CSC547: Type Systems for OO Languages

First-Order Imperative Calculi

Abadi and Cardelli, Chapter 11

Typing

Subject reduction
Typing

We just need to add type rules for let and clone, which are as expected:

(Val Clone) (where $A$ is an object type)
If $E \vdash a : A$
then $E \vdash \text{clone}(a) : A$

(Val Let)
If $E \vdash a : A$
and $E, x : A \vdash b : B$
then $E \vdash \text{let } x = a \text{ in } b : B$

How can we type-check:

\[
let x = [ \text{foo=0} ] \text{ in } \\
\text{clone } (x).\text{foo}
\]
**Subject reduction**

In order to state subject reduction, we need some notation for type-checking a heap.

Write $E \models H$ when $E$ gives the types of the pointers in heap $H$.

What is the type of the heap:

- fred $\mapsto$ [ first = "Fred", last = "Flintstone" ],
- wilma $\mapsto$ [ first = "Wilma", last = "Flintstone" ],
- both $\mapsto$ [ fst = fred, snd = wilma ]

What about:

- bar $\mapsto$ [ a = foo ]
- foo $\mapsto$ [ b = fold(Baz,bar) ]

where $\text{Baz} \triangleq \mu(B)[ a : [ b : B ] ]$
Subject reduction

The type rule for heaps:

\[(\text{Heap Typing}) \text{ (when } E \equiv p_1 : A_1, \ldots, p_n : A_n)\]
If \( E \vdash O_1 : A_1 \)
... and \( E \vdash O_n : A_n \)
then \( E \models p_1 \mapsto O_1, \ldots, p_n \mapsto O_n \)

Write \( \models H \mid a : A \) when \( a \) has type \( A \) in heap \( H \):

\[(\text{Heap Val})\]
If \( E \models H \)
and \( E \vdash a : A \)
then \( \models H \mid a : A \)

We can now state subject reduction:

**Proposition (Subject reduction)**

If \( \models H \mid a : A \) and \( H \mid a \rightarrow H' \mid a' \) then \( \models H' \mid a' : A \).
Next 2 weeks

Jump into reading research papers...

Next week: *Featherweight Java* by Atsushi Igarashi, Benjamin Pierce and Philip Wadler.