that $\{c\}_m = m'$
we have $Q(m')$.

Validity of Hoare triple

We say that the triple $c : P \Rightarrow Q$ is valid if and only if

for every memory m such that $P(m)$ and memory m' such that $\{c\}_m = m'$ we have $Q(m')$.

Validity talks only about one memory. How can we manage two memories?

Relational Property

In symbols, c is **noninterferent** if and only if for every $m_1 \sim_{\text{low}} m_2$:

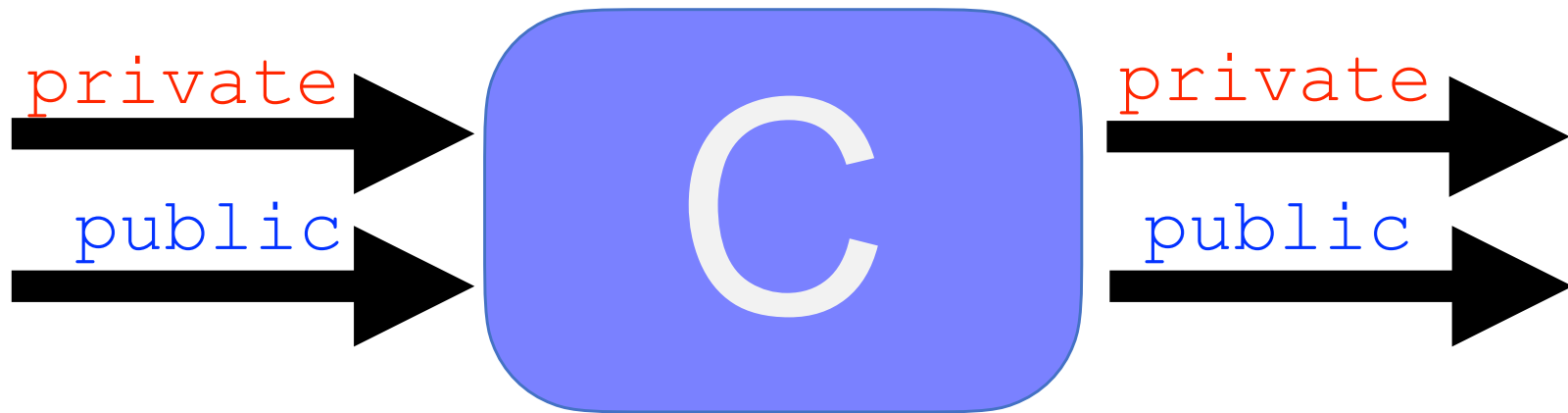
1) $\{c\}_{m_1} = \perp$ iff $\{c\}_{m_2} = \perp$

2) $\{c\}_{m_1} = m_1'$ and $\{c\}_{m_2} = m_2'$ implies $m_1' \sim_{\text{low}} m_2'$

Relational Property

In symbols, c is **noninterferent** if and only if for every $m_1 \sim_{\text{low}} m_2$:

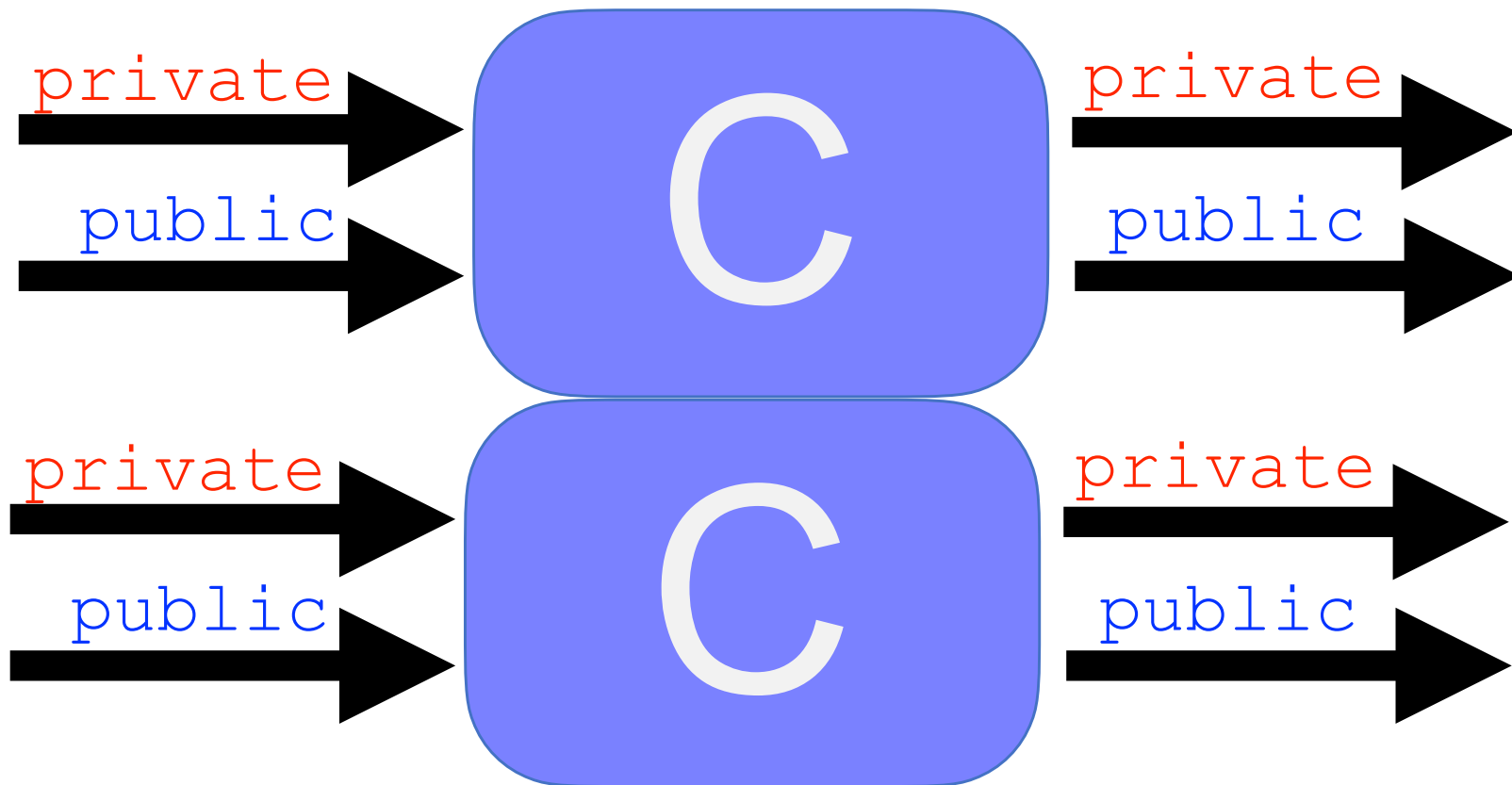
- 1) $\{c\}_{m_1} = \perp$ iff $\{c\}_{m_2} = \perp$
- 2) $\{c\}_{m_1} = m_1'$ and $\{c\}_{m_2} = m_2'$ implies $m_1' \sim_{\text{low}} m_2'$



Relational Property

In symbols, c is **noninterferent** if and only if for every $m_1 \sim_{\text{low}} m_2$:

- 1) $\{c\}_{m_1} = \perp$ iff $\{c\}_{m_2} = \perp$
- 2) $\{c\}_{m_1} = m_1'$ and $\{c\}_{m_2} = m_2'$ implies $m_1' \sim_{\text{low}} m_2'$

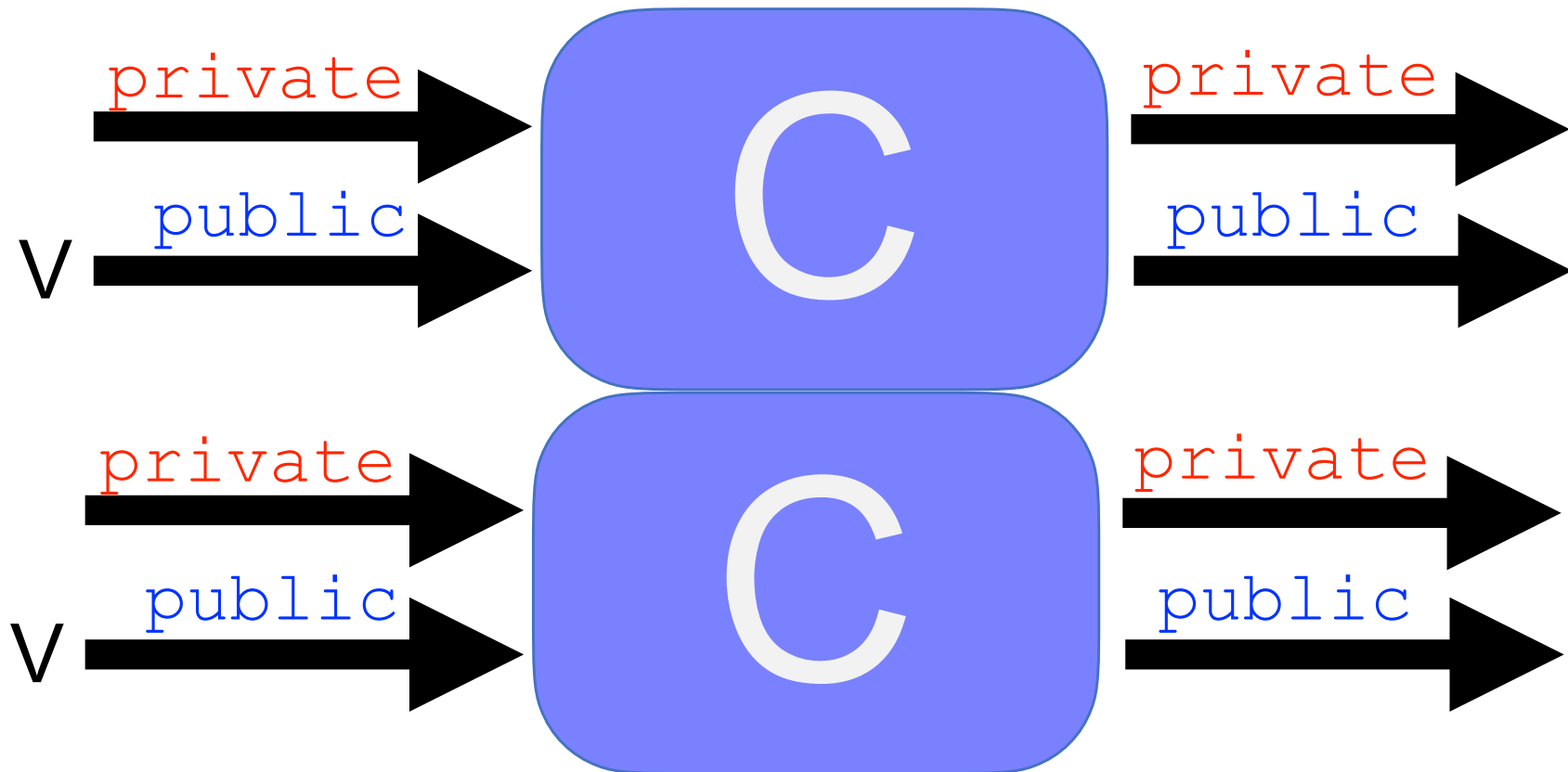


Relational Property

In symbols, c is **noninterferent** if and only if for every $m_1 \sim_{\text{low}} m_2$:

1) $\{c\}_{m_1} = \perp$ iff $\{c\}_{m_2} = \perp$

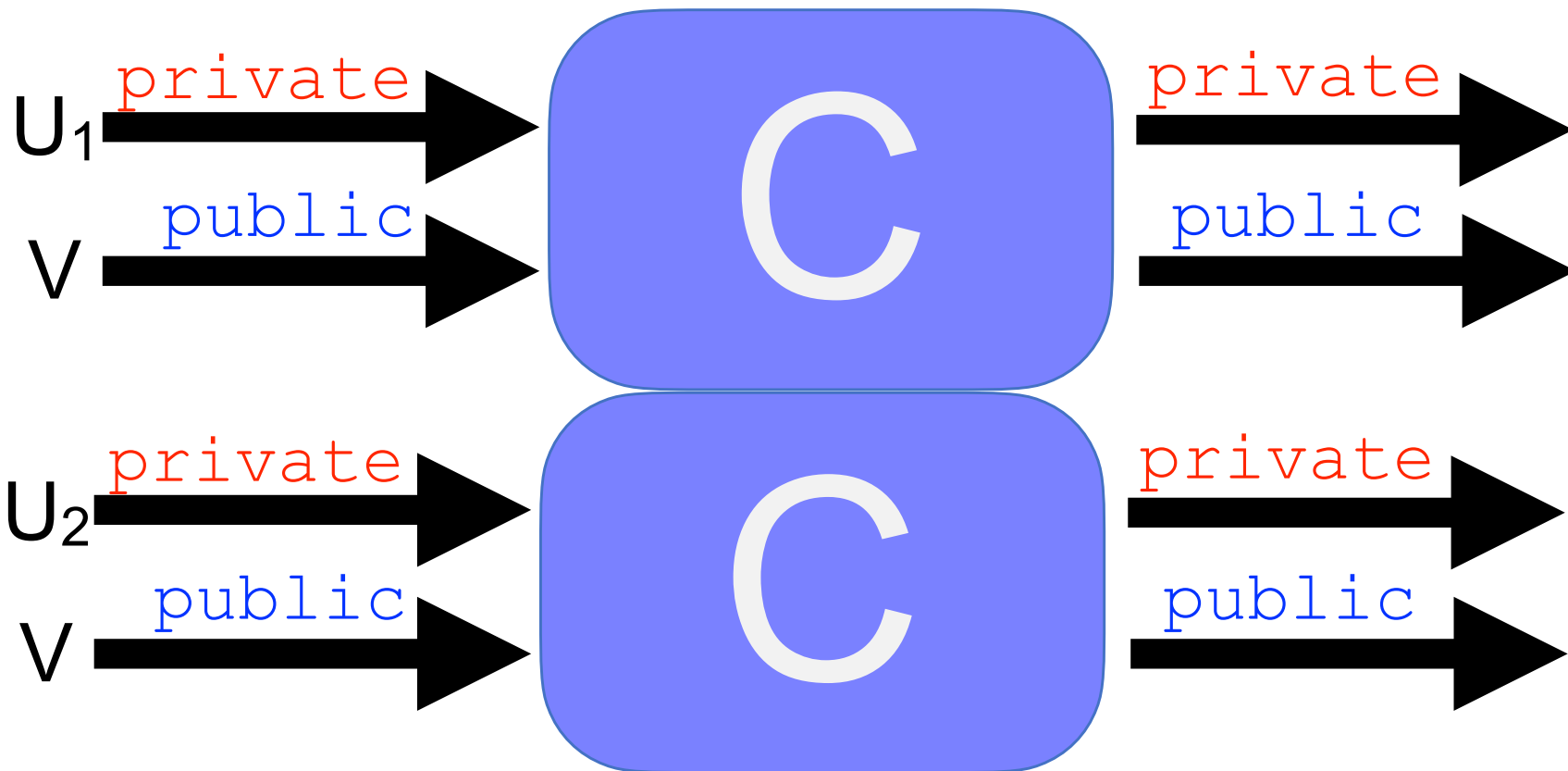
2) $\{c\}_{m_1} = m_1'$ and $\{c\}_{m_2} = m_2'$ implies $m_1' \sim_{\text{low}} m_2'$



Relational Property

In symbols, c is **noninterferent** if and only if for every $m_1 \sim_{\text{low}} m_2$:

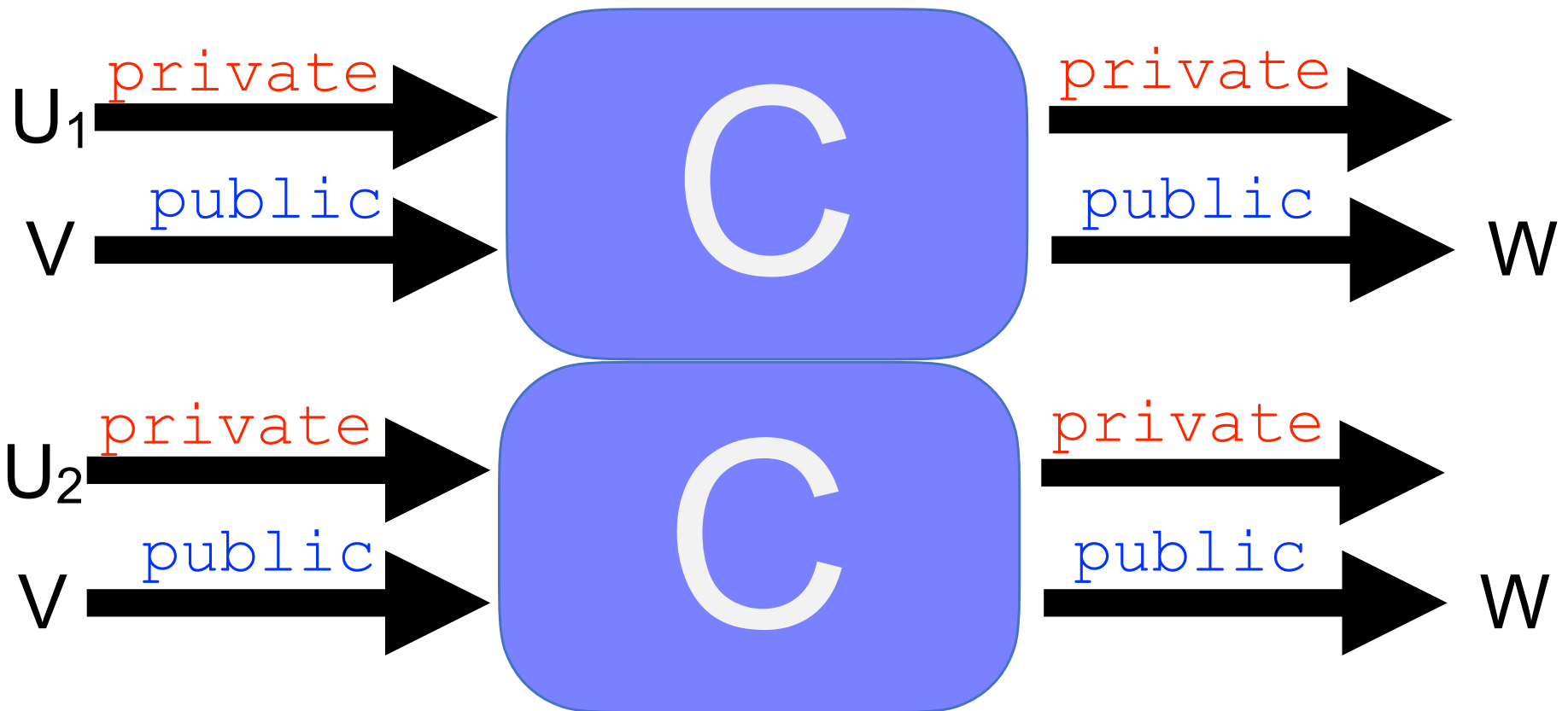
- 1) $\{c\}_{m_1} = \perp$ iff $\{c\}_{m_2} = \perp$
- 2) $\{c\}_{m_1} = m_1'$ and $\{c\}_{m_2} = m_2'$ implies $m_1' \sim_{\text{low}} m_2'$



Relational Property

In symbols, c is **noninterferent** if and only if for every $m_1 \sim_{\text{low}} m_2$:

- 1) $\{c\}_{m_1} = \perp$ iff $\{c\}_{m_2} = \perp$
- 2) $\{c\}_{m_1} = m_1'$ and $\{c\}_{m_2} = m_2'$ implies $m_1' \sim_{\text{low}} m_2'$

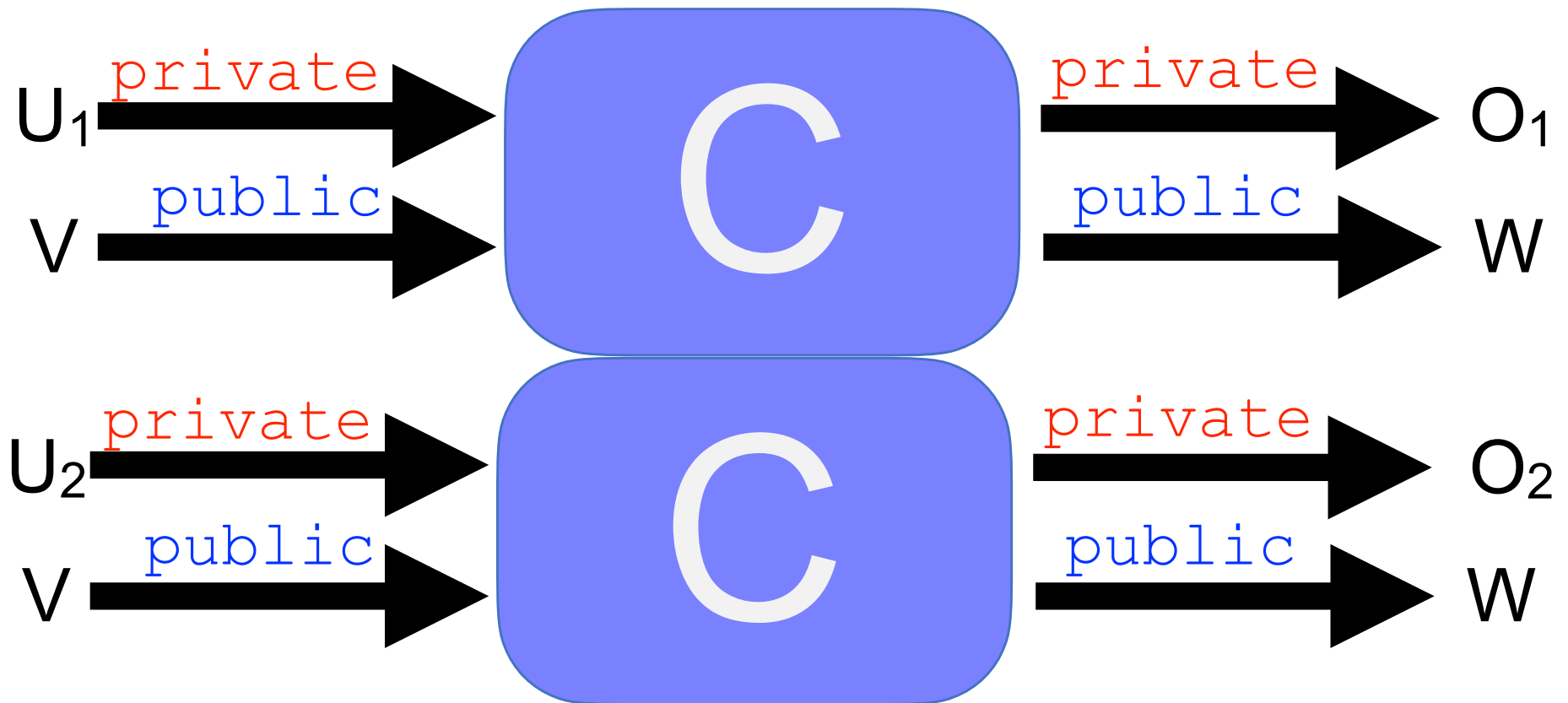


Relational Property

In symbols, c is **noninterferent** if and only if for every $m_1 \sim_{\text{low}} m_2$:

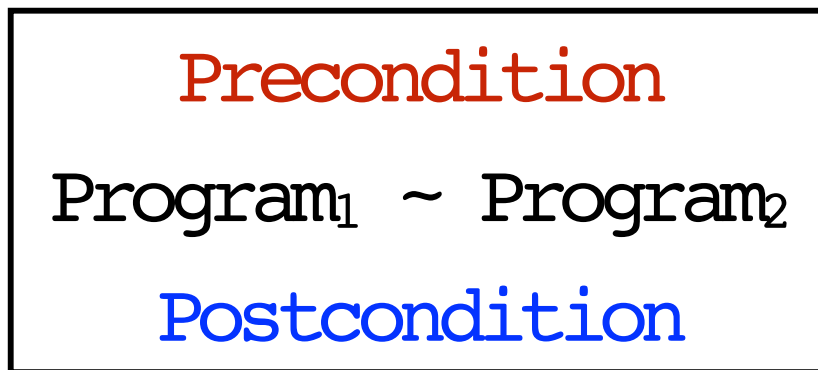
1) $\{c\}_{m_1} = \perp$ iff $\{c\}_{m_2} = \perp$

2) $\{c\}_{m_1} = m_1'$ and $\{c\}_{m_2} = m_2'$ implies $m_1' \sim_{\text{low}} m_2'$



Relational Hoare Logic - RHL

Precondition
(a logical formula)



$$c_1 \sim c_2 : P \Rightarrow Q$$



Program



Program



Postcondition
(a logical formula)

Relational Assertions

$$c_1 \sim c_2 : P \Rightarrow Q$$

Need to talk about variables
of the two memories



Relational Assertions

$$c_1 \sim c_2 : P \Rightarrow Q$$

Need to talk about variables
of the two memories

$$c_1 \sim c_2 : x\langle 1 \rangle \leq x\langle 2 \rangle \Rightarrow x\langle 1 \rangle \geq x\langle 2 \rangle$$

Relational Assertions

$$c_1 \sim c_2 : P \Rightarrow Q$$

Need to talk about variables
of the two memories

$$c_1 \sim c_2 : x\langle 1 \rangle \leq x\langle 2 \rangle \Rightarrow x\langle 1 \rangle \geq x\langle 2 \rangle$$

Tags describing which
memory we are referring to.

Validity of Hoare quadruple

We say that the quadruple $c_1 \sim c_2 : P \Rightarrow Q$ is **valid** if and only if for every pair of memories m_1, m_2 such that $P(m_1, m_2)$ we have:

1) $\{c_1\}_{m_1} = \perp$ iff $\{c_2\}_{m_2} = \perp$

2) $\{c_1\}_{m_1} = m_1'$ and $\{c_2\}_{m_2} = m_2'$ implies $Q(m_1', m_2')$.

Validity of Hoare quadruple

We say that the quadruple $c_1 \sim c_2 : P \Rightarrow Q$ is **valid** if and only if for every pair of memories m_1, m_2 such that $P(m_1, m_2)$ we have:

1) $\{c_1\}_{m_1} = \perp$ iff $\{c_2\}_{m_2} = \perp$

2) $\{c_1\}_{m_1} = m_1'$ and $\{c_2\}_{m_2} = m_2'$ implies $Q(m_1', m_2')$.

Is this easy to check?

Rules of Relational Hoare Logic

Skip

$$\vdash \text{skip} \sim \text{skip} : P \Rightarrow P$$

Correctness of an axiom

We say that an axiom is **correct** if we can prove the **validity** of each instance of the conclusion.

Correctness of an axiom

We say that an axiom is **correct** if we can prove the **validity** of each instance of the conclusion.

Is this still good for RHL?

Correctness of Skip Rule

$$\frac{}{\vdash \text{skip} \sim \text{skip} : P \Rightarrow P}$$

To show this rule **correct** we need to show the **validity of the quadruple** $\text{skip} \sim \text{skip} : P \Rightarrow P$.

Correctness of Skip Rule

$$\frac{}{\vdash \text{skip} \sim \text{skip} : P \Rightarrow P}$$

To show this rule **correct** we need to show the **validity of the quadruple** $\text{skip} \sim \text{skip} : P \Rightarrow P$.

For every m_1, m_2 such that $P(m_1, m_2)$ and m_1', m_2' such that $\{\text{skip}\}_{m_1} = m_1'$ and $\{\text{skip}\}_{m_2} = m_2'$ we need $P(m_1', m_2')$.

Correctness of Skip Rule

$$\frac{}{\vdash \text{skip} \sim \text{skip} : P \Rightarrow P}$$

To show this rule **correct** we need to show the **validity of the quadruple** $\text{skip} \sim \text{skip} : P \Rightarrow P$.

For every m_1, m_2 such that $P(m_1, m_2)$ and m_1', m_2' such that $\{\text{skip}\}_{m_1} = m_1'$ and $\{\text{skip}\}_{m_2} = m_2'$ we need $P(m_1', m_2')$.

Follow easily by our semantics:

$$\{\text{skip}\}_m = m$$

Rules of Relational Hoare Logic

Abort

$\vdash \text{abort} \sim \text{abort} : \text{true} \Rightarrow \text{false}$

Rules of Relational Hoare Logic

Abort

$\vdash \text{abort} \sim \text{abort} : \text{true} \Rightarrow \text{false}$

To show this rule **correct** we need to show the validity of the quadruple $\text{abort} \sim \text{abort} : T \Rightarrow F$.

Rules of Relational Hoare Logic

Abort

$\vdash \text{abort} \sim \text{abort} : \text{true} \Rightarrow \text{false}$

To show this rule **correct** we need to show the **validity of the quadruple** $\text{abort} \sim \text{abort} : T \Rightarrow F$.

For every m_1, m_2 such that **$P(m_1, m_2)$** we can show $\{\text{abort}\}_{m_1} = \perp$ iff $\{\text{abort}\}_{m_2} = \perp$.

Rules of Relational Hoare Logic

Abort

$\vdash \text{abort} \sim \text{abort} : \text{true} \Rightarrow \text{false}$

To show this rule **correct** we need to show the **validity of the quadruple** $\text{abort} \sim \text{abort} : T \Rightarrow F$.

For every m_1, m_2 such that $P(m_1, m_2)$ we can show $\{\text{abort}\}_{m_1} = \perp$ iff $\{\text{abort}\}_{m_2} = \perp$.

Follow easily by our semantics:

$\{\text{abort}\}_m = \perp$

Rules of Relational Hoare Logic

Assignment

$\vdash x_1 := e_1 \sim x_2 := e_2 :$

$P [e_1 \langle 1 \rangle / x_1 \langle 1 \rangle, e_2 \langle 2 \rangle / x_2 \langle 2 \rangle] \Rightarrow P$

Rules of Relational Hoare Logic

Composition

$$\frac{\vdash C_1 \sim C_2 : P \Rightarrow R \quad \vdash C_1' \sim C_2' : R \Rightarrow S}{\vdash C_1 ; C_1' \sim C_2 ; C_2' : P \Rightarrow S}$$

Rules of Relational Hoare Logic

Consequence

$$\frac{P \Rightarrow S \quad \vdash C_1 \sim C_2 : S \Rightarrow R \quad R \Rightarrow Q}{\vdash C_1 \sim C_2 : P \Rightarrow Q}$$

We can **weaken** P , i.e. replace it by something that is implied by P .
In this case S .

We can **strengthen** Q , i.e. replace it by something that implies Q .
In this case R .

Rules of Hoare Logic

If then else

$$\begin{array}{l}
 \vdash c_1 \sim c_2 : e_1 \langle 1 \rangle \wedge e_2 \langle 2 \rangle \wedge P \Rightarrow Q \\
 \vdash c_1' \sim c_2' : \neg e_1 \langle 1 \rangle \wedge \neg e_2 \langle 2 \rangle \wedge P \Rightarrow Q \\
 \hline
 \vdash \begin{array}{l} \text{if } e_1 \text{ then } c_1 \text{ else } c_1' \\ \sim \\ \text{if } e_2 \text{ then } c_2 \text{ else } c_2' \end{array} : P \Rightarrow Q
 \end{array}$$

Rules of Hoare Logic

If then else

$$\begin{array}{l} \vdash c_1 \sim c_2 : e_1 \langle 1 \rangle \wedge e_2 \langle 2 \rangle \wedge P \Rightarrow Q \\ \vdash c_1' \sim c_2' : \neg e_1 \langle 1 \rangle \wedge \neg e_2 \langle 2 \rangle \wedge P \Rightarrow Q \end{array}$$

$$\vdash \begin{array}{l} \text{if } e_1 \text{ then } c_1 \text{ else } c_1' \\ \sim \\ \text{if } e_2 \text{ then } c_2 \text{ else } c_2' \end{array} : P \Rightarrow Q$$

Is this correct?

An example

$\vdash$ $\begin{array}{l} \text{if true then skip else } x:=x+1 \\ \sim \\ \text{if false then } x:=x+1 \text{ else skip} \end{array} : \{x=n\} \Rightarrow \{x=n+1\}$

Is this a valid quadruple?

An example

$$\vdash \begin{array}{l} \text{if true then skip else } x:=x+1 \\ \sim \\ \text{if false then } x:=x+1 \text{ else skip} \end{array} : \{x=n\} \Rightarrow \{x=n+1\}$$

Is this a valid quadruple?



An example

$\vdash$ $\begin{array}{l} \text{if true then skip else } x:=x+1 \\ \sim \\ \text{if false then } x:=x+1 \text{ else skip} \end{array}$ $: \{x=n\} \Rightarrow \{x=n+1\}$

Is this a valid quadruple?

Can we prove it with the
rule above?



An example

$$\vdash \begin{array}{l} \text{if true then skip else } x:=x+1 \\ \sim \\ \text{if false then } x:=x+1 \text{ else skip} \end{array} : \{x=n\} \Rightarrow \{x=n+1\}$$

Is this a valid quadruple?

Can we prove it with the rule above?



Rules of Relational Hoare Logic

If then else

$$P \Rightarrow e_1 \langle 1 \rangle = e_2 \langle 2 \rangle$$

$$\vdash c_1 \sim c_2 : e_1 \langle 1 \rangle \wedge P \Rightarrow Q$$

$$\vdash c_1' \sim c_2' : \neg e_1 \langle 1 \rangle \wedge P \Rightarrow Q$$

$$\vdash \begin{array}{c} \text{if } e_1 \text{ then } c_1 \text{ else } c_1' \\ \sim \\ \text{if } e_2 \text{ then } c_2 \text{ else } c_2' \end{array} : P \Rightarrow Q$$

Rules of Hoare Logic

While

$$P \Rightarrow e_1 \langle 1 \rangle = e_2 \langle 2 \rangle$$

$$\vdash c_1 \sim c_2 \quad : \quad e_1 \langle 1 \rangle \wedge P \Rightarrow P$$

$$\vdash \begin{array}{l} \text{while } e_1 \text{ do } c_1 \\ \sim \\ \text{while } e_2 \text{ do } c_2 \end{array} : P \Rightarrow P \wedge \neg e_1 \langle 1 \rangle$$


Invariant

How can we prove this?

```
x:private  
y:public
```

```
x:=y
```

$\sim$

```
x:=y
```

```
: =low  $\Rightarrow$  =low
```

How can we prove this?

```
x:private  
y:public
```

```
y:=x
```

```
: =low ⇒ ¬ (=low)
```

How can we prove this?

```
x:private  
y:public
```

```
y:=x
```

```
: =low ⇒ ¬ (=low)
```

Can we prove it?

How can we prove this?

```
x:private  
y:public
```

```
y:=x  
y:=5
```

```
: =low ⇒ =low
```