Section 5:

Types and Programming Languages

CS 164 @ UC Berkeley, Spring 2024

Reminders

WA 2 is due on Friday, Feb 23

PA2 is released, and is due on March 18 at 11:59 PST.

Start early!!

Reminder to take care of yourselves, and to prioritize your health! WAs are worth 5% of your grade so don't stress too much about them!

Free Variables

Variables which are not defined in a given expression

```
def f() -> int:
    return x + y


def f(x : int, y : int) -> int:
    return g(x + y)
```

Type Environments

Map free variables to their types

```
x : int = 24                          {x : int}

def f(y : int) -> bool:

    z : int = 0                       {??}

    return y > z

f(x)                                  {??}
```

Type Environments and Functions

Entries for functions contain argument and return types, as well as types for local variables:

```
x : int = 24

def f(y : int) -> bool:
    z : int = 0
    return y > z
f(x)
```

```
{x : int}

{x : int,
 f : {int -> bool
 y : int,
 z : int }}
```

Type Inference Rules

Hypotheses above the bar imply the conclusion below the bar

“O ⊢ … v : T” is read as

“Under the scope with type environment O, v can be proven to have type T”

$$\frac{O \vdash e_1 : \text{int} \qquad O \vdash e_2 : \text{int}}{O \vdash e_1 + e_2 : \text{int}} \; [\text{add}]$$

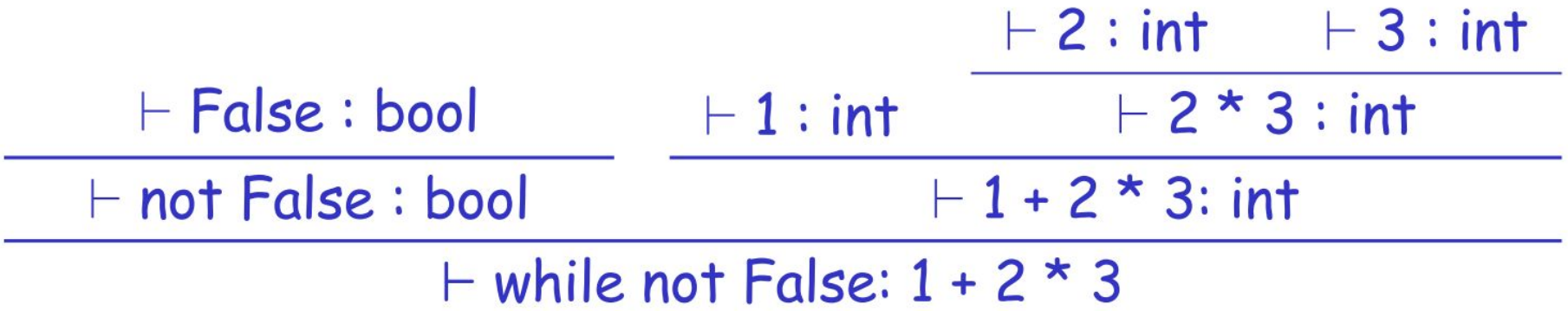
Type Inference Tree Proofs

Full expression at root of tree

Each subtree proves the hypotheses needed to satisfy rule

Each subtree proves the type of a subexpression of the full expression

(Somewhat) akin to top-down parsing



Anonymous feedback form:
http://tinyurl.com/SoraDisFeedback