CS 591: Formal Methods in Security and Privacy

Noninterference and Relational Hoare Logic

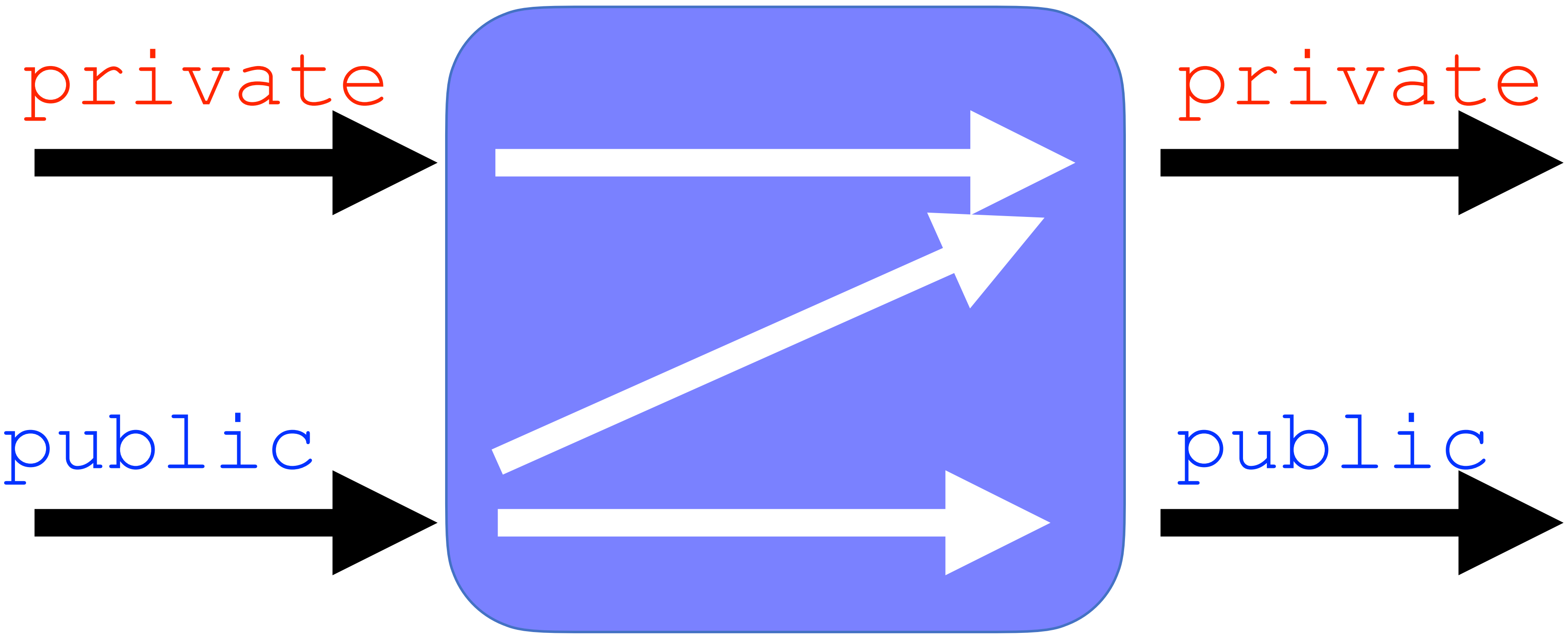
Marco Gaboardi
gaboardi@bu.edu

Alley Stoughton
stough@bu.edu

From the previous classes

Information Flow Control

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



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: $m_1 \sim_{low} m_2$

NonInterference

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

1) $\{c\}_{m1}=\perp$ iff $\{c\}_{m2}=\perp$

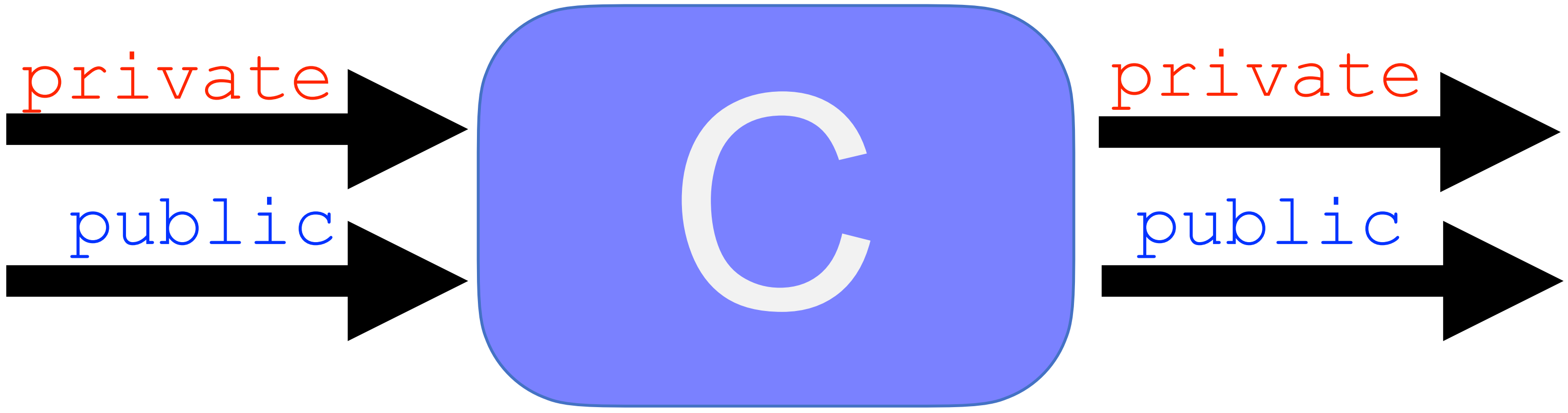
2) $\{c\}_{m1}=m_1'$ and $\{c\}_{m2}=m_2'$ implies $m_1' \sim_{low} m_2'$

NonInterference

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

1) $\{c\}_{m1} = \perp$ iff $\{c\}_{m2} = \perp$

2) $\{c\}_{m1} = m_1'$ and $\{c\}_{m2} = m_2'$ implies $m_1' \sim_{low} m_2'$

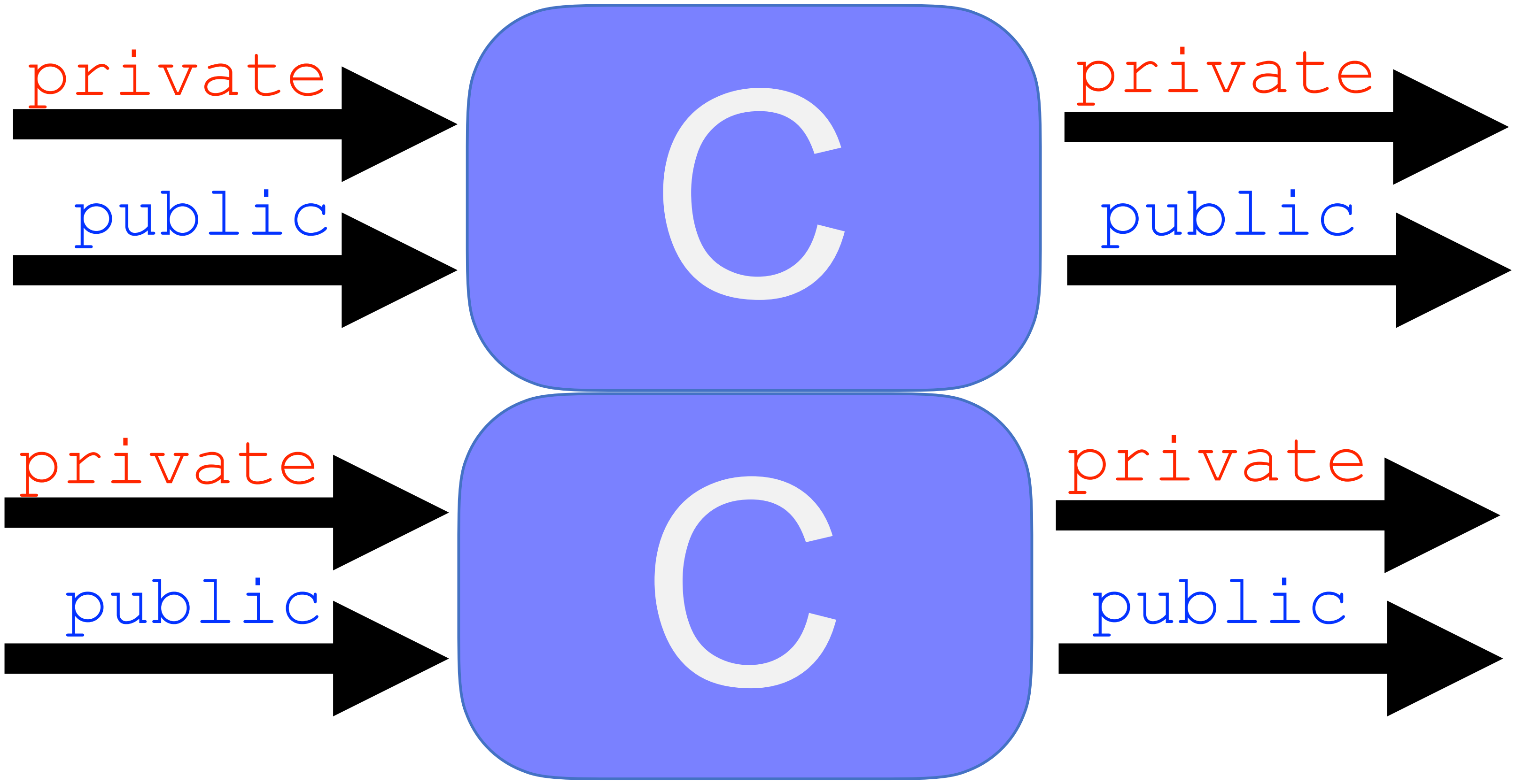


NonInterference

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

1) $\{c\}_{m1} = \perp$ iff $\{c\}_{m2} = \perp$

2) $\{c\}_{m1} = m_1'$ and $\{c\}_{m2} = m_2'$ implies $m_1' \sim_{low} m_2'$

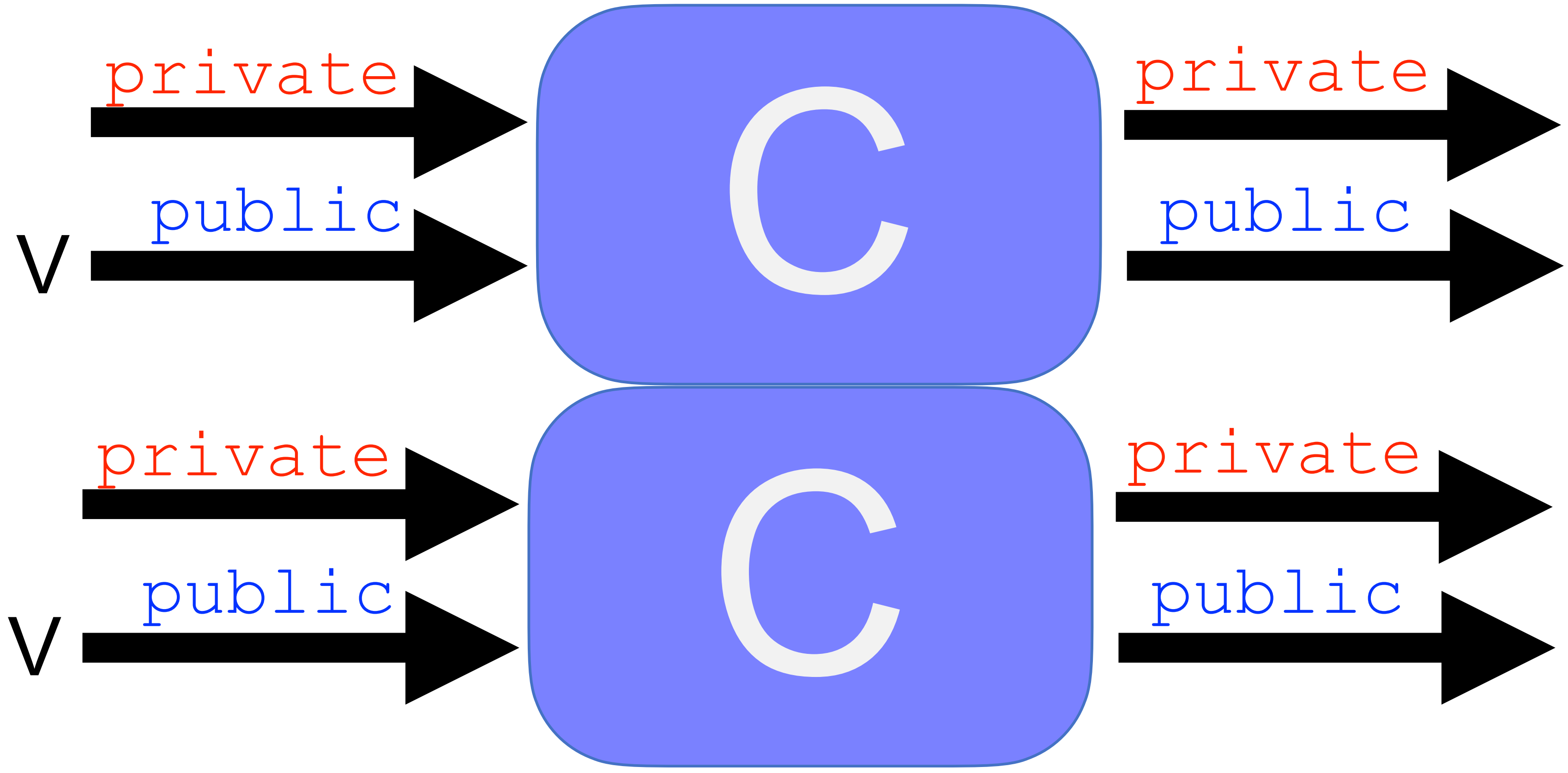


NonInterference

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

1) $\{c\}_{m1}=\perp$ iff $\{c\}_{m2}=\perp$

2) $\{c\}_{m1}=m_1'$ and $\{c\}_{m2}=m_2'$ implies $m_1' \sim_{low} m_2'$

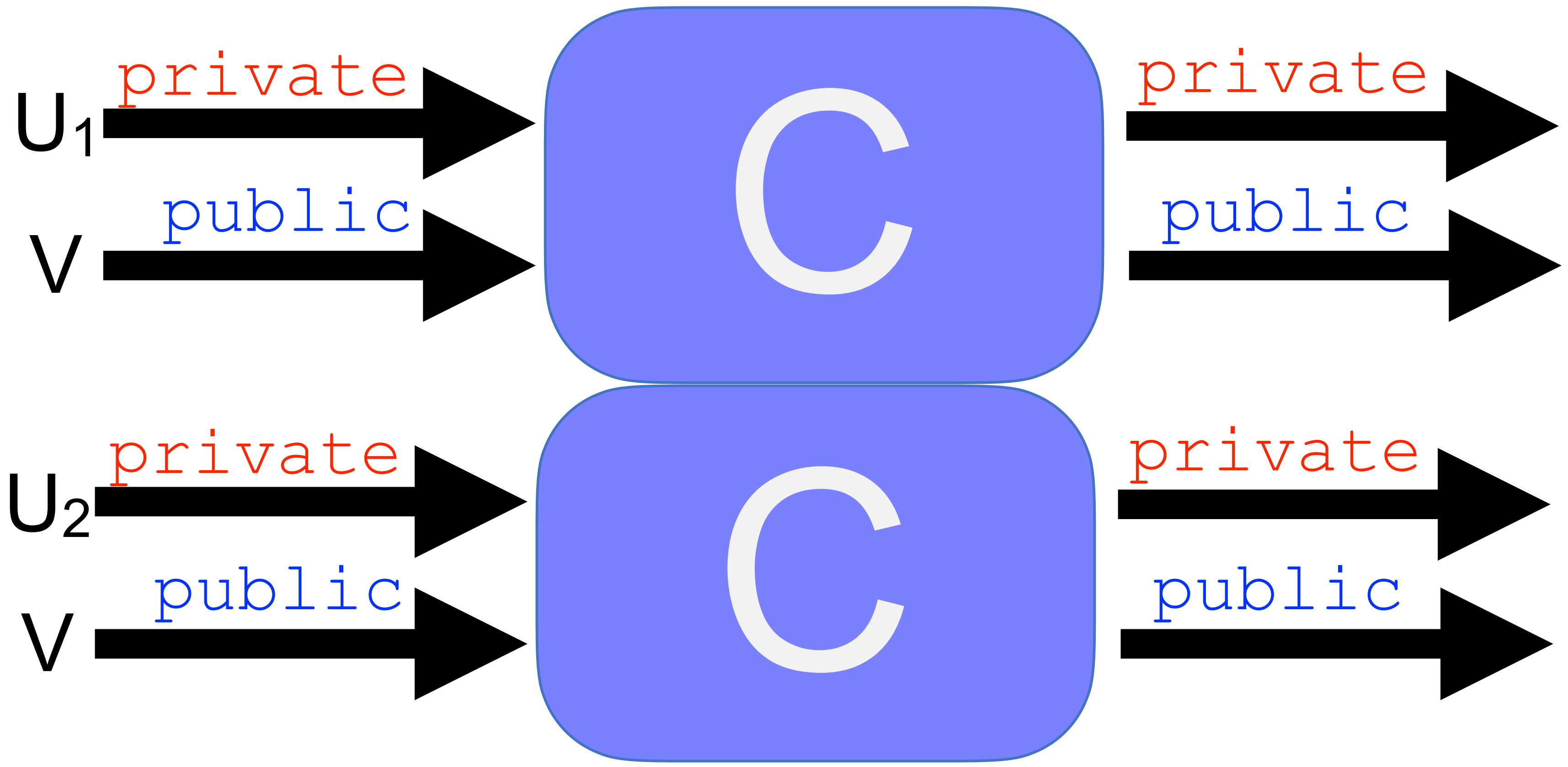


NonInterference

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

1) $\{c\}_{m1}=\perp$ iff $\{c\}_{m2}=\perp$

2) $\{c\}_{m1}=m_1'$ and $\{c\}_{m2}=m_2'$ implies $m_1' \sim_{low} m_2'$

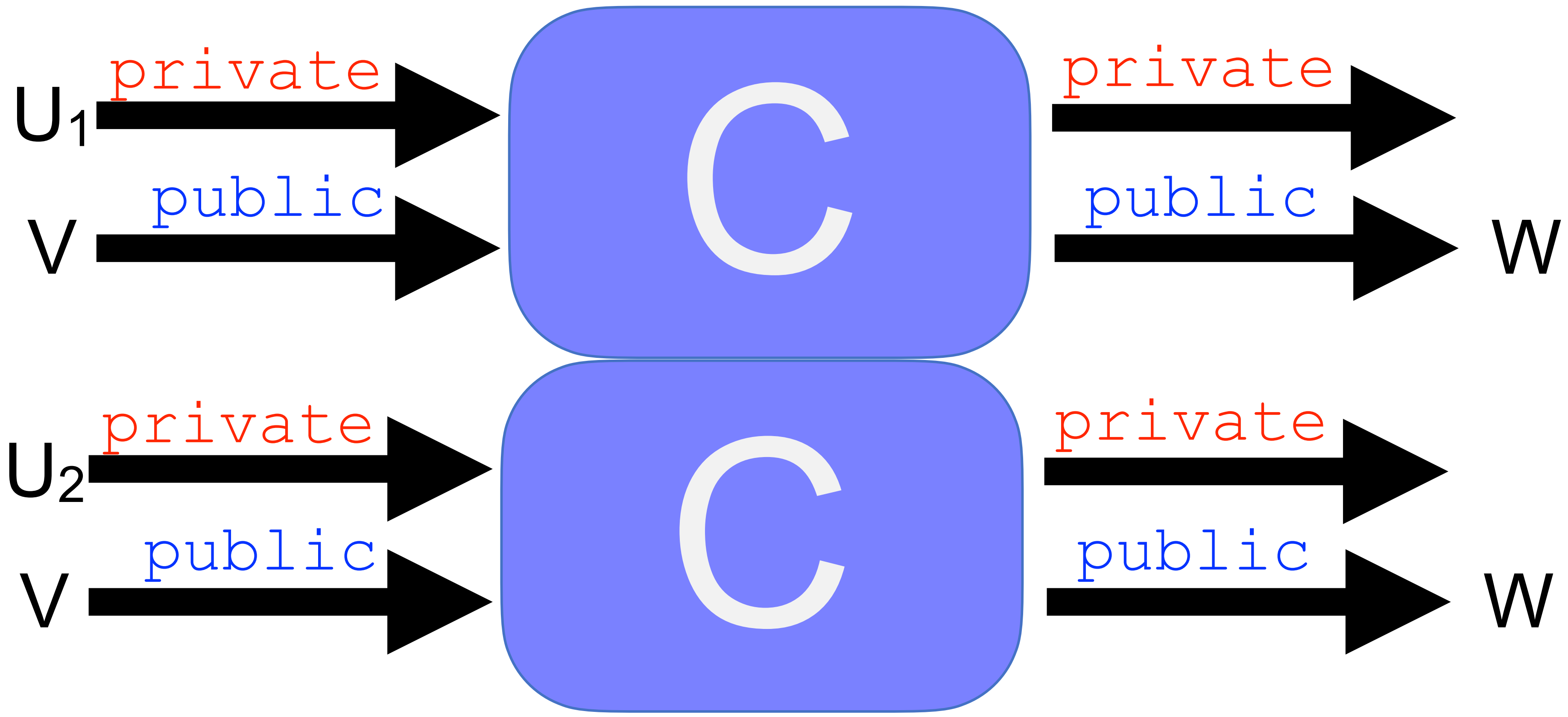


NonInterference

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

1) $\{c\}_{m1}=\perp$ iff $\{c\}_{m2}=\perp$

2) $\{c\}_{m1}=m_1'$ and $\{c\}_{m2}=m_2'$ implies $m_1' \sim_{low} m_2'$

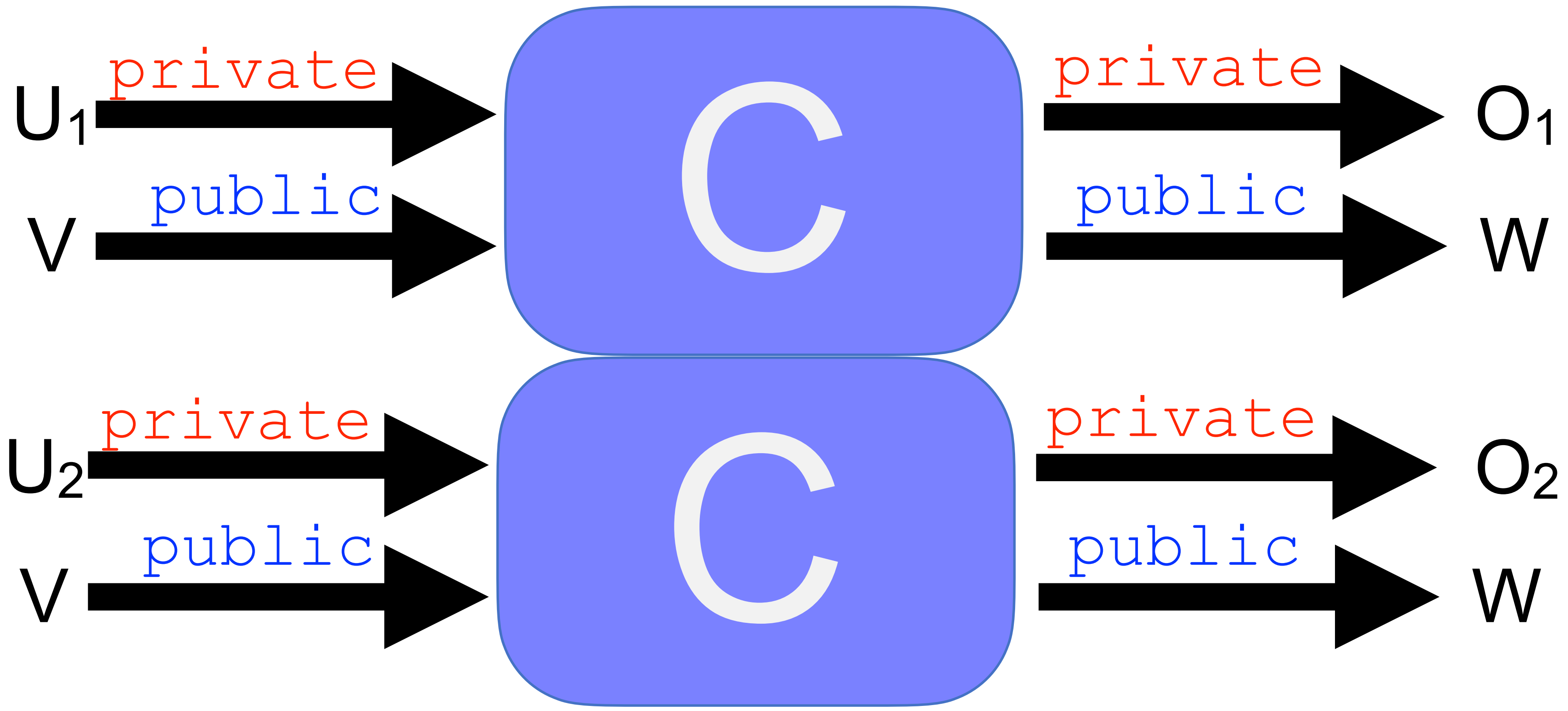


NonInterference

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

1) $\{c\}_{m1}=\perp$ iff $\{c\}_{m2}=\perp$

2) $\{c\}_{m1}=m_1'$ and $\{c\}_{m2}=m_2'$ implies $m_1' \sim_{low} m_2'$



Relational Hoare Logic - RHL

Precondition

$\text{Program}_1 \sim \text{Program}_2$

Postcondition

$$\underset{\underset{\text{Program}}{\uparrow}}{c_1} \sim \underset{\underset{\text{Program}}{\uparrow}}{c_2} : \overset{\overset{\text{Precondition (a logical formula)}}{\downarrow}}{P} \Rightarrow \underset{\underset{\text{Postcondition (a logical formula)}}{\uparrow}}{Q}$$

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) $\{c_1\}_{m1} = \perp$ iff $\{c_2\}_{m2} = \perp$
2) $\{c_1\}_{m1} = m_1'$ and $\{c_2\}_{m2} = m_2'$ implies $Q(m_1', m_2')$.

Some Rules of Relational Hoare Logic

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

$$\frac{}{\vdash \texttt{abort} \sim \texttt{abort} : \texttt{true} \Rightarrow \texttt{false}}$$

$$\frac{}{\vdash x_1 := e_1 \sim x_2 := e_2 : P[e_1\langle 1\rangle / x_1\langle 1\rangle, e_2\langle 2\rangle / x_2\langle 2\rangle] \Rightarrow P}$$

$$\frac{\vdash c_1 \sim c_2 : P \Rightarrow R \qquad \vdash c_1' \sim c_2' : R \Rightarrow S}{\vdash c_1 ; c_1' \sim c_2 ; c_2' : P \Rightarrow S}$$

$$\frac{P \Rightarrow S \qquad \vdash c_1 \sim c_2 : S \Rightarrow R \qquad R \Rightarrow Q}{\vdash c_1 \sim c_2 : P \Rightarrow Q}$$

Today: More Relational Hoare Logic

Rules of Relational Hoare Logic

Assignment Example

$\vdash$`x:=x+1 ~ y:=y-1:`

`x<1>+1=-(y<2>-1)` $\Rightarrow$ `x<1>=-y<2>`

Rules of Relational Hoare Logic

Assignment Example

```
⊢x:=x+1 ~ y:=y-1:
 (x<1>=-y<2>)
 [(x+1)<1>/x<1>,(y-1)<2>/y<2>]
⇒
x<1> = -y<2>
```

Rules of Relational Hoare Logic

Assignment Example

$\vdash$`x:=x+1 ~ y:=y-1:`

```
 (x<1> = -y<2>)
 [(x<1>+1)/x<1>,(y<2>-1)/y<2>]
```

$\Rightarrow$

```
x<1> = -y<2>
```

Consequence + Assignment

Example

$$\vdash \texttt{x:=x+1} \sim \texttt{y:=y-1} : \texttt{x<1>=-y<2>} \Rightarrow \texttt{x<1>=-y<2>}$$

Consequence + Assignment

Example

$$\frac{\begin{array}{c} x\langle 1\rangle=-y\langle 2\rangle \Rightarrow x\langle 1\rangle+1=-(y\langle 2\rangle-1) \\ \vdash x:=x+1 \sim y:=y-1 : \\ x\langle 1\rangle+1=-(y\langle 2\rangle-1) \Rightarrow x\langle 1\rangle=-y\langle 2\rangle \\ x\langle 1\rangle=-y\langle 2\rangle \Rightarrow x\langle 1\rangle=-y\langle 2\rangle \end{array}}{\begin{array}{c} \vdash x:=x+1 \sim y:=y-1 : \\ x\langle 1\rangle=-y\langle 2\rangle \Rightarrow x\langle 1\rangle=-y\langle 2\rangle \end{array}}$$

Rules of Hoare Logic
If then else

$$\frac{\vdash c_1 \sim c_2 : e_1\langle 1\rangle \wedge e_2\langle 2\rangle \wedge P \Rightarrow Q \qquad \vdash c_1' \sim c_2' : \neg e_1\langle 1\rangle \wedge \neg e_2\langle 2\rangle \wedge P \Rightarrow Q}{\vdash \begin{array}{c}\texttt{if } e_1 \texttt{ then } c_1 \texttt{ else } c_1' \\ \sim \\ \texttt{if } e_2 \texttt{ then } c_2 \texttt{ else } c_2'\end{array} : P \Rightarrow Q}$$

Is this correct?

An example

$$\vdash \begin{array}{c} \texttt{if true then x:=x else x:=x+1} \\ \sim \\ \texttt{if false then x:=x+1 else x:=x} \end{array} : \begin{array}{c} \{\texttt{x<1>=n}\} \\ \Rightarrow \\ \{\texttt{x<1>=n+1}\} \end{array}$$

$\boxed{\text{Is this a valid quadruple?}}$

An example

$$\vdash \begin{array}{l}\texttt{if true then x:=x else x:=x+1} \\ \qquad\qquad \sim \\ \texttt{if false then x:=x+1 else x:=x}\end{array} : \begin{array}{l}\{x\langle 1\rangle = n\} \\ \quad\Rightarrow \\ \{x\langle 1\rangle = n+1\}\end{array}$$

$\boxed{\text{Is this a valid quadruple?}}$ ✗

An example

$$\vdash \begin{array}{c}\texttt{if true then x:=x else x:=x+1} \\ \sim \\ \texttt{if false then x:=x+1 else x:=x}\end{array} : \begin{array}{l}\{\texttt{x<1>=n}\} \\ \Rightarrow \\ \{\texttt{x<1>=n+1}\}\end{array}$$

Is this a valid quadruple? ✗

Can we prove it with the rule above?

An example

$$\vdash \begin{array}{c} \texttt{if true then x:=x else x:=x+1} \\ \sim \\ \texttt{if false then x:=x+1 else x:=x} \end{array} : \begin{array}{l} \{\texttt{x<1>=n}\} \\ \Rightarrow \\ \{\texttt{x<1>=n+1}\} \end{array}$$

Is this a valid quadruple? ✗

Can we prove it with the rule above? ✓

Rules of Relational Hoare Logic
If then else

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

Rules of Hoare Logic While

$$\frac{P \Rightarrow e_1\langle 1\rangle = e2\langle 2\rangle \qquad \vdash c_1 \sim c_2 : e_1\langle 1\rangle \wedge P \Rightarrow P}{\vdash \begin{array}{c}\texttt{while } e_1 \texttt{ do } c_1 \\ \sim \\ \texttt{while } e_2 \texttt{ do } c_2\end{array} : P \Rightarrow P \wedge \neg e_1\langle 1\rangle}$$

Invariant

How can we prove this?

```
x:private
y:public

x:=y
```

$: \quad =_{\mathtt{low}} \Rightarrow \; =_{\mathtt{low}}$

How can we prove this?

```
x:private
y:public

x:=y

:y<1>=y<2> ⇒ y<1>=y<2>
```

Assignment

```
⊢x:=y ~ x:=y:
(y<1>=y<2>)[y<1>/x<1>,y<2>/y<2>]
⇒
y<1> = y<2>
```

Assignment

```
⊢x:=y ~ x:=y:
y<1>=y<2>
⇒
y<1> = y<2>
```

How can we prove this?

```
x:private
y:public

y:=x
```

$: \; =_{\mathtt{low}} \Rightarrow \neg(=_{\mathtt{low}})$

How can we prove this?

```
x:private
y:public

y:=x
```

$: \; =_{low} \Rightarrow \neg(=_{low})$

Can we prove it?

How can we prove this?

```
x:private
y:public

y:=x

:y<1>=y<2> ⇒¬(y<1>=y<2>)
```

Can we prove it?

How can we prove this?

```
x:private
y:public

y:=x
y:=5
```

$: \quad =_{\mathtt{low}} \Rightarrow \quad =_{\mathtt{low}}$

How can we prove this?

```
x:private
y:public

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

$: \quad =_{\mathrm{low}} \Rightarrow \; =_{\mathrm{low}}$

How can we prove this?

```
x:private
y:public

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

$: \quad =_{\mathtt{low}} \Rightarrow \; =_{\mathtt{low}}$

How can we prove this?

```
x:private
y:public

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

Can we prove it?

$: \;\; =_{low} \Rightarrow \; =_{low}$

Rules of Relational Hoare Logic
If then else

$$\frac{P \Rightarrow e_1\langle 1\rangle = e_2\langle 2\rangle \qquad \vdash c_1 \sim c_2 : e_1\langle 1\rangle \land P \Rightarrow Q \qquad \vdash c_1' \sim c_2' : \neg e_1\langle 1\rangle \land P \Rightarrow Q}{\vdash \begin{array}{c}\texttt{if } e_1 \texttt{ then } c_1 \texttt{ else } c_1' \\ \sim \\ \texttt{if } e_2 \texttt{ then } c_2 \texttt{ else } c_2'\end{array} : P \Rightarrow Q}$$

Rules of Relational Hoare Logic
If then else - left

$$\frac{\vdash c_1 \sim c_2 : e\langle 1\rangle \wedge P \Rightarrow Q \qquad \vdash c_1' \sim c_2 : \neg e\langle 1\rangle \wedge P \Rightarrow Q}{\vdash \begin{array}{c}\texttt{if } e \texttt{ then } c_1 \texttt{ else } c_1' \\ \sim \\ c_2\end{array} : P \Rightarrow Q}$$

Rules of Relational Hoare Logic
If then else - left

$$\dfrac{\vdash c_1 \sim c_2 : e\langle 2\rangle \wedge P \Rightarrow Q \qquad \vdash c_1 \sim c_2' : \neg e\langle 2\rangle \wedge P \Rightarrow Q}{\vdash \begin{array}{c} c_1 \\ \sim \\ \texttt{if } e \texttt{ then } c_2 \texttt{ else } c_2' \end{array} : P \Rightarrow Q}$$

How can we prove this?

```
x:private
y:public

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

$: \; =_{\mathrm{low}} \Rightarrow \; =_{\mathrm{low}}$

How can we prove this?

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

y:=0
z:=0
if x=0 then z:=1;
if z=0 then y:=1
```

$: \quad =_{low} \Rightarrow \quad =_{low}$

How can we prove this?

```
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 do
 if not(s1[i]=s2[i]) then
     r:=1
 i:=i+1

: n>0 /\ =low ⇒ =low
```

Rules of Relational Hoare-Logic

One-sided Rules

What do we do if our two programs have different forms? There are three pairs of *one-sided* rules.

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

Rules of Relational Hoare Logic

If-then-else — left

$$\frac{\vdash c_1 \sim c_2 : e\langle 1\rangle \land P \Rightarrow Q \qquad \vdash c_1' \sim c_2 : \neg e\langle 1\rangle \land P \Rightarrow Q}{\vdash \begin{array}{c}\texttt{if } e \texttt{ then } c_1 \texttt{ else } c_1' \\ \sim \\ c_2\end{array} : P \Rightarrow Q}$$

Rules of Relational Hoare Logic

If-then-else — right

$$\frac{\vdash c_1 \sim c_2 : e\langle 2\rangle \land P \Rightarrow Q \qquad \vdash c_1 \sim c_2' : \neg e\langle 2\rangle \land P \Rightarrow Q}{\vdash \begin{array}{c} c_1 \\ \sim \\ \texttt{if } e \texttt{ then } c_2 \texttt{ else } c_2' \end{array} : P \Rightarrow Q}$$

Rules of Relational Hoare Logic

Assignment — left

$$\frac{}{\vdash \texttt{x:=e} \sim \texttt{skip}: \quad \texttt{P[e<1>/x<1>]} \Rightarrow \texttt{P}}$$

Rules of Relational Hoare Logic

Assignment — right

$$\frac{}{\vdash \texttt{skip} \sim \texttt{x:=e} : \texttt{P[e<2>/x<2>]} \Rightarrow \texttt{P}}$$

Also pair of one-sided rules for while — we'll ignore for now

Rules of Relational Hoare Logic

Program Equivalence Rule

$\models P : c_1 \equiv c_2$ means $\{c_1\}_m = \{c_2\}_m$ for all m such that $P(m)$

$$\frac{\models P : c_1' \equiv c_1 \qquad \models P : c_2' \equiv c_2 \qquad c_1' \sim c_2' : P \Rightarrow Q}{\vdash c_1 \sim c_2 : P \Rightarrow Q}$$

Rules of Relational Hoare Logic

Program Equivalences

```
⊨P : skip;c ≡ c

⊨P : c;skip ≡ c

⊨P:(c1;c2);c3 ≡ c1;(c2;c3)
```

…

Rules of Relational Hoare Logic

Combining Composition and Equivalence

We can combine the Composition and Program Equivalence Rules to split commands where we like:

$$\frac{\vdash c_1;c_2 \sim c_1' : P \Rightarrow R \qquad \vdash c_3 \sim c_2';c_3' : R \Rightarrow Q}{\vdash c_1;c_2;c_3 \sim c_1';c_2';c_3' : P \Rightarrow Q}$$

Rules of Relational Hoare Logic

Combining Composition and Equivalence

$$\frac{\dfrac{\vdash \texttt{c}_1 \sim \texttt{skip} : P \Rightarrow R \qquad \vdash \texttt{c}_2 \sim \texttt{c}_1' : R \Rightarrow Q}{\vdash \texttt{c}_1;\texttt{c}_2 \sim \texttt{skip};\texttt{c}_1' : P \Rightarrow Q}}{\vdash \texttt{c}_1;\texttt{c}_2 \sim \texttt{c}_1' : P \Rightarrow Q}$$

Rules of Relational Hoare Logic

Combining Composition and Equivalence

$$\dfrac{\dfrac{\vdash \texttt{c}_1 \sim \texttt{c}_1\texttt{'}: \texttt{P} \Rightarrow \texttt{R} \qquad \vdash \texttt{c}_2 \sim \texttt{skip}: \texttt{R} \Rightarrow \texttt{Q}}{\vdash \texttt{c}_1\texttt{;c}_2 \sim \texttt{c}_1\texttt{';skip}: \texttt{P} \Rightarrow \texttt{Q}}}{\vdash \texttt{c}_1\texttt{;c}_2 \sim \texttt{c}_1\texttt{'}: \texttt{P} \Rightarrow \texttt{Q}}$$

Relational Hoare Logic in EasyCrypt

- EasyCrypt’s implementation of Relational Hoare Logic has much in common with its implementation of Hoare Logic.
- Look for the pRHL tactics in Section 3.4 of the EasyCrypt Reference Manual (the “p” stands for “probabilistic”, but ignore that for now).