Information Flow Control

Language-Based Security
aslan@cs.au.dk

AU PLS summer school, August 2026

Plan for today

- System model = simple imperative language
  - Single-threaded, deterministic
- Simple model of just 2 security levels: public and secret
- Recall: **Policy**/Mechanism/Assurance
  - A *noninterference* security policy for confidentiality:
    - public behavior should not reveal any information about secrets

Simple imperative language

The grammar

Commands

$$c ::= \text{skip} \mid x := e \mid c; c \mid \text{if } e \text{ then } c \text{ else } c \mid \text{while } e \text{ do } c$$

Expressions

$$e ::= n \mid x \mid e \ op \ e$$

Example program `while x>0 do x:=x-1`

Values are only natural numbers

Memory *m* - function from variable names to values

Example: starting from initial memory m such that m(x)=5, the evaluation of our example program results in final memory m', such that m'(x)=0

Formalizing the evaluation: expressions

Constant expression *n* evaluates to value *n* in any memory *m*

Variable expression *x* evaluates to value *v* in memory *m* when *m(x)=v*

E-CONST

$$\frac{}{\langle n, m\rangle \Downarrow n}$$

E-VAR

$$\frac{m(x) = v}{\langle x, m\rangle \Downarrow v}$$

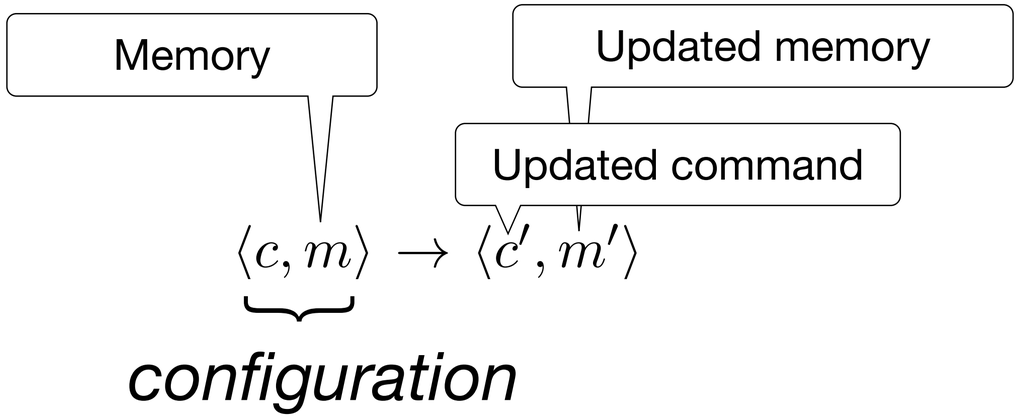
E-OP

$$\frac{\langle e_i, m\rangle \Downarrow v_i, i = 1, 2 \qquad v = v_1 \ \mathbf{op}\ v_2}{\langle e_1\ op\ e_2, m\rangle \Downarrow v}$$

Expressions of the form e_1 *op* e_2, where op is a binary operator such as (+,-,<,==, etc) evaluates to value v if e_1 *evaluates to* v_1, *and* e_2 *evaluates to* v_2, *and v equals* v_1 *op* v_2

Q: what's the expression derivation tree for $\langle x + 1, m \rangle$ when $m(x)=42$

Formalizing the evaluation: step relation



Memory update $m[x \mapsto v] \triangleq \lambda y \ . \ \text{if } x = y \text{ then } v \text{ else } m(y)$

Reachability in zero or more steps $\langle c, m \rangle \to^* \langle c', m' \rangle$

does not appear in source syntax

Terminal configuration: $\langle \mathsf{stop}, m \rangle$

Step relation

S-SKIP

$$\frac{}{\langle \mathsf{skip}, m\rangle \to \langle \mathsf{stop}, m\rangle}$$

S-ASSIGN

$$\frac{\langle e, m\rangle \Downarrow v}{\langle x := e, m\rangle \to \langle \mathsf{stop}, m[x \mapsto v]\rangle}$$

S-SEQ1

$$\frac{\langle c_1, m\rangle \to \langle \mathsf{stop}, m'\rangle}{\langle c_1; c_2, m\rangle \to \langle c_2, m'\rangle}$$

S-SEQ2

$$\frac{\langle c_1, m\rangle \to \langle c_1', m'\rangle \qquad c_1' \neq \mathsf{stop}}{\langle c_1; c_2, m\rangle \to \langle c_1; c_2, m'\rangle}$$

S-IF1

$$\frac{\langle e, m\rangle \Downarrow v \qquad v \neq 0}{\langle \mathsf{if}\ e\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2, m\rangle \to \langle c_1, m\rangle}$$

S-IF2

$$\frac{\langle e, m\rangle \Downarrow v \qquad v = 0}{\langle \mathsf{if}\ e\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2, m\rangle \to \langle c_2, m\rangle}$$

S-WHILE

$$\frac{}{\langle \mathsf{while}\ e\ \mathsf{do}\ c, m\rangle \to \langle \mathsf{if}\ e\ \mathsf{then}\ c; \mathsf{while}\ e\ \mathsf{do}\ c\ \mathsf{else}\ \mathsf{skip}, m\rangle}$$

Q: how many steps does it take to evaluate the example program `while x>0 do x:=x-1` starting from memory *m* such that *m*(x)=5

Building an intuition towards a security policy through examples

```
x_p := y_s
```

```
x_p := 42
```

```
y_s := 42; x_p := y_s
```

Building an intuition towards a security policy through examples

```
if y_s > 0 then x_p := 1 else x_p := 0
```

```
if y_s > 0 then x_p := 0 else x_p := 0
```

```
while y_s > 0 do skip
```

```
x_p:= 1; (while y_s > 0 do skip) x_p := 2
```

Security definition

- Threat model
  - Secrets are in the initial memory
  - Attacker
    - knows the source code
    - knows the initial public slice of the memory
  - Observations ~ events
    - simplification: events are assignments during the execution
    - semantics “decorated” with events and traces
- Knowledge-based approach
  - “What does attacker know” based on their observations
  - Set-based approach
- Main intuition: *what is attacker allowed to know?*

Enforcement via a type system

Type system

variable security type environment: mapping from variable names to levels

Typing for expressions: $\Gamma \vdash e : \ell.$

$$\text{T-INT}\quad \frac{}{\Gamma \vdash n : \ell}$$

$$\text{T-VAR}\quad \frac{\Gamma(x) = \ell}{\Gamma \vdash x : \ell}$$

$$\text{T-OP}\quad \frac{\Gamma \vdash e_i : \ell_i, (i = 1, 2)}{\Gamma \vdash e_1 \ op \ e_2 : \ell_1 \sqcup \ell_2}$$

$\ell_1 \sqcup \ell_2$ is public if both ℓ_1 and ℓ_2 are public; secret otherwise

Example: in environment Γ, where Γ(x) = public and Γ(y) = secret, we have:

Γ ⊢ 42 : public Γ ⊢ x : public Γ ⊢ x + 1 : public

Γ ⊢ y +1 : secret Γ ⊢x + y : secret Γ ⊢ y - y : secret

Type system

program counter label

Typing for expressions: $\Gamma, pc \vdash c$

This prevents updates to public variables in secret-dependent branches

$\ell_1 \sqsubseteq \ell_2$ when ℓ_1 is public OR ℓ_2 is secret

T-SKIP

$$\frac{}{\Gamma, pc \vdash \text{skip}}$$

T-ASSIGN

$$\frac{\Gamma \vdash e : \ell \qquad pc \sqcup \ell \sqsubseteq \Gamma(x)}{\Gamma, pc \vdash x := e}$$

T-SEQ

$$\frac{\Gamma, pc \vdash c_i, (i = 1, 2)}{\Gamma, pc \vdash c_1; c_2}$$

T-IF

$$\frac{\Gamma \vdash e : \ell \qquad \Gamma, pc \sqcup \ell \vdash c_i, (i = 1, 2)}{\Gamma, pc \vdash \text{if } e \text{ then } c_1 \text{ else } c_2}$$

raises the pc-label for the branches

T-WHILE

$$\frac{\Gamma \vdash e : \ell \qquad \Gamma, pc \sqcup \ell \vdash c}{\Gamma, pc \vdash \text{while } e \text{ do } c}$$

Example derivation tree

$$\dfrac{\dfrac{\Gamma \vdash 0 : \mathsf{Public} \qquad \mathsf{Public} \sqsubseteq \Gamma(\mathtt{x})}{\Gamma, \mathsf{Public} \vdash \mathtt{x\_p} := 0} \qquad \dfrac{\dfrac{\Gamma(\mathtt{y\_s}) = \mathsf{Secret}}{\Gamma \vdash \mathtt{y\_s} : \mathsf{Secret}} \qquad \dfrac{\dfrac{\Gamma(\mathtt{y\_s}) = \mathsf{Secret}}{\Gamma \vdash \mathtt{y\_s} : \mathsf{Secret}} \qquad \Gamma \vdash 1 : \mathsf{Public}}{\Gamma \vdash \mathtt{y\_s} - 1 : \mathsf{Secret}} \qquad \mathsf{Secret} \sqsubseteq \Gamma(\mathtt{y\_s})}{\Gamma, \mathsf{Secret} \vdash \mathtt{y\_s} := \mathtt{y\_s} - 1} \qquad \dfrac{\Gamma \vdash 1 : \mathsf{Public} \qquad \mathsf{Public} \sqsubseteq \Gamma(\mathtt{x})}{\Gamma, \mathsf{Public} \vdash \mathtt{x\_p} := 1}}{\dfrac{\Gamma, \mathsf{Public} \vdash \mathtt{while\ y\_s\ do\ y\_s} := \mathtt{y\_s} - 1}{\Gamma, \mathsf{Public} \vdash \mathtt{while\ y\_s\ do\ y\_s} := \mathtt{y\_s} - 1;\ \mathtt{x\_p} := 1}}}{\Gamma, \mathsf{Public} \vdash \mathtt{x\_p} := 0;\ \mathtt{while\ y\_s\ do\ y\_s} := \mathtt{y\_s} - 1;\ \mathtt{x\_p} := 1}$$

[extra]

Proof sketch

$\langle c, m\rangle \rightarrow^* \langle stop, m'\rangle$ means there

are intermediate configurations $\langle c_1, m_1\rangle$, $\langle c_2, m_2\rangle$ … such that

$\langle c, m\rangle \rightarrow \langle c_1, m_1\rangle \rightarrow \langle c_2, m_2\rangle \rightarrow \ldots \rightarrow \langle c_n, m_n\rangle \rightarrow \langle stop, m'\rangle$

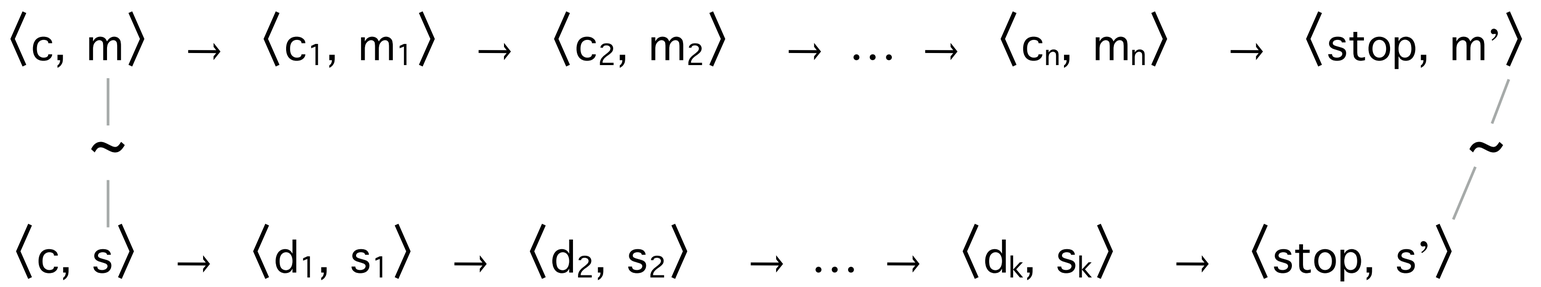
Similarly, for $\langle c, s\rangle \rightarrow^* \langle stop, s'\rangle$ we have

$\langle c, s\rangle \rightarrow \langle d_1, s_1\rangle \rightarrow \langle d_2, s_2\rangle \rightarrow \ldots \rightarrow \langle d_k, s_k\rangle \rightarrow \langle stop, s'\rangle$

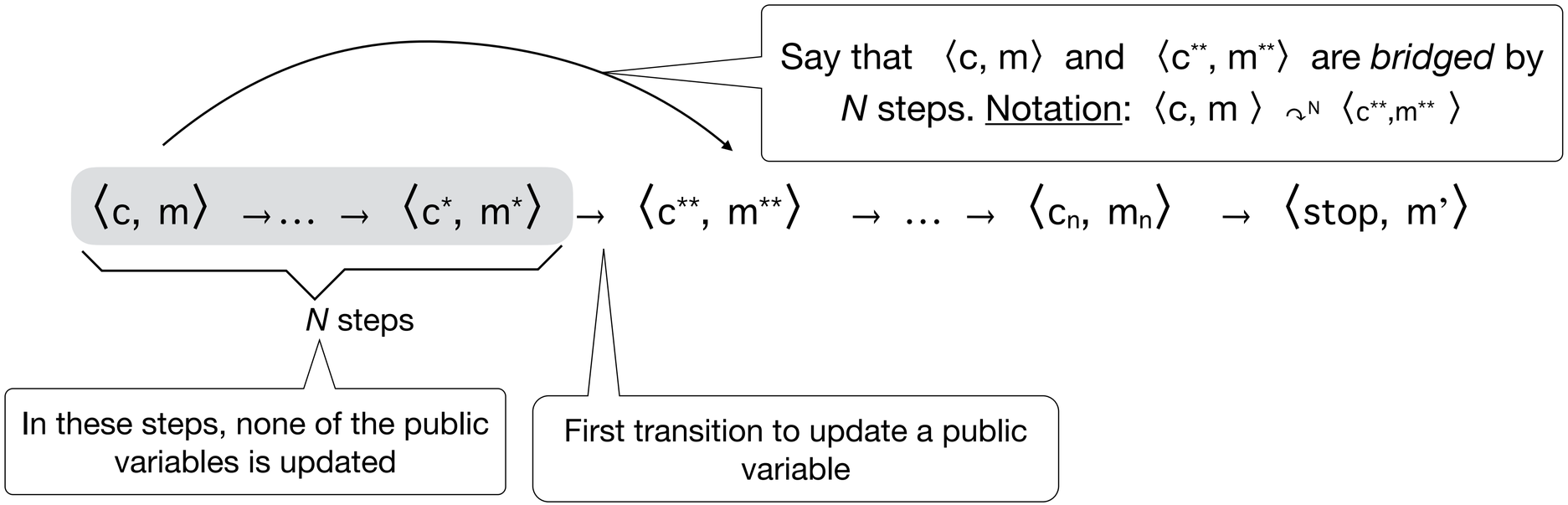
(note the potentially different number of steps and the different intermediate configurations)

What we need to prove

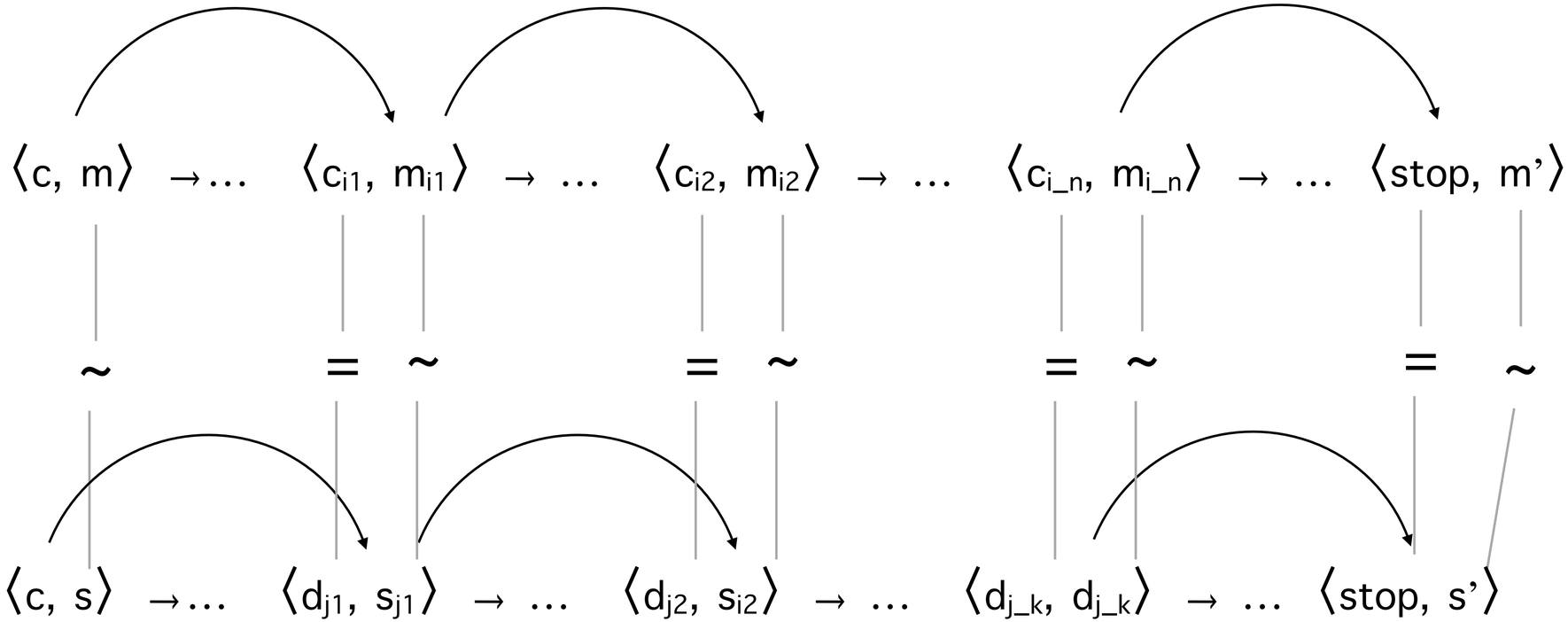
Assuming that Γ, pc $\vdash$ c, and m~s, it must be that m' ~ s'



Observation: if m and m' disagree on public variables, then m' must have been changed at some intermediate steps



Intuition for the proof

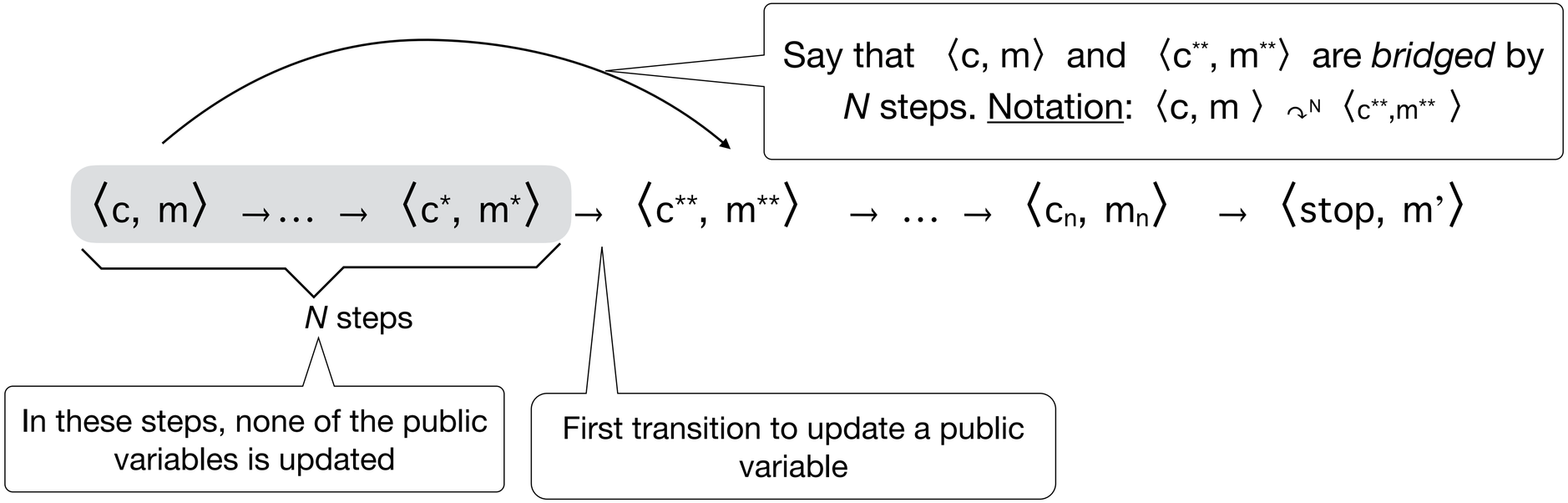


Proof: “triple-nested” induction

- by induction on the number of bridges in the first run
  - by induction on the number of intermediate steps within each bridge
    - by induction on the structure of the commands making the steps

need to formalize the bridge relation

Lemma: Bridge NI (coming later)



Bridge as an *inductive* relation

$$\frac{\langle c,\ m\rangle \ \rightarrow\ \langle stop,m'\rangle \quad m\sim m'}{\langle c,\ m\rangle \ \curvearrowright^{0}\ \langle stop,m'\rangle}\ \text{Bridge-Stop}$$

some public variable is updated

$$\frac{\langle c,\ m\rangle \ \rightarrow\ \langle c',m'\rangle \quad m\not\sim m'}{\langle c,\ m\rangle \ \curvearrowright^{0}\ \langle c',m'\rangle}\ \text{Bridge-Public}$$

none of public vars is updated

$$\frac{\langle c,\ m\rangle \ \rightarrow\ \langle c',m'\rangle \quad m\sim m' \quad c'\neq stop \quad \langle c',\ m'\rangle \ \curvearrowright^{N}\ \langle c'',m''\rangle}{\langle c,\ m\rangle \ \curvearrowright^{N+1}\ \langle c'',m''\rangle}\ \text{Bridge-Multi}$$

In this definition, Bridge-Stop and Bridge-Public are the two base cases (N=0), and Bridge-Multi is the inductive clause

Main lemma: bridge noninterference

Given Γ, pc, c, m, s, N, N' such that

Γ, pc $\vdash$ c and

m $\sim$ s and

$\langle c, m \rangle \curvearrowright^{N} \langle c', m' \rangle$ and

$\langle c, s \rangle \curvearrowright^{N'} \langle d', s' \rangle$

Then: m' $\sim$ s' and c' = d'

Now that bridge is formally defined, this actually makes sense

Proof: “two-dimensional” induction. Outer layer: by induction on N; inner layer by induction on the structure of c

We also need a bunch of auxiliary lemmas

Auxiliary facts we need (1/3)

Lemma: noninterference for expressions:

Given Γ, e, m, s, ℓ such that m~s and $\Gamma \vdash$ e : ℓ and

$\langle$ e, m $\rangle \Downarrow$ v and $\langle$ e, s $\rangle \Downarrow$ u

Then: ℓ = Public $\Rightarrow$ v = u

Proof: by induction on the *structure* of e

Expression e can be one of n | x | e_1 op e_2. Consider them all

Base cases:

- e is n: We have $\Gamma \vdash$ n : Public. But also, both memories evaluate to the same value n, because $\langle$n, m $\rangle \Downarrow$ n and $\langle$n, s $\rangle \Downarrow$ n
- e is x: We have $\Gamma \vdash$ x : ℓ, where Γ(x) = ℓ. We have two cases
  - ℓ = Secret. In this case we're done
  - ℓ = Public. We have $\langle$x, m $\rangle \Downarrow$ v and $\langle$x, s $\rangle \Downarrow$ u, where v = m(x) and u = s(x). Because m ~ s, and Γ(x) = Public, we get that v = u

Inductive case: e is e_1 op e_2

Induction hypothesis: assume that the lemma holds for all e' that are structurally smaller than e (this includes e_1 and e_2)

IH: Given Γ, e' *that is structurally smaller than e*, m, s, ℓ such that m~s and Γ ⊢ e' : ℓ and ⟨e', m⟩ ⇓ v and ⟨e', s⟩ ⇓ u. Then: ℓ = Public ⇒ v = u

Inductive case: e is e_1 op e_2

We have Γ ⊢ e_1 op e_2 : ℓ

If ℓ = Secret, we're done

If ℓ = Public then recall the typing rule for operations

$$\frac{\Gamma \vdash e_i : \ell_i, (i = 1, 2)}{\Gamma \vdash e_1 \ op \ e_2 : \ell_1 \sqcup \ell_2} \text{T-OP}$$

$\ell_1 \sqcup \ell_2$ is public if both ℓ_1 and ℓ_2 are public; secret otherwise

Since ℓ = Public, it means that both ℓ_1 and ℓ_2 must be public, too. Now, recall the rules for evaluating e_1 op e_2

$$\frac{\langle e_1, m\rangle \Downarrow v_1 \quad \langle e_2, m\rangle \Downarrow v_2 \quad v = v_1 \ \textbf{op} \ v_2}{\langle e_1 \ op \ e_2, m\rangle \Downarrow v}$$

$$\frac{\langle e_1, s\rangle \Downarrow u_1 \quad \langle e_2, s\rangle \Downarrow u_2 \quad u = u_1 \ \textbf{op} \ u_2}{\langle e_1 \ op \ e_2, s\rangle \Downarrow u}$$

We can apply the IH to e_1 because
- e_1 is structurally smaller than e
- m ~ s
- Γ ⊢ e1 : ℓ_1

From the IH and because ℓ_1 = Public, we get that $v_1 = u_1$.

Similarly, by IH for e_2, we get $v_2 = u_2$

Assuming that **op** is deterministic, from $v_1 = u_1$ and $v_2 = u_2$ we get that v = u

Auxiliary facts we need (2/3)

Lemma: preservation of types by the step relation:

Given Γ, pc, c, m such that

$$\Gamma, \text{pc} \vdash c \text{ and } \langle c, m \rangle \rightarrow \langle c', m' \rangle$$

Then: Γ, pc $\vdash$ c' or c' = stop

This clause is needed because we have no typing rule for *stop*

Proof: by induction on the structure of c

(details omitted, try yourself at home)

Auxiliary facts we need (3/3)

Lemma: Secret PC

Given Γ, c, m such that Γ, Secret $\vdash$ c and $\langle c, m \rangle \rightarrow \langle c', m' \rangle$

Then: $m \sim m'$

Proof: by induction on the structure of c

(details omitted, try yourself at home)

Corollary: Secret PC*

Given Γ, c, m such that Γ, Secret $\vdash$ c and $\langle c, m \rangle \rightarrow^* \langle c', m' \rangle$

Then: $m \sim m'$

Proof: Using lemma above + preservation

Main lemma: bridge noninterference

Given Γ, pc, c, m, s, N, N' such that $\Gamma, pc \vdash c$ and $m \sim s$ and

$\langle c, m\rangle \curvearrowright^{N} \langle c', m'\rangle$ and $\langle c, s\rangle \curvearrowright^{N'} \langle d', s'\rangle$ Then: $m' \sim s'$ and $c' = d'$

Proof: "two-dimensional" induction. Outer layer: by induction on N; inner layer by induction on the structure of c

Outer base case N = 0.

Inner induction: on the structure of c.

$$c ::= \text{skip} \mid x := e \mid c; c \mid \sout{\text{if } e \text{ then } e \text{ else } e} \mid \sout{\text{while } e \text{ do } e}$$

Impossible with N=0

Base cases

Inductive case

Case c is skip; consider the possibilities for the m-run

$$\frac{\langle \text{skip}, m\rangle \to \langle \text{stop}, m\rangle \quad m \sim m}{\langle \text{skip}, m\rangle \curvearrowright^{0} \langle \text{stop}, m\rangle}$$ Bridge-Stop

Not applicable

$$\frac{\langle \text{skip}, m\rangle \to \langle \text{stop}, m\rangle \quad m \not\sim m}{\langle \text{skip}, m\rangle \curvearrowright^{0} \langle \text{stop}, m\rangle}$$ Bridge-Public

For the s-run, because c=skip, the only possibility is that N'=0, also only via [Bridge-Stop], and then we can conclude that $\langle \text{skip}, s\rangle \curvearrowright^{0} \langle \text{stop}, s\rangle$

Main lemma: bridge noninterference

Given Γ, pc, c, m, s, N, N' such that Γ, pc $\vdash$ c and m $\sim$ s and

$\langle c, m\rangle \rightsquigarrow^{N} \langle c', m'\rangle$ and $\langle c, s\rangle \rightsquigarrow^{N'} \langle d', s'\rangle$ Then: m' $\sim$ s' and c' = d'

Outer base case N = 0. Inner induction: on the structure of c.

Case c is x := e. There are two possibilities

$$\frac{\langle x{:=}e,\ m\rangle \rightarrow \langle stop, m'\rangle \quad m \sim m'}{\langle x{:=}e,\ m\rangle \rightsquigarrow^{0} \langle stop, m'\rangle}\ \text{Bridge-Stop} \qquad \frac{\langle x{:=}e,\ m\rangle \rightarrow \langle stop, m'\rangle \quad m \nsim m'}{\langle x{:=}e,\ m\rangle \rightsquigarrow^{0} \langle stop, m'\rangle}\ \text{Bridge-Public}$$

In either case, it must be that $\langle x := e, m\rangle \rightarrow \langle stop, m[x \mapsto v]\rangle$ s.t. $\langle e, m\rangle \Downarrow v$

For the s-run, because c is x:=e, the only possibility is that N'=0, and it must also be that $\langle x{:=}e, s\rangle \rightarrow \langle stop, s'[x \mapsto u]\rangle$, s.t. $\langle e, s\rangle \Downarrow u$

1) Consider [Bridge-Public] first. From m$\nsim$m' it must be that Γ(x) = public.

Recall the typing rule for assignments: $$\frac{\Gamma \vdash e : \ell \qquad pc \sqcup \ell \sqsubseteq \Gamma(x)}{\Gamma, pc \vdash x := e}$$

$\ell_1 \sqsubseteq \ell_2$ when ℓ_1 is public OR ℓ_2 is secret

Γ(x) = public means that pc = public AND ℓ is public

From NI for expressions we get that u = v; hence m' $\sim$ s'

2) Consider [Bridge-Stop]. Two subcases:
   (a) Γ(x) = Public, but it must be that m(x) = m'(x) . Proceed as in case 1 above
   (b) Γ(x) = Secret. Immediate

Main lemma: bridge noninterference

Given Γ, pc, c, m, s, N, N' such that $\Gamma, pc \vdash c$ and $m \sim s$ and

$\langle c, m\rangle \rightsquigarrow^{N} \langle c',m'\rangle$ and $\langle c, s\rangle \rightsquigarrow^{N'} \langle d',s'\rangle$ Then: $m' \sim s'$ and $c' = d'$

Outer base case N = 0. Inner induction: on the structure of c.

Case c is $c_1;c_2$.

In the m-run, because N=0, the only possibility is

$$\frac{\langle c_1;c_2,\ m\rangle \rightarrow \langle c',m'\rangle \qquad m \nsim m'}{\langle c_1;c_2,\ m\rangle \rightsquigarrow^{0} \langle c',m'\rangle}\ \text{Bridge-Public}$$

Recall the semantics for sequential composition. Two possibilities are

$$\frac{\langle c_1,\ m\rangle \rightarrow \langle c'_1,m'\rangle \qquad c'_1 \neq \text{stop}}{\langle c_1;c_2,\ m\rangle \rightarrow \langle c'_1;c_2',m'\rangle}$$

$$\frac{\langle c_1,\ m\rangle \rightarrow \langle c'_1,r}{\langle c_1;c_2,\ m\rangle \rightarrow \langle c_2',m'\rangle}$$

Done by IH

In either case, it is c1 that performs the public update, so we write

$\langle c_1,\ m\rangle \rightarrow \langle c'_1,m'\rangle$, where c'_1 could be stop

In the s-run, we know less, so more is possible

$$\frac{\langle c_1;c_2,\ s\rangle \rightarrow \langle c',s'\rangle \qquad s \nsim s'}{\langle c_1;c_2,\ s\rangle \rightsquigarrow^{0} \langle d',s'\rangle}\ \text{Bridge-Public}$$

In this case, it is c1 that performs the public update

$\langle c_1,\ s\rangle \rightarrow \langle d'_1,s'\rangle$, where d'_1 could be stop

Alternatively:

$$\frac{\langle c_1;c_2,\ s\rangle \rightarrow \langle d^*,s^*\rangle \qquad s \sim s^* \qquad d^* \neq \text{stop} \qquad \langle d^*,\ s^*\rangle \rightsquigarrow^{N'-1} \langle d',s'\rangle}{\langle c_1;c_2,\ s\rangle \rightsquigarrow^{N'} \langle d',s'\rangle}\ \text{Bridge-Multi}$$

This means, that c_1 does not perform the public update

$\langle c_1,\ s\rangle \rightarrow \langle d^*_1,s^*\rangle$, where d'_1 could be stop

Impossible by IH, because $m' \sim s^* \sim s \sim m \nsim m'$ is a contradiction

Main lemma: bridge noninterference

Given Γ, pc, c, m, s, N, N' such that $\Gamma, pc \vdash c$ and $m \sim s$ and

$\langle c, m \rangle \curvearrowright^{N} \langle c', m' \rangle$ and $\langle c, s \rangle \curvearrowright^{N'} \langle d', s' \rangle$ Then: $m' \sim s'$ and $c' = d'$

Proof: "two-dimensional" induction. Outer layer: by induction on N; inner layer by induction on the structure of c

Outer inductive case N > 0.

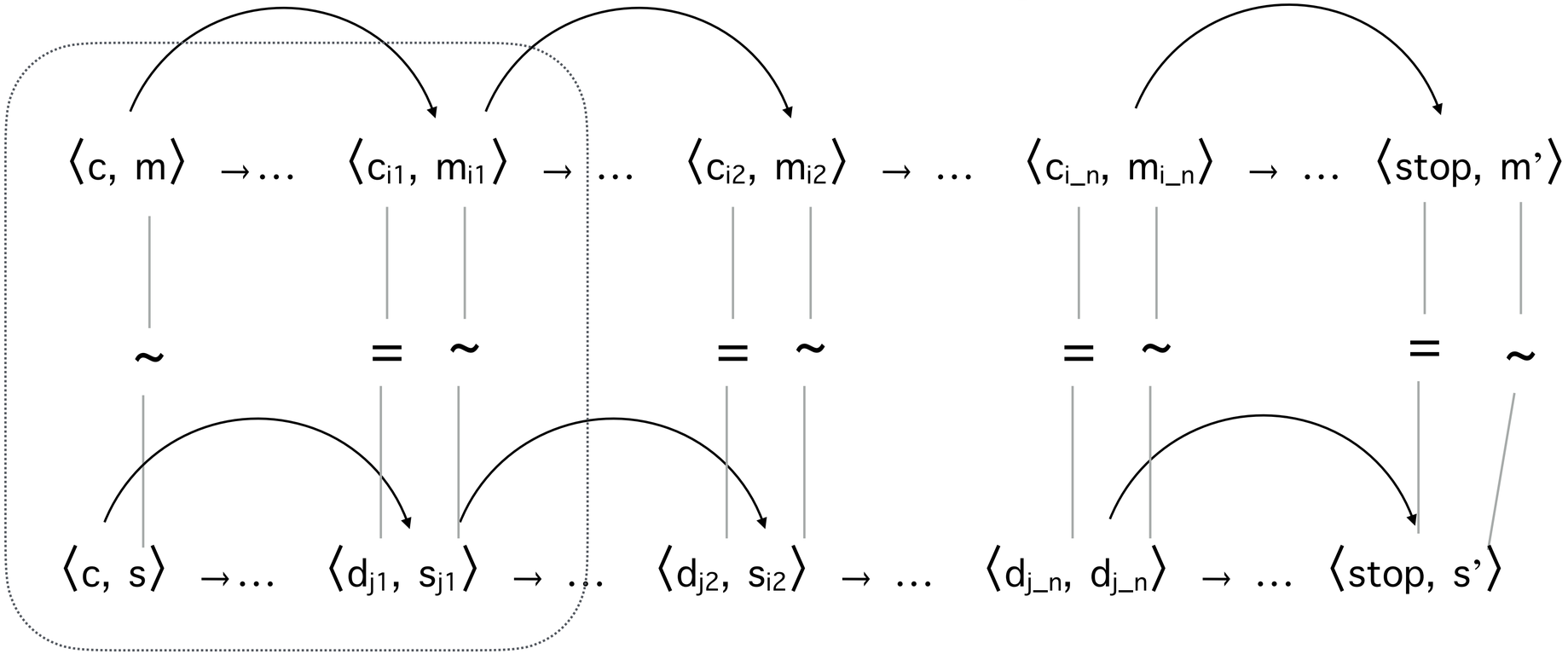
Inner induction: on the structure of c.

$$c ::= \sout{\text{skip} \mid x := e} \mid c; c \mid \text{if } e \text{ then } c \text{ else } c \mid \text{while } e \text{ do } c$$

Impossible with N > 0

- Case $c_1;c_2$ – similar to the one we considered for N=0; but here we have 4 cases instead of 2 to apply the IH for c1. We also now have the possibility of when c1 does not update public var; there we also apply the IH to c2.
- Case if e then c_1 else c_2. Two cases depending on the level of pc $\sqcup \ell$, where $\Gamma \vdash e: \ell$
  - If pc $\sqcup \ell$ = public, then both runs take the same branch; Done by either of the IH
  - If pc $\sqcup \ell$ = secret, then by Lemma Secret PC*, public memories are not updated in either of the runs
- Case while e do c. Done by the outer IH

Intuition for the proof



Proof: “triple-nested” induction

- by induction on the number of bridges in the first run
  - by induction on the number of intermediate steps within each bridge
    - by induction on the structure of the commands making the steps

Lemma: Bridge NI