CS 5430

Information-Flow Control

Elisavet Kozyri

Spring 2017

Review: Labels represent IF policies

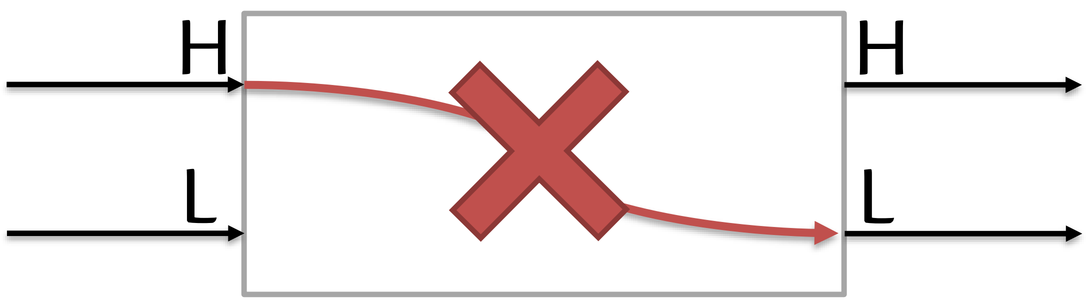
A label ℓ on some data represents an IF policy.

Possible labels for confidentiality:

- Classifications
  - Unclassified (U), Confidential (C), Secret (S), Top Secret (TS)
  - Low (L), high (H)
- Sets of principals
  - {Alice, Bob}, {Alice}, {Bob}, {}

Review: Noninterference

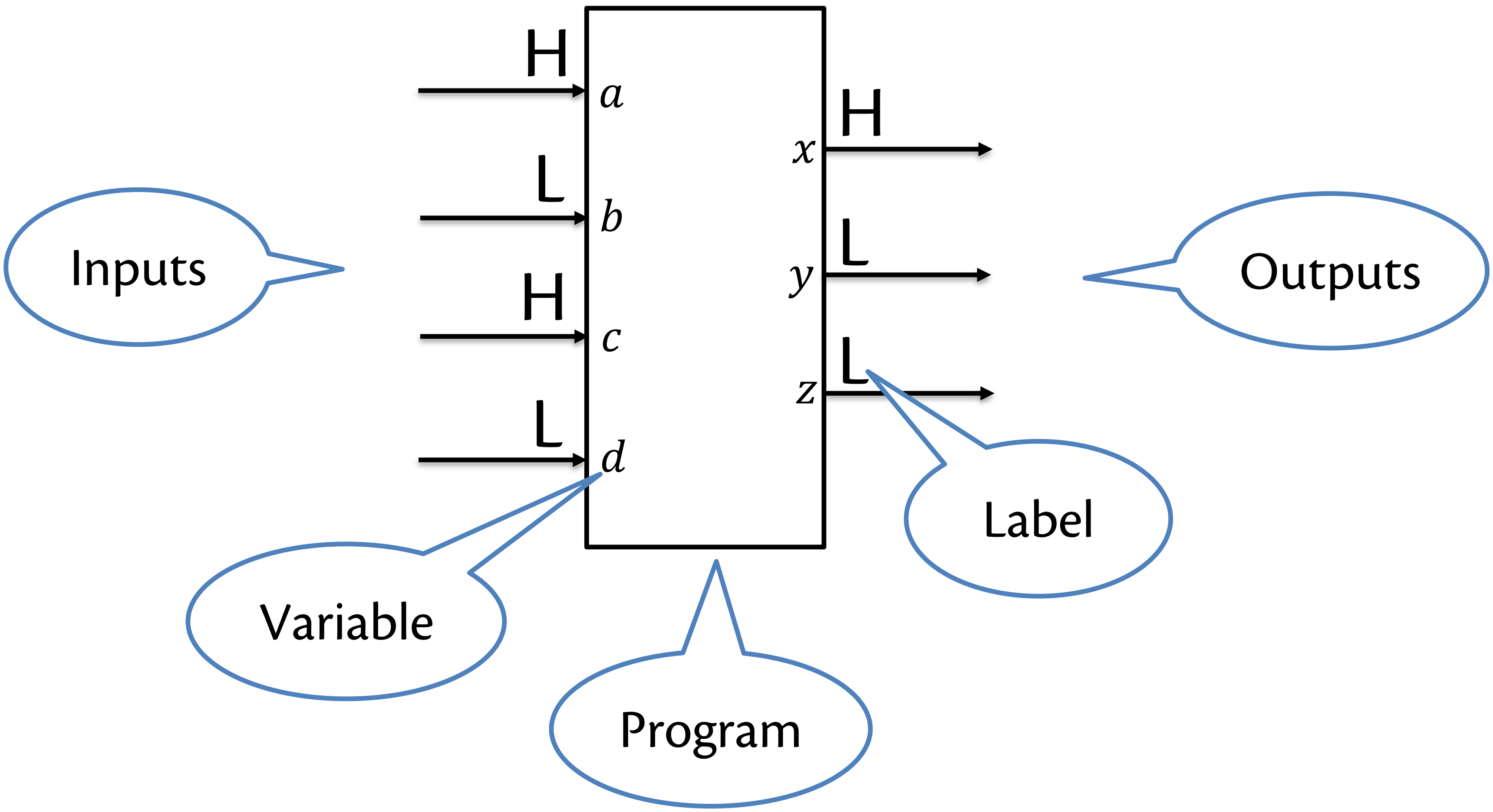
- Given a program, and
- given a mapping from variables to labels,
- it usually suffices to enforce noninterference.



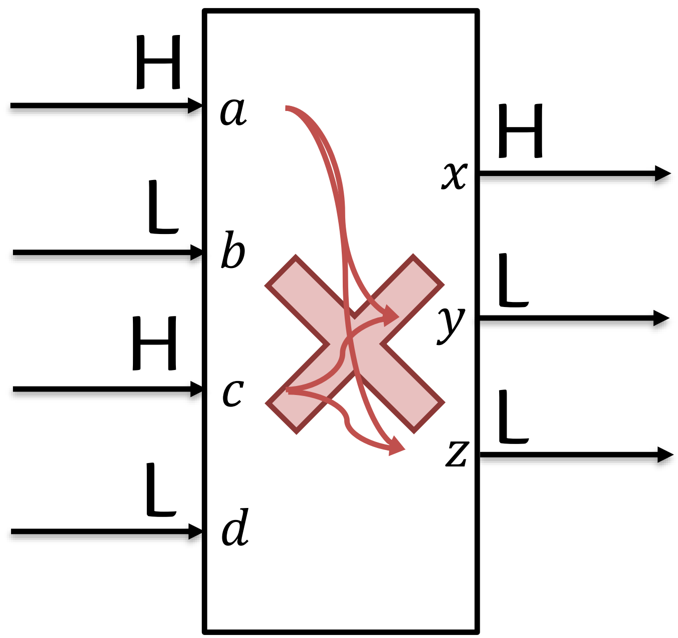
Today: Information Flow Control

- **Goal**: Enforce IF policies that tag variables in a program.
- There is a mapping Γ from variables to labels, which represent desired IF policies.
- The enforcement mechanism should ensure that a given program and a given Γ satisfy noninterference.

Information Flow Control

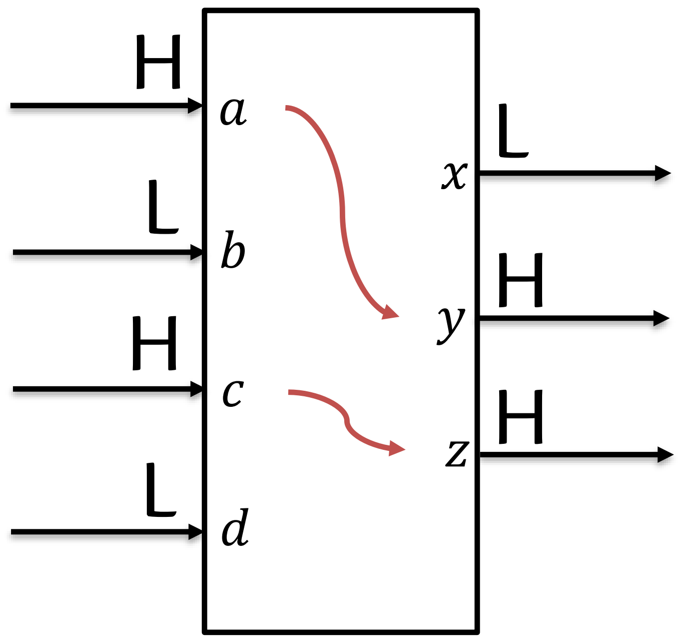


Information Flow Control: fixed Γ



- Γ remains the same during the analysis of the program.
- The mechanism checks that Γ satisfies noninterference.
- The program is rejected, if at least one red arrow appears in the program.

Information flow control: flow-sensitive Γ



- Γ may change during the analysis of the program.
- The mechanism deduces Γ(x), Γ(y), Γ(z) such that noninterference is satisfied.
- The program is never rejected.

Enforcing IF policies

- Static mechanism
  – Checking and/or deduction of labels before execution.
- Dynamic mechanism
  – Checking and/or deduction of labels during execution.
- Hybrid mechanism
  – Combination of static and dynamic.

Enforcement mechanism for today:

- Static
- Fixed Γ
  - So, for a program, the mechanism only needs to check whether Γ satisfies noninterference (NI).

Programs are written using this syntax:

```
e ::= x | n | e1+e2 | ...

c ::= x := e
    | if e then c1 else c2
    | while e do c
    | c1; c2
```

Checking an assignment

```
x := y
```

<u>Examples for confidentiality</u>

$\Gamma(\mathbf{x})$ is L.
$\Gamma(\mathbf{y})$ is L.
Does this assignment satisfy NI?

$\Gamma(\mathbf{x})$ is H.
$\Gamma(\mathbf{y})$ is L.
Does this assignment satisfy NI?

$\Gamma(\mathbf{x})$ is L.
$\Gamma(\mathbf{y})$ is H.
Does this assignment satisfy NI?

Order relation on labels

- $\ell \sqsubseteq \ell'$ iff ℓ' is **at least as restrictive as** ℓ.
- Values in variables tagged with ℓ *may flow to* variables tagged with ℓ'.
- Examples (for confidentiality):
  - $L \sqsubseteq H$
  - {Alice} $\sqsubseteq$ {}
  - {Alice,Bob} $\sqsubseteq$ {Alice}
- Relation $\sqsubseteq$ should be:
  - reflexive, transitive, and antisymmetric.
- There is a label $\perp$ (**bottom**) that is less restrictive than all other labels.
- There is a label $\top$ (**top**) that is more restrictive than all other labels.

Checking an assignment

Assignments cause **explicit** flows of values.

$$\mathbf{x} := \mathbf{y}$$

It satisfies NI, if $\Gamma(\mathbf{y}) \sqsubseteq \Gamma(\mathbf{x})$.

Checking an assignment:
connection with MLS

$$\mathbf{x} := \mathbf{y}$$

It satisfies NI, if $\Gamma(\mathbf{y}) \sqsubseteq \Gamma(\mathbf{x})$.

<u>MLS for confidentiality</u>

"no read up":

S may read O iff Label(O) $\sqsubseteq$ Label (S)

"no write down":

S may write O' iff Label(S) $\sqsubseteq$ Label (O')

Checking an assignment:
connection with MLS

$$\mathbf{x} := \mathbf{y}$$

It satisfies NI, if $\Gamma(\mathbf{y}) \sqsubseteq \Gamma(\mathbf{x})$.

<u>MLS for confidentiality</u>

"no read up":

CPU may read $\mathbf{y}$ iff Label($\mathbf{y}$) $\sqsubseteq$ Label (CPU)

"no write down":

CPU may write $\mathbf{x}$ iff Label(CPU) $\sqsubseteq$ Label ($\mathbf{x}$)

Checking an assignment

```
x := y + z
```

It satisfies NI, if $\Gamma(\mathtt{y}) \sqsubseteq \Gamma(\mathtt{x})$ and $\Gamma(\mathtt{z}) \sqsubseteq \Gamma(\mathtt{x})$.

It satisfies NI, if $\Gamma(\mathtt{y+z}) \sqsubseteq \Gamma(\mathtt{x})$.

Operator for combining labels

- For each ℓ and ℓ', there should exist label $\ell \sqcup \ell'$, such that:
  - $\ell \sqsubseteq \ell \sqcup \ell'$, $\ell' \sqsubseteq \ell \sqcup \ell'$, and
  - if $\ell \sqsubseteq \ell''$ and $\ell' \sqsubseteq \ell''$, then $\ell \sqcup \ell' \sqsubseteq \ell''$.
- $\ell \sqcup \ell'$ is called the **join** of ℓ and ℓ'.
- Operator $\sqcup$ is associative and commutative.

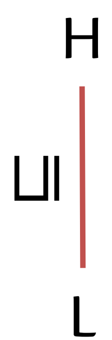
Checking an assignment

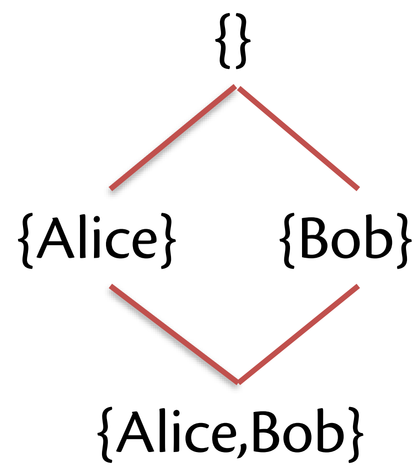
```
x := y + z
```

It satisfies NI, if $\Gamma(\mathbf{y}) \sqcup \Gamma(\mathbf{z}) \sqsubseteq \Gamma(\mathbf{x})$.

Lattice of labels

- The set of labels and relation ⊑ define a lattice, with join operator ⊔.
- Examples (confidentiality):





Checking an if-statement

```
if z>0 then
    if y>0 then x:=1 else x:=2
else
    x:=3
```

<u>Examples for confidentiality</u>

$\Gamma(\mathbf{x})$ is L.
$\Gamma(\mathbf{y})$, $\Gamma(\mathbf{z})$is L.
Does this if-statement satisfy NI?

$\Gamma(\mathbf{x})$, $\Gamma(\mathbf{z})$is L.
$\Gamma(\mathbf{y})$ is H.
Does this if-statement satisfy NI?

$\Gamma(\mathbf{x})$, $\Gamma(\mathbf{y})$is L.
$\Gamma(\mathbf{z})$ is H.
Does this if-statement satisfy NI?

Checking an if-statement

```
if z>0 then
    if y>0 then x:=1 else x:=2
else
    x:=3
```

Conditional commands (e.g., if-statements and while-statements) cause **implicit** flows of values.

Context

```
if z>0 then
    if y>0 then x:=1 else x:=2
else
    x:=3
```

It reveals information about **z>0**.

They reveal information about **z>0** and **y>0**.

Context label ctx

```
if z>0 then
    if y>0 then x:=1 else x:=2
else
    x:=3
```

Their ctx' is $\Gamma(\mathbf{z}) \sqcup \Gamma(\mathbf{y})$.

Its ctx is $\Gamma(\mathbf{z})$.

Context label ctx

```
if z>0 then
    if y>0 then x:=1 else x:=2
else
    x:=3
```

Check if $ctx' \sqsubseteq \Gamma(\mathbf{x})$,
where $ctx' = \Gamma(\mathbf{z}) \sqcup \Gamma(\mathbf{y})$.

Check if $ctx \sqsubseteq \Gamma(\mathbf{x})$,
where $ctx = \Gamma(\mathbf{z})$.

Context label *ctx*

```
if z>0 then
    if y>0 then x:=e else x:=2
else
    x:=3
```

Check if
$ctx' \sqcup \Gamma(\mathbf{e}) \sqsubseteq \Gamma(\mathbf{x})$.

Implicit flow

Explicit flow

Typing system for IF control

- Static
- Fixed Γ
- Labels as types
  - Label $\Gamma(\mathbf{x})$ is the type of $\mathbf{x}$.
- Typing rules for all possible commands.
- **Goal**: type-correctness $\Rightarrow$ noninterference

We are already familiar with typing systems!

Example of typing rule from Java or OCaml:

```
x + y : int
  if x : int
  and y : int
```

Typing rules for expressions

Judgement $\Gamma \vdash \mathbf{e} : \ell$

- According to mapping Γ, expression $\mathbf{e}$ has type (i.e., label) ℓ.

Constant: $\Gamma \vdash \mathbf{n} : \bot$

Variable: $\Gamma \vdash \mathbf{x} : \Gamma(\mathbf{x})$

Expression: $\Gamma \vdash \mathbf{e+e'} : \ell \sqcup \ell'$

if $\Gamma \vdash \mathbf{e} : \ell$

and $\Gamma \vdash \mathbf{e'} : \ell'$

Typing rules for expressions

Expression: $\Gamma \vdash \mathtt{e+e'} : \ell \sqcup \ell'$
$\quad\quad\quad\quad\quad$ **if** $\Gamma \vdash \mathtt{e} : \ell$
$\quad\quad\quad\quad\quad$ **and** $\Gamma \vdash \mathtt{e'} : \ell'$

Inference rule:

Premises →
Conclusion →

$$\frac{\Gamma \vdash \mathtt{e} : \ell \qquad \Gamma \vdash \mathtt{e'} : \ell'}{\Gamma \vdash \mathtt{e+e'} : \ell \sqcup \ell'}$$

Example

- Let $\Gamma(\mathbf{x}) = \mathrm{L}$ and $\Gamma(\mathbf{y}) = \mathrm{H}$.
- What is the type of **x+y+1**?
- *Proof tree:*

$$
\dfrac{
\dfrac{\Gamma(\mathtt{x}) = \mathrm{L}}{\Gamma \vdash \mathtt{x} : \mathrm{L}}
\qquad
\dfrac{\Gamma(\mathtt{y}) = \mathrm{H}}{\Gamma \vdash \mathtt{y} : \mathrm{H}}
\qquad
\dfrac{}{\Gamma \vdash \mathtt{1} : \mathrm{L}}
}{
\Gamma \vdash \mathtt{x + y + 1} : \mathrm{H}
}
$$

Typing rules for commands

Judgement $\Gamma, ctx \vdash$ `c`

- According to mapping Γ, and context label ctx, command `c` is type correct.

Assignment rule

$\Gamma, ctx \vdash \mathbf{\texttt{x:=e}}$

$\quad \mathbf{\texttt{if}} \;\; \Gamma \vdash \mathbf{\texttt{e}} : \ell$

$\quad \mathbf{\texttt{and}} \;\; \ell \sqcup ctx \sqsubseteq \Gamma\mathbf{\texttt{(x)}}$

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

If-rule

$$\frac{\Gamma \vdash \mathbf{\texttt{e}} : \ell \qquad \Gamma, \ell \sqcup ctx \vdash \mathbf{\texttt{c1}} \qquad \Gamma, \ell \sqcup ctx \vdash \mathbf{\texttt{c2}}}{\Gamma, ctx \vdash \mathbf{\texttt{if e then c1 else c2}}}$$

If-rule (example)

$$\dfrac{\Gamma \vdash \mathtt{\mathbf{z>0}} : \Gamma(\mathtt{\mathbf{z}}) \qquad \begin{array}{l}\Gamma, \Gamma(\mathtt{\mathbf{z}}) \sqcup \mathrm{L} \vdash \mathtt{\mathbf{if\ y>0\ then\ x:=1}} \\ \qquad\qquad\qquad\quad \mathtt{\mathbf{else\ x:=2}}\end{array} \qquad \Gamma, \Gamma(\mathtt{\mathbf{z}}) \sqcup \mathrm{L} \vdash \mathtt{\mathbf{x:=3}}}{\begin{array}{l}\Gamma, \mathrm{L} \vdash \mathtt{\mathbf{if\ z>0\ then\ \{if\ y>0\ then\ x:=1\ else\ x:=2\}}} \\ \qquad\qquad\quad \mathtt{\mathbf{else\ \{x:=3\}}}\end{array}}$$

What is the relation between $\Gamma(\mathtt{\mathbf{x}})$, $\Gamma(\mathtt{\mathbf{y}})$, and $\Gamma(\mathtt{\mathbf{z}})$, such that the above judgement can be proved?

If-rule (example)

$$\dfrac{\Gamma \vdash \mathtt{z>0} : \Gamma(\mathtt{z}) \quad \dfrac{\Gamma \vdash \mathtt{y>0} : \Gamma(\mathtt{y}) \quad \Gamma, \Gamma(\mathtt{z}) \sqcup \Gamma(\mathtt{y}) \vdash \mathtt{x:=1} \quad \Gamma, \Gamma(\mathtt{z}) \sqcup \Gamma(\mathtt{y}) \vdash \mathtt{x:=2}}{\Gamma, \Gamma(\mathtt{z}) \vdash \mathtt{if\ y>0\ then\ x:=1\ else\ x:=2}} \quad \Gamma, \Gamma(\mathtt{z}) \vdash \mathtt{x:=3}}{\Gamma, \mathrm{L} \vdash \mathtt{if\ z>0\ then\ \{if\ y>0\ then\ x:=1\ else\ x:=2\}\ else\ \{x:=3\}}}$$

If-rule (example)

$$
\dfrac{
\Gamma \vdash \mathtt{z>0} : \Gamma(\mathtt{z})
\qquad
\dfrac{
\Gamma \vdash \mathtt{y>0} : \Gamma(\mathtt{y})
\qquad
\Gamma, \Gamma(\mathtt{z}) \sqcup \Gamma(\mathtt{y}) \vdash \mathtt{x:=1}
\qquad
\Gamma, \Gamma(\mathtt{z}) \sqcup \Gamma(\mathtt{y}) \vdash \mathtt{x:=2}
}{
\Gamma, \Gamma(\mathtt{z}) \vdash \mathtt{if\ y>0\ then\ x:=1\ else\ x:=2}
}
\qquad
\Gamma, \Gamma(\mathtt{z}) \vdash \mathtt{x:=3}
}{
\Gamma, \mathrm{L} \vdash \mathtt{if\ z>0\ then\ \{if\ y>0\ then\ x:=1\ else\ x:=2\}\ else\ \{x:=3\}}
}
$$

If-rule (example)

$$
\dfrac{
\dfrac{\Gamma(\mathbf{z}) \sqcup \Gamma(\mathbf{y}) \sqsubseteq \Gamma(\mathbf{x})}{\Gamma, \Gamma(\mathbf{z}) \sqcup \Gamma(\mathbf{y}) \vdash \mathtt{x:=1}}
\quad
\dfrac{\Gamma(\mathbf{z}) \sqcup \Gamma(\mathbf{y}) \sqsubseteq \Gamma(\mathbf{x})}{\Gamma, \Gamma(\mathbf{z}) \sqcup \Gamma(\mathbf{y}) \vdash \mathtt{x:=2}}
\quad
\dfrac{\Gamma(\mathbf{z}) \sqsubseteq \Gamma(\mathbf{x})}{\Gamma, \Gamma(\mathbf{z}) \vdash \mathtt{x:=3}}
}{
\begin{array}{l}\Gamma, \mathrm{L} \vdash \mathtt{if\ z>0\ then\ \{if\ y>0\ then\ x:=1\ else\ x:=2\}} \\ \qquad\qquad \mathtt{else\ \{x:=3\}}\end{array}
}
$$

If-rule (example)

$$\dfrac{\dfrac{\Gamma(\mathbf{z}) \sqcup \Gamma(\mathbf{y}) \sqsubseteq \Gamma(\mathbf{x})}{\Gamma, \Gamma(\mathbf{z}) \sqcup \Gamma(\mathbf{y}) \vdash \mathtt{x:=1}} \quad \dfrac{\Gamma(\mathbf{z}) \sqcup \Gamma(\mathbf{y}) \sqsubseteq \Gamma(\mathbf{x})}{\Gamma, \Gamma(\mathbf{z}) \sqcup \Gamma(\mathbf{y}) \vdash \mathtt{x:=2}} \quad \dfrac{\Gamma(\mathbf{z}) \sqsubseteq \Gamma(\mathbf{x})}{\Gamma, \Gamma(\mathbf{z}) \vdash \mathtt{x:=3}}}{\Gamma, \mathrm{L} \vdash \mathtt{if\ z>0\ then\ \{if\ y>0\ then\ x:=1\ else\ x:=2\}\ else\ \{x:=3\}}}$$

What is the relation between $\Gamma(\mathbf{x})$, $\Gamma(\mathbf{y})$, and $\Gamma(\mathbf{z})$, such that the above judgement can be proved?

If-rule (example)

$$\frac{\Gamma(\mathbf{z}) \sqcup \Gamma(\mathbf{y}) \sqsubseteq \Gamma(\mathbf{x})}{\Gamma, \mathrm{L} \vdash \texttt{if z>0 then \{if y>0 then x:=1 else x:=2\} else \{x:=3\}}}$$

while-rule

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

Sequence rule

$$\frac{\Gamma, ctx \vdash \mathtt{c1} \qquad \Gamma, ctx \vdash \mathtt{c2}}{\Gamma, ctx \vdash \mathtt{c1\,;\,c2}}$$

Sequence rule (example)

$$\frac{\dfrac{\Gamma, \ell \sqcup \Gamma(\mathtt{e}) \vdash \mathtt{x:=1} \qquad \Gamma, \ell \sqcup \Gamma(\mathtt{e}) \vdash \mathtt{x:=2}}{\begin{array}{l}\Gamma, \ell \vdash \mathtt{if\ e\ then\ \{x:=1\}} \\ \qquad\qquad\quad \mathtt{else\ \{x:=2\}}\end{array}} \qquad \Gamma, \ell \vdash \mathtt{x:=3}}{\Gamma, \ell \vdash \mathtt{if\ e\ then\ \{x:=1\}\ else\ \{x:=2\};\ x:=3}}$$

Theorem

Type correctness $\Rightarrow$ Noninterference

Upcoming events

- [May 10] A6 due
- [May 18] Final exam

A type system is the most cost effective unit test you'll ever have. – Peter Hallam