Calculational Design of Information Flow Monitors

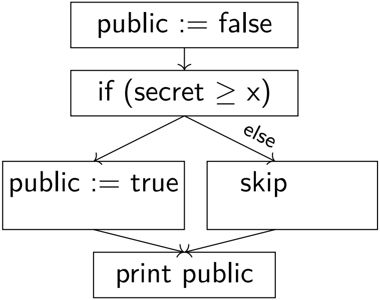
Mounir Assaf David Naumann

Stevens Institute of Technology, Hoboken, NJ

November 9th, 2016

SoSySec Seminar, Rennes

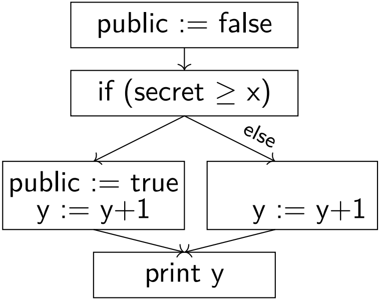
- Information security :
  - Confidentiality
  - Integrity
  - Availability
- Traditionally, dissemination of information is prevented through Access control :
  - Deals with what piece of information can be accessed? by whom?
  - Yet, is this piece of information handled correctly when accessed?
- Information Flow Control :
  - Tracks how information is propagated through a program
  - Verifies that **information flows** are secure with respect to a security policy



Implicit flows

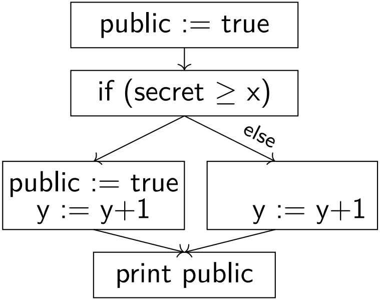
- from conditional expressions to variables assigned inside conditionals

- An implicit flow from variable `secret` to variable `public`
  - Detected by [Volpano et al.,96], [Amtoft & Banerjee,04], [Hunt & Sands,06], [Jif], [Flow Caml], [Le Guernic et al.,06] ...



- No information flow from variable `secret` to variable `y`
  - Static analysis by abstract interpretation of self-composed programs [Kovács et al.,13], [Müller et al.,15]

Pervasive Flows

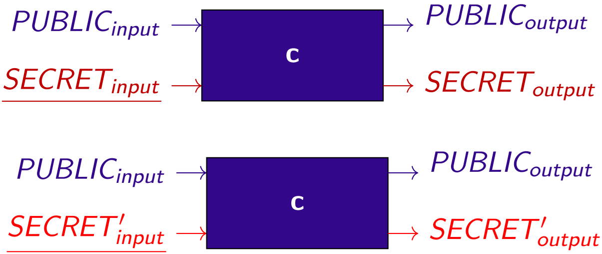


- No information flow from variable `secret` to variable `y`
- No information flow from variable `secret` to variable `public`
  - Hybrid monitoring by relying on complex static analyses of non-executed branches [Besson et al.,13], [Besson et al.,16]

- How do we account for information leaks, without rejecting too many secure programs?
    - More reasoning on program semantics?
      Abstract interpretation!

- This talk : Monitoring information flow as calculational abstract interpretation
  Follow up on monitoring as abstract interpretation [Chudnov et al.,14]
    - Systematic design and derivation of information flow monitors
    - Leveraging a large body of the literature in abstract interpretation, since seminal papers [Cousots,77 & 79]
    - Semantic characterization of information flow monitors as a starting point

- **Termination-Insensitive Non-Interference:**

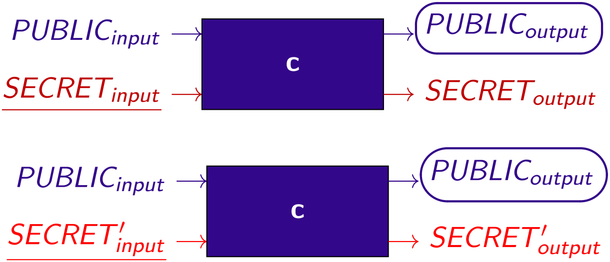


- **A pre/post relational logic formulation:** [Benton,04]
  For any two input memories, agreement over "Public inputs" leads to agreement over "Public outputs" for the output memories

$$\textbf{assume } \mathbb{A} Public_{input} ;\ \textbf{c} \ ;\ \textbf{assert } \mathbb{A} Public_{output}$$

Ideal Monitor

- **Termination-Insensitive Non-Interference:**



- **A pre/post relational logic formulation:** [Benton,04]
  For any two input memories, agreement over "Public inputs" leads to agreement over "Public outputs" for the output memories

$$\textbf{assume } \mathbb{A}Public_{input};\ \mathbf{c}\ ;\ \textbf{assert } \mathbb{A}Public_{output}$$

assume $\mathbb{A}$public;

c;

assert $\mathbb{A}$outputs

Major execution

$public^1 secret^{10}$

assume $\mathbb{A}$public;

c;

assert $\mathbb{A}$outputs

Major execution ***Minor executions***

$public^1 \, secret^{10}$

$public^1 \, secret^{20}$ $public^1 \, secret^{30}$

$public^0 \, secret^{11}$ $public^3 \, secret^{10}$ ...

assume $\mathbb{A}$public;

c;

assert $\mathbb{A}$outputs

Major execution

Minor executions

$public^1 secret^{10}$

$public^1 secret^{20}$ $public^1 secret^{30}$

$public^0 secret^{11}$ $public^3 secret^{10}$ . . .

assume $\mathbb{A}$public;

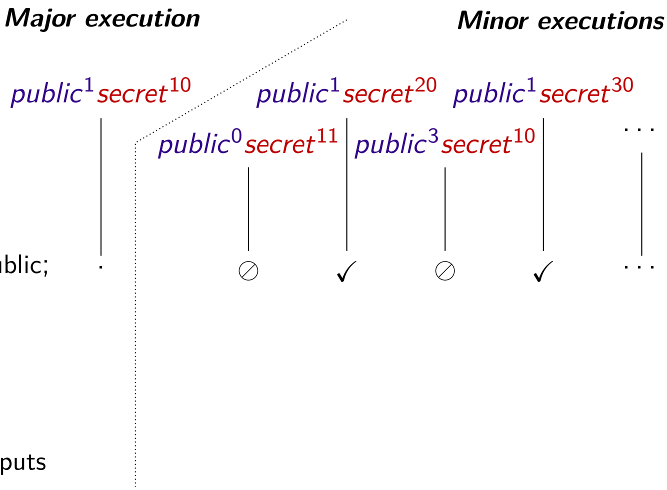
c;

assert $\mathbb{A}$outputs

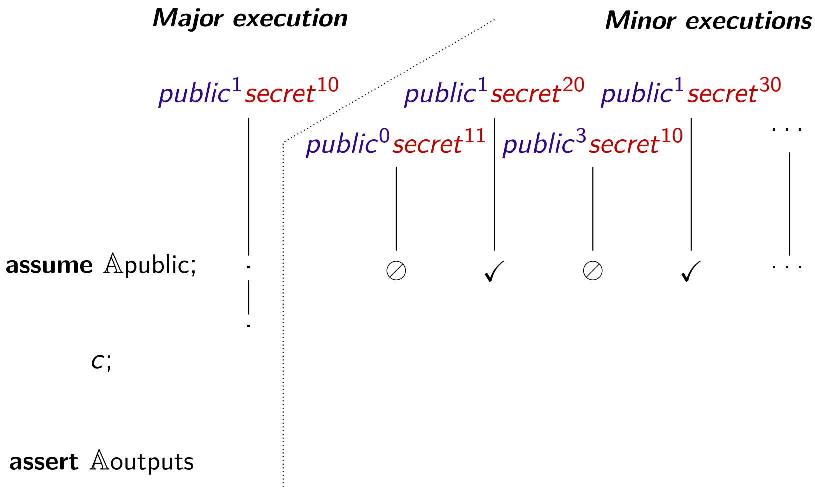
assume $\mathbb{A}$public;

c;

assert $\mathbb{A}$outputs



Ideal Monitor



Ideal Monitor

Major execution ***Minor executions***

$public^1 \, secret^{10}$ $public^1 \, secret^{20}$ $public^1 \, secret^{30}$

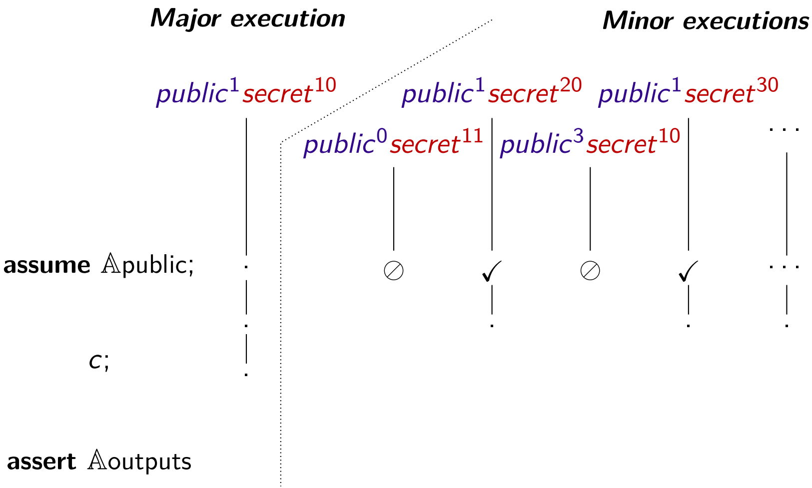
$public^0 \, secret^{11}$ $public^3 \, secret^{10}$ …

assume $\mathbb{A}$public; $\oslash$ ✓ $\oslash$ ✓ …

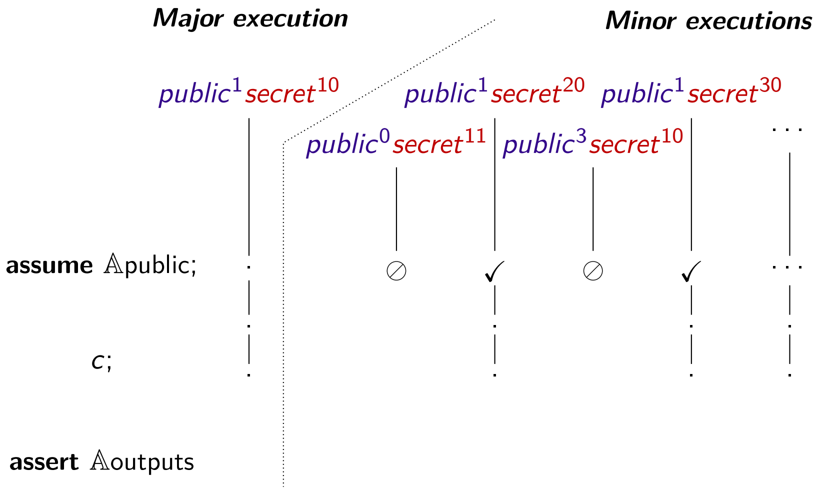
c;

assert $\mathbb{A}$outputs

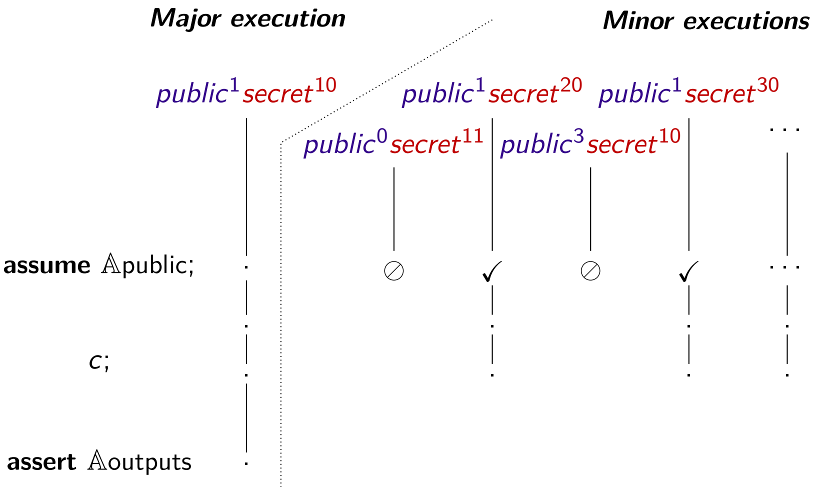
Ideal Monitor

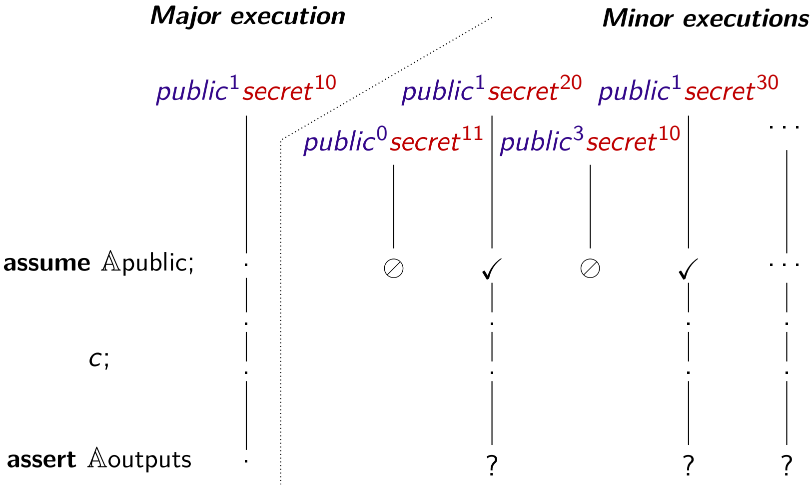


Ideal Monitor



Ideal Monitor





- Monitor either returns result of major execution, or withholds result if **assert** command fails ↯

- **Ideal monitor:** a simultaneous execution of the program on a major state, and a ***tracking set*** (all – relevant – minor states)

$$\llbracket c \rrbracket \in \text{States} \to \text{States}, \qquad (\!| c |\!)_\sigma \in \mathcal{P}(\text{States}) \to \mathcal{P}(\text{States})$$

$$\begin{tikzcd}
& \sigma \arrow[rr, dashed, "\llbracket c \rrbracket"] & & \sigma' \\
\mathcal{P}(\text{States}) \arrow[ur, dash, dashed] \arrow[rr, "(\!| c |\!)_\sigma"'] & & \mathcal{P}(\text{States}) \arrow[ur, dash, dashed] &
\end{tikzcd}$$

- Pick an abstraction $\mathscr{A}$
- Give abstract objects $a \in \mathscr{A}$ a meaning by linking them to concrete objects $c \in \mathscr{C}$ through a Galois connection: $(\mathscr{C}, \subseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\mathscr{A}, \sqsubseteq^\sharp)$
- Best approximation of a transformer $f \in \mathscr{C} \to \mathscr{C}$ given by : $\alpha \circ f \circ \gamma \in \mathscr{A} \to \mathscr{A}$

- Pick an abstraction $\mathscr{A}$
- Give abstract objects $a \in \mathscr{A}$ a meaning by linking them to concrete objects $c \in \mathscr{C}$ through a Galois connection: $(\mathscr{C}, \subseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\mathscr{A}, \sqsubseteq^\sharp)$
- Best approximation of a transformer $f \in \mathscr{C} \to \mathscr{C}$ given by : $\alpha \circ f \circ \gamma \in \mathscr{A} \to \mathscr{A}$

Let $\mathscr{C}$ to be a powerset of reachable states, and $\mathscr{A}$ be a powerset of parity predicates on variables:

$$\gamma(\text{Even y})$$

$$= \quad \{\sigma \in \text{States} \mid \sigma(y) \text{ is even}\}$$

- Pick an abstraction $\mathscr{A}$
- Give abstract objects $a \in \mathscr{A}$ a meaning by linking them to concrete objects $c \in \mathscr{C}$ through a Galois connection: $(\mathscr{C}, \subseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\mathscr{A}, \sqsubseteq^\sharp)$
- Best approximation of a transformer $f \in \mathscr{C} \to \mathscr{C}$ given by : $\alpha \circ f \circ \gamma \in \mathscr{A} \to \mathscr{A}$

Let $\mathscr{C}$ to be a powerset of reachable states, and $\mathscr{A}$ be a powerset of parity predicates on variables:

$$\alpha \circ \{\!|\text{x:=y+1}|\!\} \circ \gamma(\text{Even y})$$

- Pick an abstraction $\mathscr{A}$
- Give abstract objects $a \in \mathscr{A}$ a meaning by linking them to concrete objects $c \in \mathscr{C}$ through a Galois connection: $(\mathscr{C}, \subseteq) \xleftrightarrow[\alpha]{\gamma} (\mathscr{A}, \sqsubseteq^\sharp)$
- Best approximation of a transformer $f \in \mathscr{C} \to \mathscr{C}$ given by : $\alpha \circ f \circ \gamma \in \mathscr{A} \to \mathscr{A}$

Let $\mathscr{C}$ to be a powerset of reachable states, and $\mathscr{A}$ be a powerset of parity predicates on variables:

$$\alpha \circ \{\!|\text{x:=y+1}|\!\} \circ \gamma(\text{Even y})$$
$$= \alpha \circ \{\!|\text{x:=y+1}|\!\}\big(\ \{\sigma \in \text{States} \mid \sigma(y) \text{ is even}\}\ \big)$$

- Pick an abstraction $\mathscr{A}$
- Give abstract objects $a \in \mathscr{A}$ a meaning by linking them to concrete objects $c \in \mathscr{C}$ through a Galois connection: $(\mathscr{C}, \subseteq) \xleftarrow[\alpha]{\gamma} \xrightarrow{} (\mathscr{A}, \sqsubseteq^\sharp)$
- Best approximation of a transformer $f \in \mathscr{C} \to \mathscr{C}$ given by : $\alpha \circ f \circ \gamma \in \mathscr{A} \to \mathscr{A}$

Let $\mathscr{C}$ to be a powerset of reachable states, and $\mathscr{A}$ be a powerset of parity predicates on variables:

$$\alpha \circ \{\!\!|\text{x:=y+1}|\!\!\} \circ \gamma(\text{Even y})$$
$$= \alpha \circ \{\!\!|\text{x:=y+1}|\!\!\}(\ \{\sigma \in \text{States} \mid \sigma(y) \text{ is even}\}\ )$$
$$= \alpha(\ \{\sigma[x \mapsto \sigma(y) + 1] \in \text{States} \mid \sigma(y) \text{ is even}\}\ )$$

- Pick an abstraction $\mathscr{A}$
- Give abstract objects $a \in \mathscr{A}$ a meaning by linking them to concrete objects $c \in \mathscr{C}$ through a Galois connection: $(\mathscr{C}, \subseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\mathscr{A}, \sqsubseteq^\sharp)$
- Best approximation of a transformer $f \in \mathscr{C} \to \mathscr{C}$ given by : $\alpha \circ f \circ \gamma \in \mathscr{A} \to \mathscr{A}$

Let $\mathscr{C}$ to be a powerset of reachable states, and $\mathscr{A}$ be a powerset of parity predicates on variables:

$$
\begin{aligned}
&\alpha \circ \{\!|\mathsf{x{:=}y{+}1}|\!\} \circ \gamma(\mathsf{Even\ y}) \\
&\quad = \alpha \circ \{\!|\mathsf{x{:=}y{+}1}|\!\}(\ \{\sigma \in \text{States} \mid \sigma(y) \text{ is even}\}\ ) \\
&\quad = \alpha(\ \{\sigma[x \mapsto \sigma(y) + 1] \in \text{States} \mid \sigma(y) \text{ is even}\}\ ) \\
&\quad = \{\mathit{Even}\ y, \mathit{Odd}\ x\}
\end{aligned}
$$

- Assuming a Galois connection: $(\mathcal{P}(\text{States}), \subseteq) \xleftarrow[\alpha]{\gamma}\!\!\!\!\!\!\!\!\xrightarrow{} (\mathscr{A}, \sqsubseteq^\sharp)$
- Best approximation of static collecting semantics $\{\!|c|\!\} \in \mathcal{P}(\text{States}) \to \mathcal{P}(\text{States})$ given by : $\alpha \circ \{\!|c|\!\} \circ \gamma$

$$\begin{tikzcd}
\mathscr{A} \arrow[r, "\alpha\circ\{\!|c|\!\}\circ\gamma"] \arrow[d, "\gamma"'] & \mathscr{A} \\
\mathcal{P}(\text{States}) \arrow[r, "\{\!|c|\!\}"'] & \mathcal{P}(\text{States}) \arrow[u, "\alpha"']
\end{tikzcd}$$

- Assuming a Galois connection: $(\mathcal{P}(\text{States}), \subseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\mathscr{A}, \sqsubseteq^\sharp)$
- Best approximation of static collecting semantics $\{\!|c|\!\} \in \mathcal{P}(\text{States}) \to \mathcal{P}(\text{States})$ given by : $\alpha \circ \{\!|c|\!\} \circ \gamma$

$$
\begin{tikzcd}
\mathscr{A} \arrow[rr, bend left=50, "\{\!|c|\!\}^\sharp"] \arrow[rr, "\alpha\circ\{\!|c|\!\}\circ\gamma"] \arrow[rr, phantom, shift left=3ex, "\dot{\sqsubseteq}^\sharp"] \arrow[d, "\gamma"'] & & \mathscr{A} \\
\mathcal{P}(\text{States}) \arrow[r, "\{\!|c|\!\}"'] & \mathcal{P}(\text{States}) \arrow[ur, "\alpha"] &
\end{tikzcd}
$$

- Pick an abstraction: relational formulas
- Define a Galois connection interpreting relational formula: Monitoring wrt. a major state means that the abstraction should be interpreted wrt. a major state $\sigma \in \text{States}$:

$$\gamma_\sigma(\mathbb{A}x) \triangleq \{\tau \in \text{States} \mid \tau(x) = \sigma(x)\}$$

$$\alpha_\sigma(\Sigma) \triangleq \{\Phi \mid \forall \tau \in \Sigma, \tau \mid \sigma \models \Phi\}$$

$$\begin{tikzcd}
& \sigma \arrow[rr, dashed, "\llbracket c \rrbracket"] \arrow[dl, dash, dashed] & & \sigma' \arrow[dl, dash, dashed] \\
\mathcal{P}(\text{States}) \arrow[rr, "(\!| c |\!)_\sigma"'] & & \mathcal{P}(\text{States}) &
\end{tikzcd}$$

Monitoring as Calculational Abstract Interpretation

$$
\begin{tikzcd}
 & \sigma \arrow[rr, dashed, "{[\![c]\!]}"] & & \sigma' \\
\mathcal{P}(\mathsf{States}) \arrow[ur, dashed, no head] \arrow[rr, "{(\!|c|\!)_\sigma}"'] & & \mathcal{P}(\mathsf{States}) \arrow[ur, dashed, no head] &
\end{tikzcd}
$$

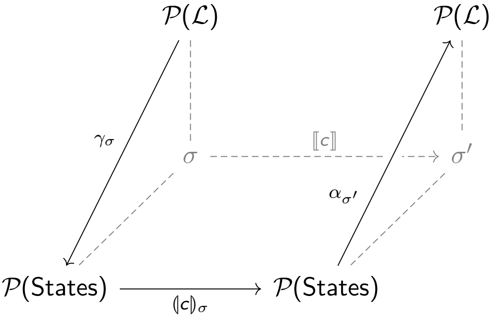
$$\begin{tikzcd}
 & \mathcal{P}(\mathcal{L}) \arrow[dl, "\gamma_\sigma"'] \arrow[d, dash, dashed] & & \mathcal{P}(\mathcal{L}) \arrow[d, dash, dashed] \\
 & \sigma \arrow[rr, dashed, "{[\![c]\!]}"] \arrow[dl, dash, dashed] & & \sigma' \arrow[dl, dash, dashed] \\
\mathcal{P}(\text{States}) \arrow[rr, "(\!|c|\!)_\sigma"'] & & \mathcal{P}(\text{States}) &
\end{tikzcd}$$

$$(\!|c|\!)_\sigma \circ \gamma_\sigma(\mathbb{A}x)$$

- Galois connections indexed by major states $\sigma \in$ States

$$(\mathcal{P}(\text{States}), \subseteq) \underset{\alpha_\sigma}{\overset{\gamma_\sigma}{\leftrightarrows}} (\mathcal{P}(\mathcal{L}), \sqsubseteq^\sharp)$$

$\mathcal{P}(\mathcal{L})$: sets of relational formulas interpreted conjunctively



$$\alpha_{\sigma'} \circ (\!|\mathsf{c}|\!)_\sigma \circ \gamma_\sigma(\mathbb{A}x)$$

- Galois connections indexed by major states $\sigma \in \text{States}$

$$(\mathcal{P}(\text{States}), \subseteq) \underset{\alpha_\sigma}{\overset{\gamma_\sigma}{\leftrightarrows}} (\mathcal{P}(\mathcal{L}), \sqsubseteq^\sharp)$$

$\mathcal{P}(\mathcal{L})$: sets of relational formulas interpreted conjunctively

Monitoring as Calculational Abstract Interpretation

$$\begin{tikzcd}
& \mathcal{P}(\mathcal{L}) \arrow[rr, "\alpha_{\sigma'} \circ (\!|\mathsf{c}|\!)_\sigma \circ \gamma_\sigma"] \arrow[ddl, "\gamma_\sigma"'] \arrow[d, dash, dashed] & & \mathcal{P}(\mathcal{L}) \arrow[d, dash, dashed] \\
& \sigma \arrow[rr, dashed, "[\![c]\!]"] \arrow[dl, dash, dashed] & & \sigma' \arrow[dl, dash, dashed] \\
\mathcal{P}(\text{States}) \arrow[rr, "(\!|\mathsf{c}|\!)_\sigma"'] & & \mathcal{P}(\text{States}) \arrow[uur, "\alpha_{\sigma'}"] &
\end{tikzcd}$$

$$\alpha_{\sigma'} \circ (\!|\mathsf{c}|\!)_\sigma \circ \gamma_\sigma(\mathbb{A}x)$$

- Galois connections indexed by major states $\sigma \in$ States

$$(\mathcal{P}(\text{States}), \subseteq) \underset{\alpha_\sigma}{\overset{\gamma_\sigma}{\leftrightarrows}} (\mathcal{P}(\mathcal{L}), \sqsubseteq^\sharp)$$

$\mathcal{P}(\mathcal{L})$: sets of relational formulas interpreted conjunctively

$$
\begin{tikzcd}
 & \mathcal{P}(\mathcal{L}) \arrow[rr, "\alpha_{\sigma'} \circ (\!|c|\!)_\sigma \circ \gamma_\sigma"] \arrow[rr, bend left, "(\!|c|\!)^\sharp_\sigma"] \arrow[rr, phantom, bend left=20, "\dot{\sqsubseteq}^\sharp"] \arrow[ddl, "\gamma_\sigma"'] \arrow[d, dashed, no head] & & \mathcal{P}(\mathcal{L}) \arrow[d, dashed, no head] \\
 & \sigma \arrow[rr, dashed, "\llbracket c \rrbracket"] \arrow[dl, dashed, no head] & & \sigma' \arrow[dl, dashed, no head] \\
\mathcal{P}(\text{States}) \arrow[rr, "(\!|c|\!)_\sigma"'] & & \mathcal{P}(\text{States}) \arrow[uur, "\alpha_{\sigma'}"] &
\end{tikzcd}
$$

$$\alpha_{\sigma'} \circ (\!|c|\!)_\sigma \circ \gamma_\sigma(\mathbb{A}x)$$

- Galois connections indexed by major states $\sigma \in \text{States}$

$$(\mathcal{P}(\text{States}), \subseteq) \xrightleftharpoons[\alpha_\sigma]{\gamma_\sigma} (\mathcal{P}(\mathcal{L}), \sqsubseteq^\sharp)$$

$\mathcal{P}(\mathcal{L})$: sets of relational formulas interpreted conjunctively

Let $\sigma' = [\![\textbf{if } b \textbf{ then } c_1 \textbf{ else } c_2]\!]\sigma$ and observe:

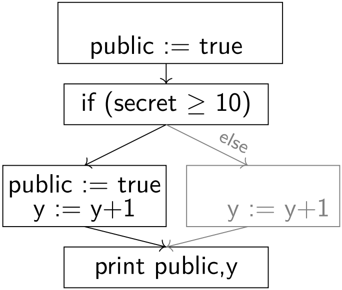
$$
\begin{array}{ll}
& \alpha_{\sigma'} \circ (\![\textbf{if } b \textbf{ then } c_1 \textbf{ else } c_2]\!)_\sigma \circ \gamma_\sigma \\
= & \quad \text{consider case } [\![b]\!]\sigma = 1 \text{, so } \sigma' = [\![c_1]\!]\sigma \\
& \alpha_{\sigma'} \circ \left((\![c_1]\!)_\sigma \circ \mathsf{grd}^b_\sigma \circ \gamma_\sigma \ \sqcup \ \{\![c_2]\!\} \circ \mathsf{grd}^{\neg b} \circ \gamma_\sigma \right) \\
= & \quad \text{Galois: } \alpha \text{ preserve joins} \\
& \alpha_{\sigma'} \circ (\![c_1]\!)_\sigma \circ \mathsf{grd}^b_\sigma \circ \gamma_\sigma \quad \sqcup^\sharp \quad \alpha_{\sigma'} \circ \{\![c_2]\!\} \circ \mathsf{grd}^{\neg b} \circ \gamma_\sigma \\
\sqsubseteq^\sharp & \quad \text{Galois: } id \sqsubseteq^\sharp \gamma_\sigma \circ \alpha_\sigma \\
& \alpha_{\sigma'} \circ (\![c_1]\!)_\sigma \circ \gamma_\sigma \circ \alpha_\sigma \circ \mathsf{grd}^b_\sigma \circ \gamma_\sigma \\
& \qquad \sqcup^\sharp \quad \alpha_{\sigma'} \circ \{\![c_2]\!\} \circ \gamma_\sigma \circ \alpha_\sigma \circ \mathsf{grd}^{\neg b} \circ \gamma_\sigma \\
\sqsubseteq^\sharp & \quad \text{by ind hyp: } \alpha_{\sigma'} \circ (\![c_1]\!)_\sigma \circ \gamma_\sigma \sqsubseteq^\sharp (\![c_1]\!)^\sharp_\sigma \\
& (\![c_1]\!)^\sharp_\sigma \circ \alpha_\sigma \circ \mathsf{grd}^b_\sigma \circ \gamma_\sigma \\
& \qquad \sqcup^\sharp \quad \alpha_{\sigma'} \circ \{\![c_2]\!\} \circ \gamma_\sigma \circ \alpha_\sigma \circ \mathsf{grd}^{\neg b} \circ \gamma_\sigma \\
\sqsubseteq^\sharp & \quad \text{posit a sound static analysis: } \alpha_{\sigma'} \circ \{\![c_2]\!\} \circ \gamma_\sigma \mathrel{\dot\sqsubseteq^\sharp} \{\![c_2]\!\}^\sharp \\
& (\![c_1]\!)^\sharp_\sigma \circ \alpha_\sigma \circ \mathsf{grd}^b_\sigma \circ \gamma_\sigma \ \sqcup^\sharp \ \{\![c_2]\!\}^\sharp \circ \alpha_\sigma \circ \mathsf{grd}^{\neg b} \circ \gamma_\sigma
\end{array}
$$

- A specification from which a tractable monitor is derived through the calculational framework of abstract interpretation [Cousots, 77, 79 & 99]

$$\alpha_{\sigma'} \circ (\!|c|\!)_\sigma \circ \gamma_\sigma \mathrel{\dot{\sqsubseteq}^\sharp} (\!|c|\!)^\sharp_\sigma$$

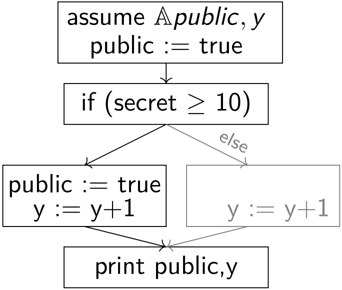
- Structural induction, standard derivation for most commands
- Design choices for the static part of the monitor in the case of “high branches”:
  - Always top: simulating a purely dynamic monitor forgetting all formulas [Besson et al.,13]
  - Modified variables: forgetting relational formulas that may be falsified [Le Guernic et al.,07] [Russo & Sabelfeld,10]
  - Interval analysis: inferring new relational formulas by leveraging actual values

Example Program Reloaded



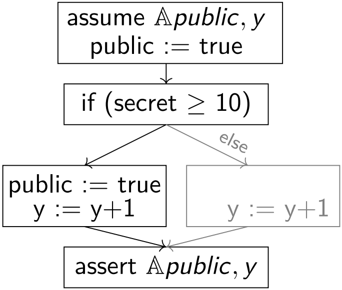
- Interval analysis of non-executed branch determines that:
  - variable public is equal to true
  - variable y is incremented by 1
- Comparison with major state after conditional determines :
  - all minor states agree with the major state on the value of both variables public and y
  - Formalised through a reduced product [Cousots, 79], providing an interface between relational formulas and intervals

Example Program Reloaded



- Interval analysis of non-executed branch determines that:
  - variable public is equal to true
  - variable y is incremented by 1
- Comparison with major state after conditional determines :
  - all minor states agree with the major state on the value of both variables public and y
  - Formalised through a reduced product [Cousots, 79], providing an interface between relational formulas and intervals

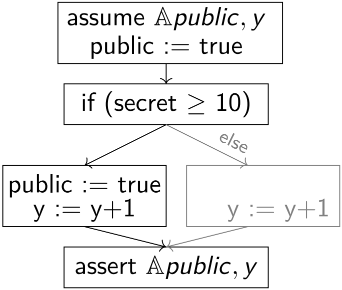
Example Program Reloaded



- Interval analysis of non-executed branch determines that:
  - variable public is equal to true
  - variable y is incremented by 1
- Comparison with major state after conditional determines :
  - all minor states agree with the major state on the value of both variables public and y
  - Formalised through a reduced product [Cousots, 79], providing an interface between relational formulas and intervals

Example Program Reloaded

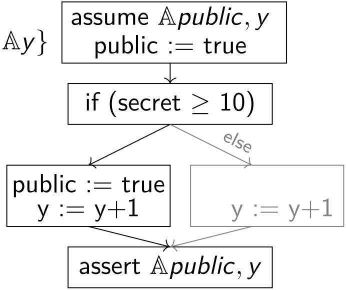
$secret^{100}\, public^{1}\, y^{5},$



- Interval analysis of non-executed branch determines that:
  - variable public is equal to true
  - variable y is incremented by 1
- Comparison with major state after conditional determines :
  - all minor states agree with the major state on the value of both variables public and y
  - Formalised through a reduced product [Cousots, 79], providing an interface between relational formulas and intervals

Example Program Reloaded

$secret^{100} public^{1} y^{5} \{\mathbb{A} public, \mathbb{A} y\}$

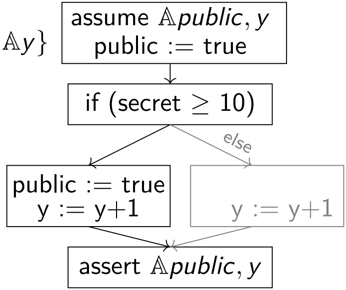


- Interval analysis of non-executed branch determines that:
  - variable public is equal to true
  - variable y is incremented by 1
- Comparison with major state after conditional determines :
  - all minor states agree with the major state on the value of both variables public and y
  - Formalised through a reduced product [Cousots, 79], providing an interface between relational formulas and intervals

Example Program Reloaded

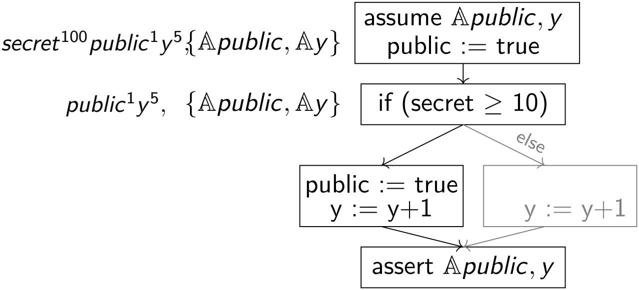
$secret^{100} public^{1} y^{5} \{\mathbb{A} public, \mathbb{A} y\}$

$public^{1} y^{5},$



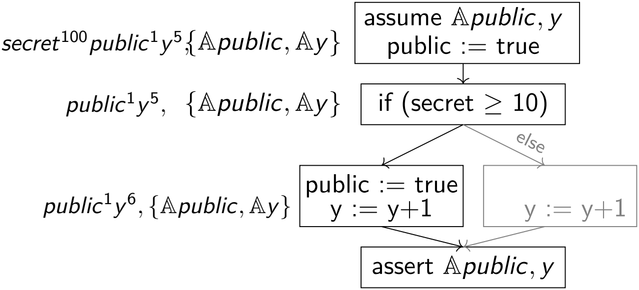
- Interval analysis of non-executed branch determines that:
  - variable public is equal to true
  - variable y is incremented by 1
- Comparison with major state after conditional determines :
  - all minor states agree with the major state on the value of both variables public and y
  - Formalised through a reduced product [Cousots, 79], providing an interface between relational formulas and intervals

Example Program Reloaded



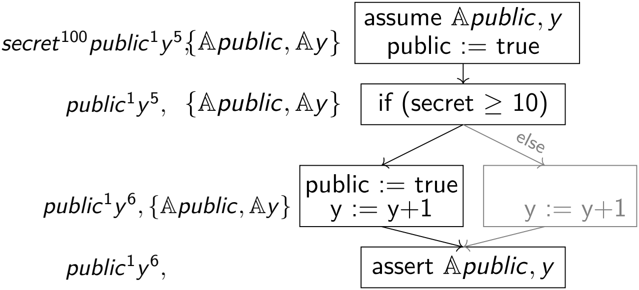
- Interval analysis of non-executed branch determines that:
  - variable public is equal to true
  - variable y is incremented by 1
- Comparison with major state after conditional determines :
  - all minor states agree with the major state on the value of both variables public and y
  - Formalised through a reduced product [Cousots, 79], providing an interface between relational formulas and intervals

Example Program Reloaded



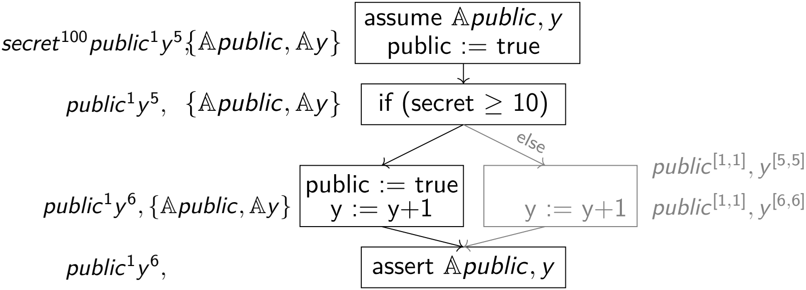
- Interval analysis of non-executed branch determines that:
  - variable public is equal to true
  - variable y is incremented by 1
- Comparison with major state after conditional determines :
  - all minor states agree with the major state on the value of both variables public and y
  - Formalised through a reduced product [Cousots, 79], providing an interface between relational formulas and intervals

Example Program Reloaded



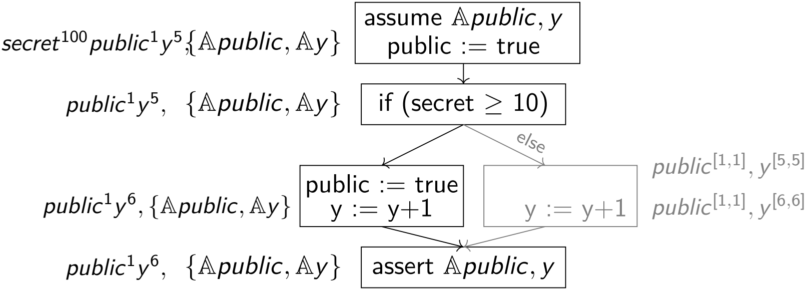
- Interval analysis of non-executed branch determines that:
  - variable public is equal to true
  - variable y is incremented by 1
- Comparison with major state after conditional determines :
  - all minor states agree with the major state on the value of both variables public and y
  - Formalised through a reduced product [Cousots, 79], providing an interface between relational formulas and intervals

Example Program Reloaded



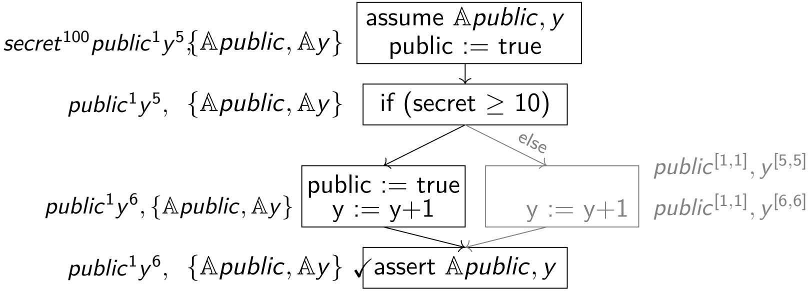
- Interval analysis of non-executed branch determines that:
  - variable public is equal to true
  - variable y is incremented by 1
- Comparison with major state after conditional determines :
  - all minor states agree with the major state on the value of both variables public and y
  - Formalised through a reduced product [Cousots, 79], providing an interface between relational formulas and intervals

Example Program Reloaded



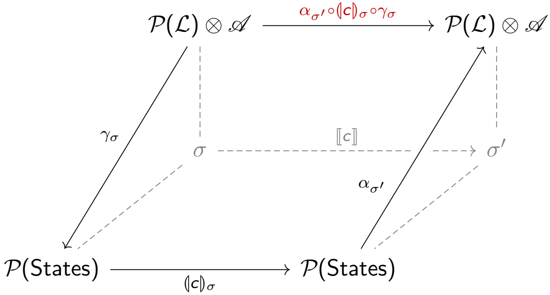
- Interval analysis of non-executed branch determines that:
  - variable public is equal to true
  - variable y is incremented by 1
- Comparison with major state after conditional determines :
  - all minor states agree with the major state on the value of both variables public and y
  - Formalised through a reduced product [Cousots, 79], providing an interface between relational formulas and intervals

Example Program Reloaded



- Interval analysis of non-executed branch determines that:
  - variable public is equal to true
  - variable y is incremented by 1
- Comparison with major state after conditional determines :
  - all minor states agree with the major state on the value of both variables public and y
  - Formalised through a reduced product [Cousots, 79], providing an interface between relational formulas and intervals

$$\begin{tikzcd}
 & \mathcal{P}(\mathcal{L}) \arrow[dl, "\gamma_\sigma"'] \arrow[rr, "\alpha_{\sigma'} \circ (\!| c |\!)_\sigma \circ \gamma_\sigma"] \arrow[d, dashed, no head] & & \mathcal{P}(\mathcal{L}) \arrow[d, dashed, no head] \\
 & \sigma \arrow[rr, dashed, "\llbracket c \rrbracket"] \arrow[dl, dashed, no head] & & \sigma' \arrow[dl, dashed, no head] \\
\mathcal{P}(\text{States}) \arrow[rr, "(\!| c |\!)_\sigma"'] & & \mathcal{P}(\text{States}) \arrow[uur, "\alpha_{\sigma'}"] &
\end{tikzcd}$$

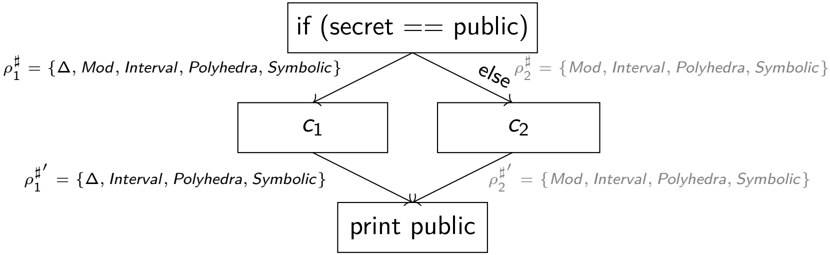


Future work

- Exploring more systematically the design space of information flow monitors
    - Better precision, more subtle policies, richer languages, less overhead . . .

Sweet spot performance/precision?

$$\rho_0^\sharp = \{\Delta, Mod, Interval, Polyhedra, Symbolic\}$$



$$reduce(\rho_1^{\sharp\prime} \sqcup^\sharp \rho_2^{\sharp\prime})$$

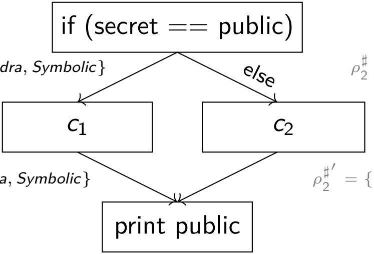
- Trade-off performance and precision during the monitoring process

Sweet spot performance/precision?

$$\rho_0^\sharp = \{\Delta, \mathit{Mod}, \mathit{Interval}, \mathit{Polyhedra}, \mathit{Symbolic}\}$$

$$\rho_1^\sharp = \{\Delta, \mathit{Mod}, \mathit{Interval}, \mathit{Polyhedra}, \mathit{Symbolic}\}$$

$$\rho_2^\sharp = \{\mathit{Mod}, \top, \top, \top\}$$



$$\rho_1^{\sharp\prime} = \{\Delta, \mathit{Interval}, \mathit{Polyhedra}, \mathit{Symbolic}\}$$

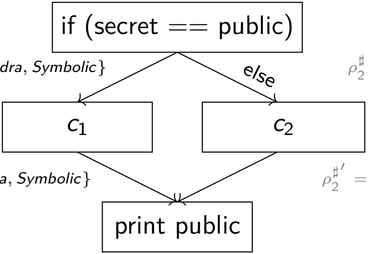
$$\rho_2^{\sharp\prime} = \{\mathit{Mod}, \top, \top, \top, \top\}$$

$$\mathit{reduce}(\rho_1^{\sharp\prime} \sqcup^\sharp \rho_2^{\sharp\prime})$$

- Trade-off performance and precision during the monitoring process

Sweet spot performance/precision?

$$\rho_0^\sharp = \{\Delta, \mathit{Mod}, \mathit{Interval}, \mathit{Polyhedra}, \mathit{Symbolic}\}$$



$\rho_1^\sharp = \{\Delta, \mathit{Mod}, \mathit{Interval}, \mathit{Polyhedra}, \mathit{Symbolic}\}$ $\rho_2^\sharp = \{\top, \top, \top, \top, \top\}$

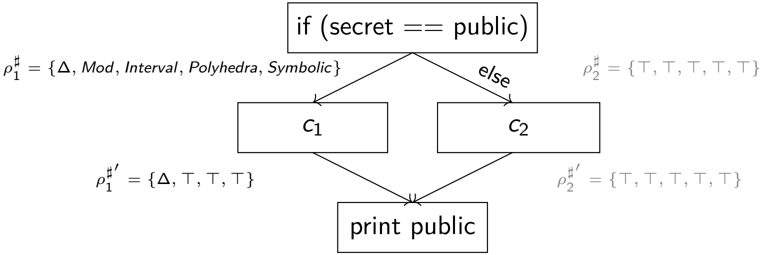
${\rho_1^\sharp}' = \{\Delta, \mathit{Interval}, \mathit{Polyhedra}, \mathit{Symbolic}\}$ ${\rho_2^\sharp}' = \{\top, \top, \top, \top, \top\}$

$$\mathit{reduce}({\rho_1^\sharp}' \sqcup^\sharp {\rho_2^\sharp}')$$

- Trade-off performance and precision during the monitoring process

Sweet spot performance/precision?

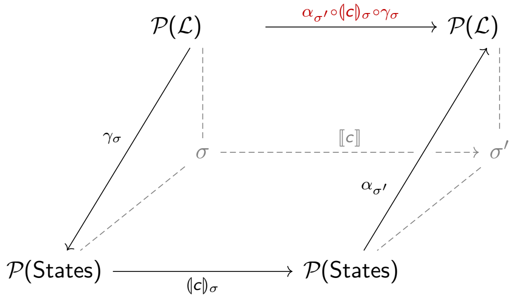
$$\rho_0^\sharp = \{\Delta, \mathit{Mod}, \mathit{Interval}, \mathit{Polyhedra}, \mathit{Symbolic}\}$$



$$\mathit{reduce}(\rho_1^{\sharp\prime} \sqcup^\sharp \rho_2^{\sharp\prime})$$

- Trade-off performance and precision during the monitoring process

From Static to Dynamic, and Back



Future work

- Exploring more systematically the design space of information flow monitors
  - Better precision, more subtle policies, richer languages, less overhead . . .
- Hand in hand with semantic-based static analysis of security requirements by calculational abstract interpretation

  [arxiv.org/abs/1608.01654, to appear'17]

$$
\begin{tikzcd}
& \mathcal{P}(\mathcal{L}) \otimes \mathscr{A} \arrow[rr, "\alpha_{\sigma'} \circ (\!|c|\!)_\sigma \circ \gamma_\sigma"] \arrow[d, dash, dashed] \arrow[ddl, "\gamma_\sigma"'] & & \mathcal{P}(\mathcal{L}) \otimes \mathscr{A} \arrow[d, dash, dashed] \\
& \sigma \arrow[rr, dashed, "\llbracket c \rrbracket"] \arrow[dl, dash, dashed] & & \sigma' \arrow[dl, dash, dashed] \\
\mathcal{P}(\text{States}) \arrow[rr, "(\!|c|\!)_\sigma"'] & & \mathcal{P}(\text{States}) \arrow[uur, "\alpha_{\sigma'}"] &
\end{tikzcd}
$$

Future work

- Exploring more systematically the design space of information flow monitors
  - Better precision, more subtle policies, richer languages, less overhead ...
- Hand in hand with semantic-based static analysis of security requirements by calculational abstract interpretation

  [arxiv.org/abs/1608.01654, to appear'17]

Questions? :)