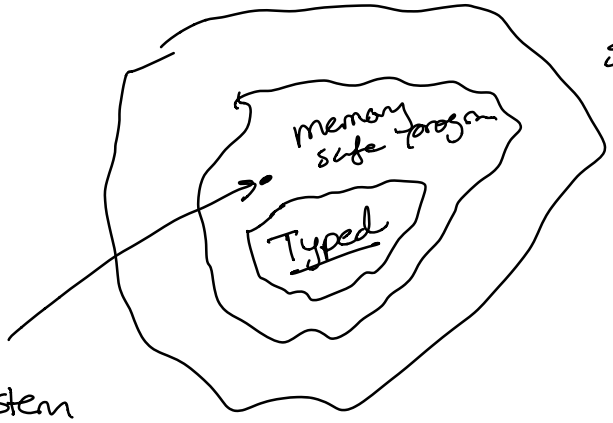
TYPES

$f: \mathbb{R} \to \mathbb{R}$

types ensure programs have desirable properties



space of all possible programs

type system can be overly conservative

Type checking program P, type T $\xrightarrow{\text{check}}$ P has type T?

Type inference given program P $\xrightarrow{\text{infer}}$ P has type T?

P "inhabits" type T

$\mathbb{Z}$ $\mathbb{R}$ $1/\pi \in \mathbb{R}$

λ calc subset

0 , succ 0 , iszero n , pred n , tru, fls

if true then t_1 else t_2 ,

Boolean value

operational semantics ≡ β reduction

define semantics

pred 0 → 0

iszero 0 → true

iszero (succ n) → false

pred (succ n) → n

if true then t_1 else t_2 → t_1

if false then t_1 else t_2 → t_2

$$\frac{t_1 \rightarrow t_1'}{\text{if } t_1 \text{ then } t_2 \text{ else } t_3 \rightarrow \text{if } t_1' \text{ then } t_2 \text{ else } t_3}$$

$$\frac{t \longrightarrow t'}{\text{succ } t \longrightarrow \text{succ } t'}$$

$T ::= \text{Bool} \mid \text{Nat}$ (OR)

$t : T$ equiv. $t \in T$

$t : T$ — more prevalent

$0 : \text{Nat}$
$\text{true} : \text{Bool}$
$\text{false} : \text{Bool}$
} Base cases

typing rules

$$\frac{t : \text{Nat}}{\text{iszero } t : \text{Bool}}\ T_{\text{iszero}}$$

same type

$$\frac{t_1 : \text{Bool} \quad t_2 \in T \quad t_3 \in T}{\text{if } t_1 \text{ then } t_2 \text{ else } t_3 \in T}\ T_{\text{if}}$$

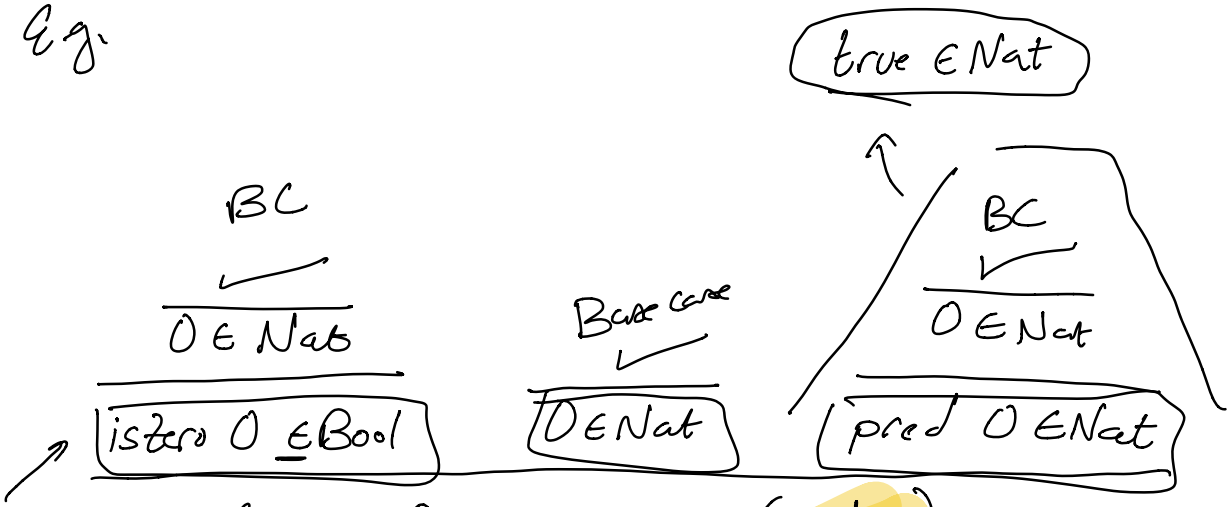
$$\frac{t \in \text{Nat}}{\text{succ } t \in \text{Nat}}\ T_{\text{succ}}$$

$$\frac{t \in \text{Nat}}{\text{pred } t \in \text{Nat}}\ T_{\text{pred}}$$

A term t is typable / well-typed if there is a type T s.t $t : T$

A typing relation is the smallest binary relation between terms and types satisfying our rules.

Eg.



Theorem (Uniqueness)

Each term t has at most 1 type

Proof

Base cases

$0 \in Nat$, $true \in Bool$, $false \in Bool$ ✓

Hypothesis

any term of size $\leq n$ has at most 1 type

Inductive step

take term of size $n+1$

assume term has type T

case 1 $succ\ t_1 \in T$

$$\frac{t_1 \in Nat}{succ\ t_1 \in T} \quad \text{o/o} \quad T = Nat$$

t_1 has at most 1 type

case 2

$$\frac{t_1 \in Bool \quad t_2, t_3 \in T}{\text{if } t_1 \text{ then } t_2 \text{ else } t_3 \in T}$$

Safety = Progress + preservation

≡ Soundness

well-typed terms don't "get stuck"

Progress: A well-typed term is not stuck
(either it is a value or can go one more step of evaluation)

Preservation: if a well-typed term takes a step of evaluation then the result is well-typed

---

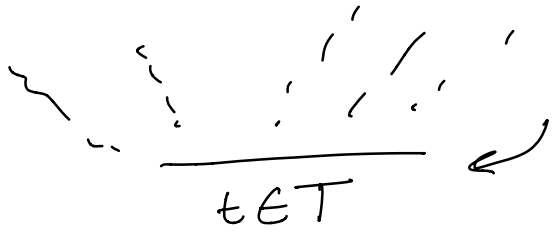
Thm (Progress)

Suppose $t \in T$. Then either t is a value or $t \to t'$

Thm (Preservation)

if $t \in T$ and $t \to t'$, $t' \in T$

Proof of preservation

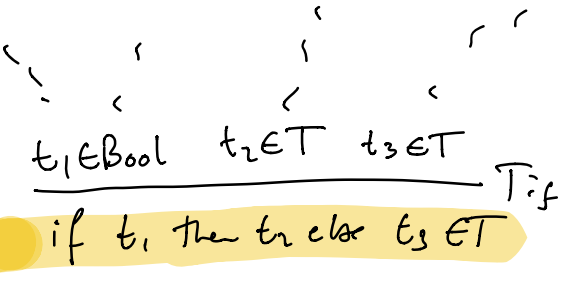


BC Base case rules

t = true or false or 0

theorem holds vacuously because $t \not\to t'$

Induction



Case 1 $t_1 = true$

$t' = t_2$

and we know $t_2 \in T$ so $t' \in T$

Case 2 $t_1 = false$

$t' = t_3$

we know $t_3 \in T$ so $t' \in T$

Case 3 $t_1 \longrightarrow t_1'$ By inductive hypothesis $t_1' \in Bool$

this means

$$\frac{t' = \text{if } t_1' \text{ then } t_2 \text{ else } t_3 \in T}{\text{if true then (succ 1) else true} \in \text{Nat}}$$

$$\frac{t_1 \in T}{\text{if true then } t_1 \text{ else } t_2 \in T}$$