Information flow challenge

Challenges to understand (limits of) type system for IFC

- defeat type systems intended to prevent different kinds of leakage,
- https://ifc-challenge.appspot.com/

Leaks arising from different language constructs

- exceptions,
- memory references
- arrays
- timing and non-termination

Exceptions

Language constructs

```
try <stmt> catch <stmt>

throw;
```

Type system:

$$pc \vdash \texttt{skip} : low \qquad \frac{\vdash e : \ell \qquad \ell \sqcup pc \sqsubseteq \Gamma(x)}{pc \vdash x = e; \; : low} \qquad \frac{pc \vdash c_1 : \ell_1 \qquad pc \vdash c_2 : \ell_2}{pc \vdash c_1; c_2 : \ell_1 \sqcup \ell_2}$$

$$\frac{\vdash e : \ell \qquad \ell \sqcup pc \vdash c_1 : \ell_1 \qquad \ell \sqcup pc \vdash c_2 : \ell_2}{pc \vdash \texttt{if}\ e\ \texttt{then}\ c_1\ \texttt{else}\ c_2 : \ell_1 \sqcup \ell_2} \qquad \frac{\vdash e : low \qquad pc \vdash c : \ell}{low \vdash \texttt{while}\ e\ \texttt{do}\ c : \ell}$$

$$pc \vdash \texttt{throw} : pc \qquad \frac{pc \vdash c_1 : \ell_1 \qquad pc \sqcup \ell_1 \vdash c_2 : \ell_2}{pc \vdash \texttt{try}\ c_1\ \texttt{catch}\ c_2 : \ell_2}$$

Memory references

Language constructs for declaring references with content security level.

Ex: `declare ref l : high ; a "low" reference with "high" content`

Type system:

$$pc \vdash \texttt{skip}$$

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

$$\frac{pc \vdash c_1 \qquad pc \vdash c_2}{pc \vdash c_1; c_2}$$

$$\frac{\vdash e : \ell \qquad \ell \sqcup pc \vdash c_1 \qquad \ell \sqcup pc \vdash c_2}{pc \vdash \texttt{if}\ e\ \texttt{then}\ c_1\ \texttt{else}\ c_2}$$

$$\frac{\vdash e : \ell \qquad \ell \sqcup pc \vdash c}{pc \vdash \texttt{while}\ e\ \texttt{do}\ c}$$

$$\frac{\Gamma(r) = [\ell_1]_{\ell_2} \qquad \ell_2 \sqcup pc \sqsubseteq \Gamma(x)}{pc \vdash x = \texttt{deref}(r);}$$

$$\frac{\Gamma(r) = [\ell_1]_{\ell_2} \qquad \vdash e : \ell \qquad \ell \sqcup pc \sqsubseteq \ell_1}{pc \vdash \texttt{ref}\ r = e;}$$

Arrays

```
declare array la : low;          /* declare security level of cells */
```

$$pc \vdash \texttt{skip}$$

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

$$\frac{pc \vdash c_1 \qquad pc \vdash c_2}{pc \vdash c_1; c_2}$$

$$\frac{\vdash e : \ell \qquad \ell \sqcup pc \vdash c_1 \qquad \ell \sqcup pc \vdash c_2}{pc \vdash \texttt{if}\ e\ \texttt{then}\ c_1\ \texttt{else}\ c_2}$$

$$\frac{\vdash e : \ell \qquad \ell \sqcup pc \vdash c}{pc \vdash \texttt{while}\ e\ \texttt{do}\ c}$$

$$\frac{\Gamma(a) = [\ell_1]_{\ell_2} \qquad \vdash e : \ell \qquad \ell \sqcup \ell_1 \sqcup \ell_2 \sqcup pc \sqsubseteq \Gamma(x)}{pc \vdash x = a[e];}$$

$$\frac{\Gamma(a) = [\ell_1]_{\ell_2} \qquad \vdash e_1 : \ell_1' \qquad \vdash e_2 : \ell_2' \qquad \ell_2' \sqcup pc \sqsubseteq \ell_1}{pc \vdash a.[e_1] = e_2}$$

Procedures

$$\Gamma, pc \vdash \texttt{skip}$$

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

$$\frac{\Gamma, pc \vdash c_1 \qquad pc \vdash c_2}{\Gamma, pc \vdash c_1; c_2}$$

$$\frac{\Gamma \vdash e : \ell \qquad \ell \sqcup pc \vdash c_1 \qquad \ell \sqcup pc \vdash c_2}{\Gamma, pc \vdash \texttt{if}\ e\ \texttt{then}\ c_1\ \texttt{else}\ c_2}$$

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

$$\frac{\Gamma[x \mapsto \ell_1, y \mapsto \ell_2], \Gamma, pc \vdash c}{\Gamma, pc \vdash \texttt{proc}\ p(\texttt{in}\ x : \ell_1, \texttt{out}\ y : \ell_2)\ c}$$

$$\frac{\Gamma, pc \vdash \texttt{proc}\ p(\texttt{in}\ x : \ell_1, \texttt{out}\ y : \ell_2)\ c \qquad \Gamma \vdash e : \ell_1' \qquad \ell_1' \sqsubseteq \ell_1 \qquad \Gamma(z) \sqsubseteq \ell_2}{\Gamma, pc \vdash p(e, z)}$$