CS 5430:

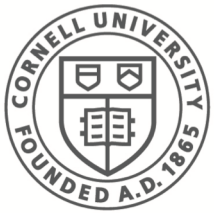
Information Flow

Part II: Dynamic Enforcement

Fred B. Schneider

Samuel B Eckert Professor of Computer Science

Department of Computer Science
Cornell University
Ithaca, New York 14853
U.S.A.



Cornell CIS
Computer Science

Enforcement of FBAC

FLI imposes restrictions on each statement.

$$v \to w \implies \Gamma(v) \sqsubseteq \Gamma(w)$$

- Static Enforcement
  - Compiler ensures type-correct programs satisfy restrictions.
- Dynamic Enforcement
  - run-time checks ensure program execution satisfies restrictions.
  - changes to labels mean program execution satisfies restrictions.

Why Dynamic Enforcement?

- Static enforcement: Rejects program if any execution could violate Flow-Label invariant.
- Dynamic enforcement: Blocks after partial execution when Flow-Label invariant could be violated.

$$\textbf{if } 0 = 0 \textbf{ then } x_L := 2 \textbf{ else } x_L := x_H \textbf{ fi}$$

Why Dynamic Enforcement?

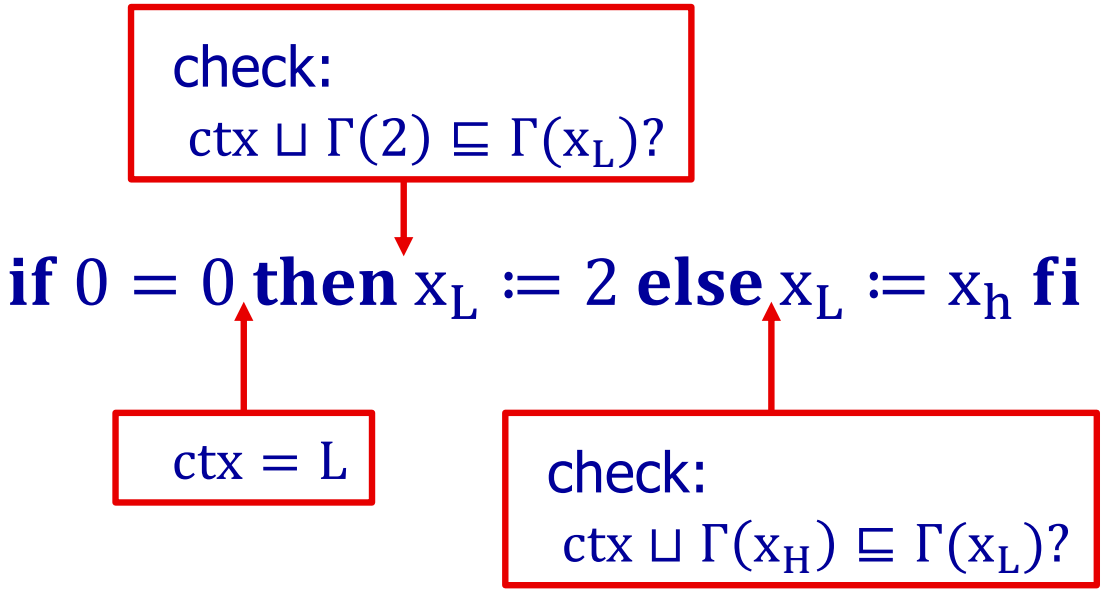
- Static enforcement: Rejects program if any execution could violate Flow-Label invariant.
- Dynamic enforcement: Blocks after partial execution when Flow-Label invariant could be violated.

Type error!

$$\textbf{if } 0 = 0 \textbf{ then } x_L := 2 \textbf{ else } x_L := x_H \textbf{ fi}$$

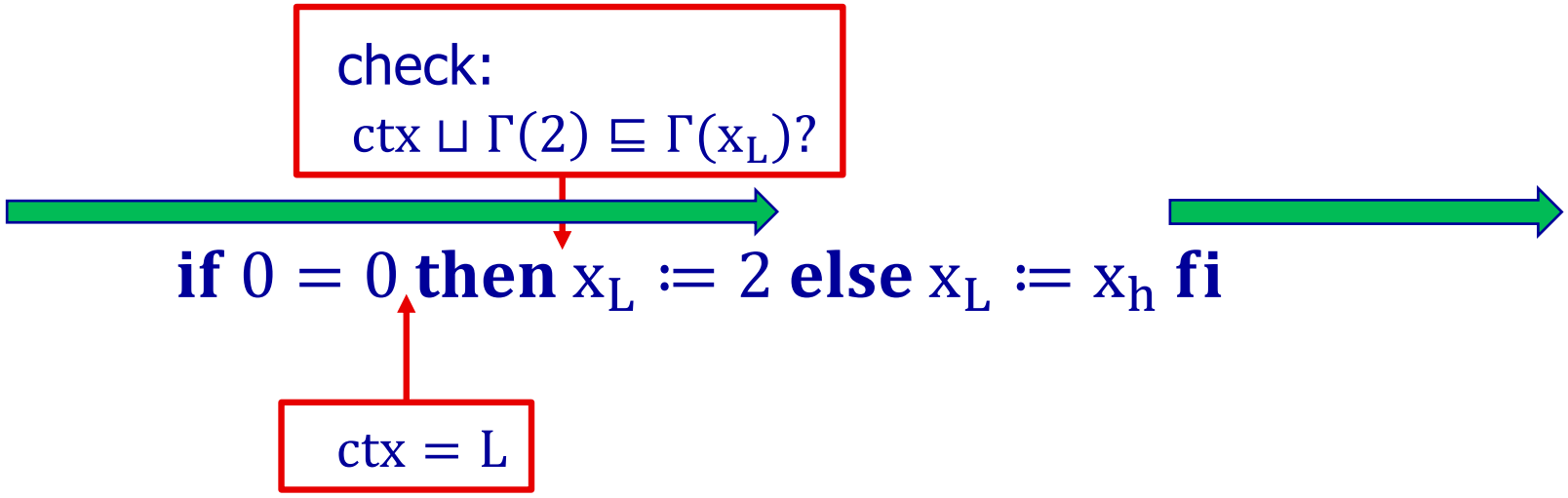
Why Dynamic Enforcement?

- Static enforcement: Rejects program if execution could violates Flow-Label invariant
- Dynamic enforcement: Blocks after partial execution when Flow-Label invariant could be violated.



Why Dynamic Enforcement?

- Static enforcement: Rejects program if execution could violates Flow-Label invariant
- Dynamic enforcement: Blocks after partial execution when Flow-Label invariant could be violated.



Implementing Dynamic Enforcement

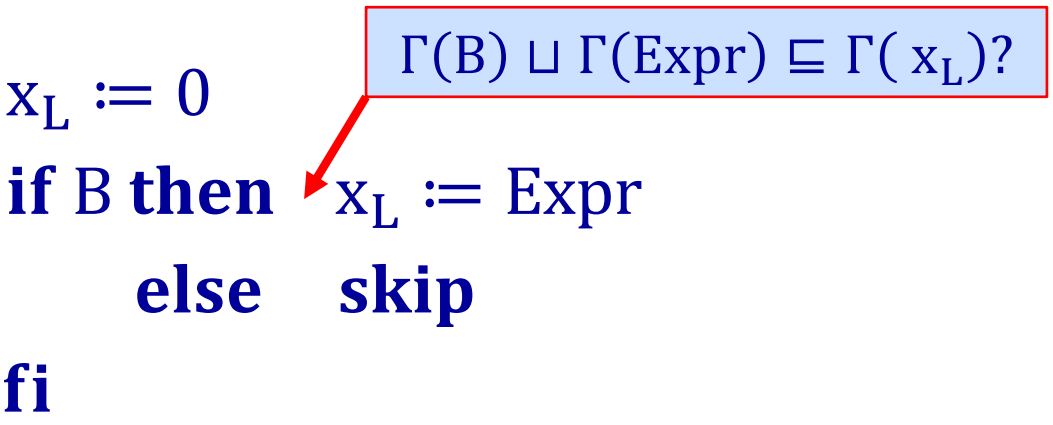
Conjecture: To implement dynamic enforcement:

- Precede $\mathrm{x} := \mathrm{Expr}$ with check: "$\mathrm{ctx} \sqcup \Gamma(\mathrm{Expr}) \sqsubseteq \Gamma(\mathrm{x})$?"
- Block execution if check fails.

Implementing Dynamic Enforcement

Conjecture: To implement dynamic enforcement:

- Precede $x := Expr$ with check: "$ctx \sqcup \Gamma(Expr) \sqsubseteq \Gamma(x)$?"
- Block execution if check fails



Implementing Dynamic Enforcement

Conjecture:

- Precede $x := Expr$ with check: "$ctx \sqcup \Gamma(Expr) \sqsubseteq \Gamma(x)$?"
- Block execution if check fails

$\Gamma(B) \sqcup \Gamma(Expr) \sqsubseteq \Gamma(x_L)$?

$x_L := 0$

if B **then** $x_L := Expr$

else **skip**

fi

But… when stop on check:

- … B=true leaks!
- Result: implemented RNI (=termination insensitive) only

Solution: Hybrid Enforcement

$x_L := 0$

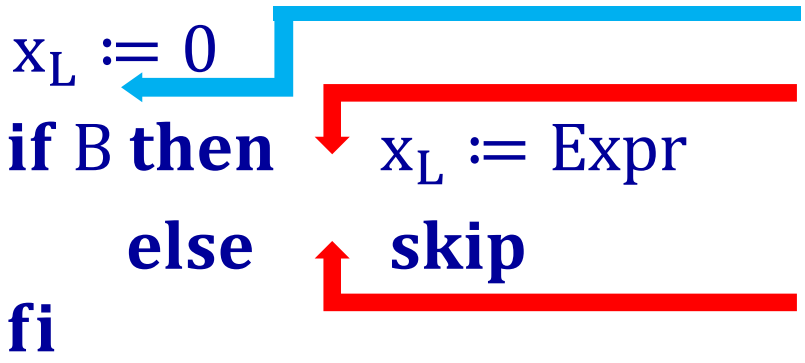
if B **then** $x_L := Expr$

else **skip**

fi

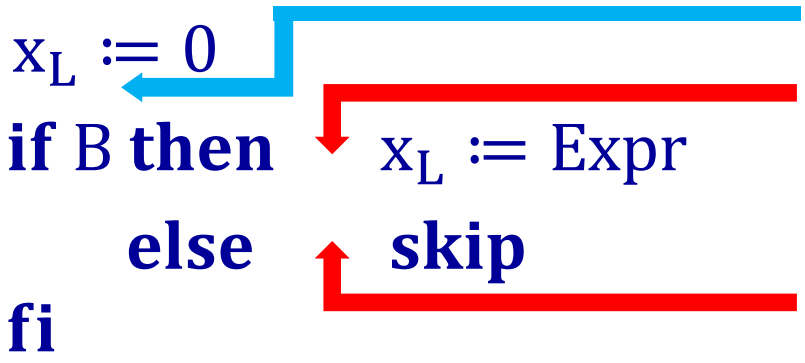
- $B \rightarrow x_L$ whether or not $x_L := Expr$ executes.
  - For $\Gamma(B) = H$, could exist memories M and M' with different H values causing termination with x_L having different values.

Solution: Hybrid Enforcement



- $B \rightarrow x_L$ whether or not $x_L := Expr$ executes.
  - For $\Gamma(B) = H$, could exist memories M and M' with different H values causing termination with x_L having different values.
  - FLI requires $\Gamma(B) \sqsubseteq x_L$
    - Before **if** -or- Within **then** and within **else**
  - FLI also requires $\Gamma(Expr) \sqsubseteq \Gamma(x_L)$ before $x_L := Expr$

Solution: Hybrid Enforcement



- $B \rightarrow x_L$ whether or not $x_L := Expr$ executes.
  - For $\Gamma(B) = H$, could exist memories M and M' with different H values causing termination with x_L having different values.
  - FLI requires $\Gamma(B) \sqsubseteq x_L$
    - Before **if** -or- Within **then** and within **else** -or- After **if**
  - FLI also requires $\Gamma(Expr) \sqsubseteq \Gamma(x_L)$ before $x_L := Expr$
- What if B is $x_H \neq x_H$? (… enforcement is conservative.)

Hybrid Enforcement: Summary

$$\textbf{if } B \textbf{ then } C_1 \textbf{ else } C_2 \textbf{ fi}$$

- Insert check $\Gamma(\mathrm{Expr}) \sqsubseteq \Gamma(x)$ before execution of each "$x := \mathrm{Expr}$" in C_1 or C_2.
- Insert check $\Gamma(B) \sqsubseteq \Gamma(x)$ within execution of both C_1 and C_2 if "$x := \ldots$" appears anywhere within C_1 or within C_2.

Flow-Sensitive Labels

A given variable might be given different **flow-sensitive** labels during execution.

Example:

$$x := Hval; \quad x := 0; \quad x_L := x$$

Observe:

- If $\Gamma(x) = H$ then program does not type check.

Flow-Sensitive Labels

A given variable might be given different **flow-sensitive** labels during execution.

Example:

$$x := \mathrm{Hval}; \quad x := 0; \quad x_L := x$$

red given label H; green given label L

Program does type check and satisfies:

$$v \to w \Rightarrow \Gamma(v) \sqsubseteq \Gamma(w)$$

Flow Sensitive Labels + Dynamic?

$x := 0 \ \{\Gamma(x) = L\}$

if $h > 0$ **then** $x := 2; \ \{\Gamma(x) = \Gamma(h) = H\}$

else **skip**

fi

- $h > 0$ is true: After **fi** $\Gamma(x) = H$
- $h > 0$ is false: After **fi** $\Gamma(x) = L$

Problem: $h \rightarrow x$ but $\Gamma(h) \not\sqsubseteq \Gamma(x)$

Flow Sensitive + … Soln 1

Rule: Block execution from entering conditional commands with high guards and lower targets.


```
x := 0
if h > 0 then     x := 2
         else     skip
fi
```

Flow Sensitive + … Soln 2

Rule: Update labels of target variables in untaken branches to capture implicit flow.

$x := 0$

if $h > 0$ **then** $x := 2;\ \Gamma(x) := \Gamma(h)$

else **skip**; $\Gamma(x) := \Gamma(h)$

fi

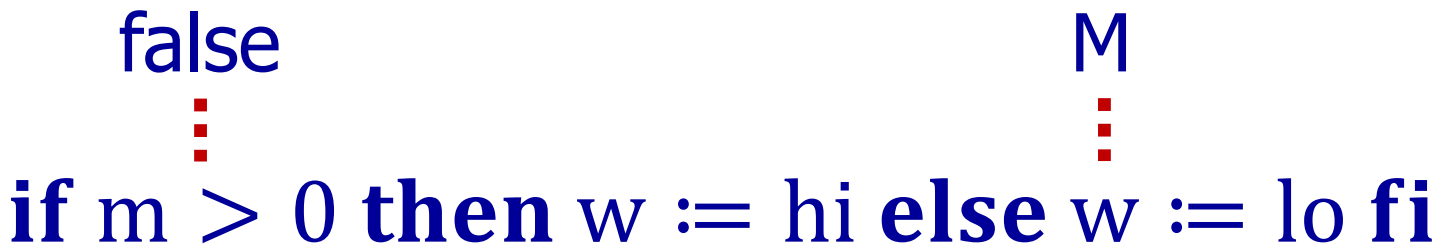
Leaks thru Flow-Sensitive Labels

Suppose: $\Gamma(\mathrm{m}) = \mathrm{M}$ and $\mathrm{L} \sqsubseteq \mathrm{M} \sqsubseteq \mathrm{H}$

if $\mathrm{m} > 0$ **then** $\mathrm{w} := \mathrm{hi}$ **else** $\mathrm{w} := \mathrm{lo}$ **fi**

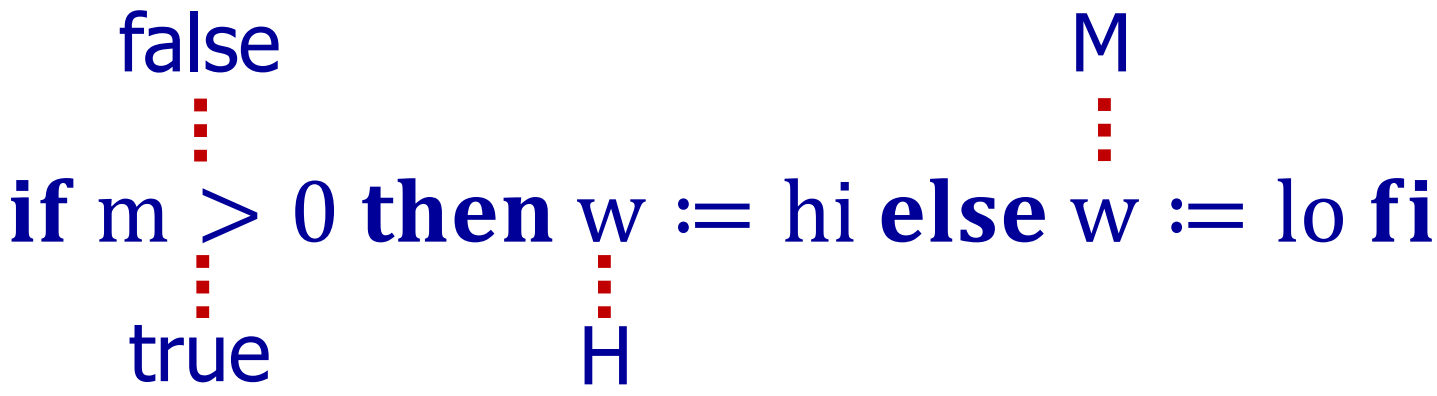
Leaks thru Flow-Sensitive Labels

Suppose: $\Gamma(\mathrm{m}) = \mathrm{M}$ and $\mathrm{L} \sqsubseteq \mathrm{M} \sqsubseteq \mathrm{H}$



Leaks thru Flow-Sensitive Labels

Suppose: $\Gamma(\mathrm{m}) = \mathrm{M}$ and $\mathrm{L} \sqsubseteq \mathrm{M} \sqsubseteq \mathrm{H}$



- Value of m leaks to label (M vs H) of w.

Avoiding Leaks thru Flow Sensitive 1

Rule: Use the same flow-sensitive label for an assignment target, independent of guard.

Example

$$\textbf{if } m > 0 \textbf{ then } w := hi \textbf{ else } w := lo \textbf{ fi}$$

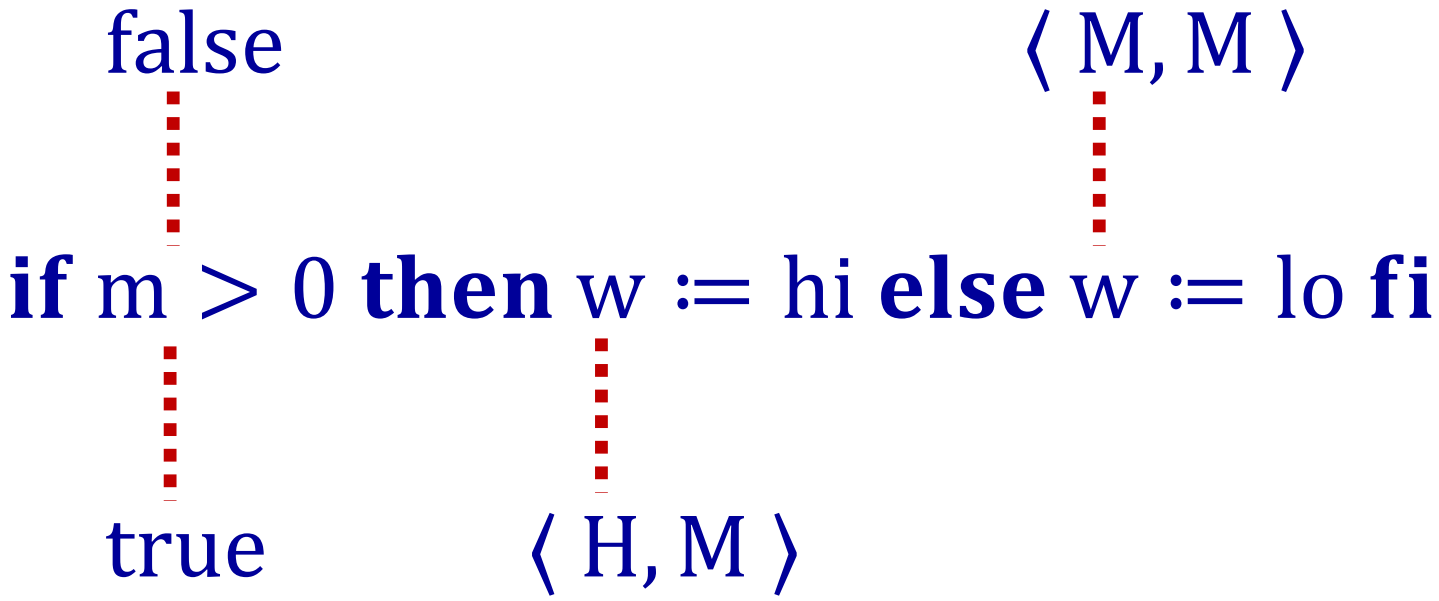
w : H (then branch), H (else branch)

(Sound but conservative.)

Avoiding Leaks thru Flow Sensitive 2

Rule: Associate a metalabel with each label.

Example:



Labels for meta-labels?

Summary

FLI: $v \to w \implies \Gamma(v) \sqsubseteq \Gamma(w)$

- Static enforcement
  - Conservative
- Dynamic enforcement
  - Insert tests
    - Mind the untaken assignment!
  - Change labels
    - Static
    - Dynamic: Leaks thru labels