Simply Typed A-cakulus

Function Types

Syntax of Types:

T:= Bool T->T

Syntax of Terms

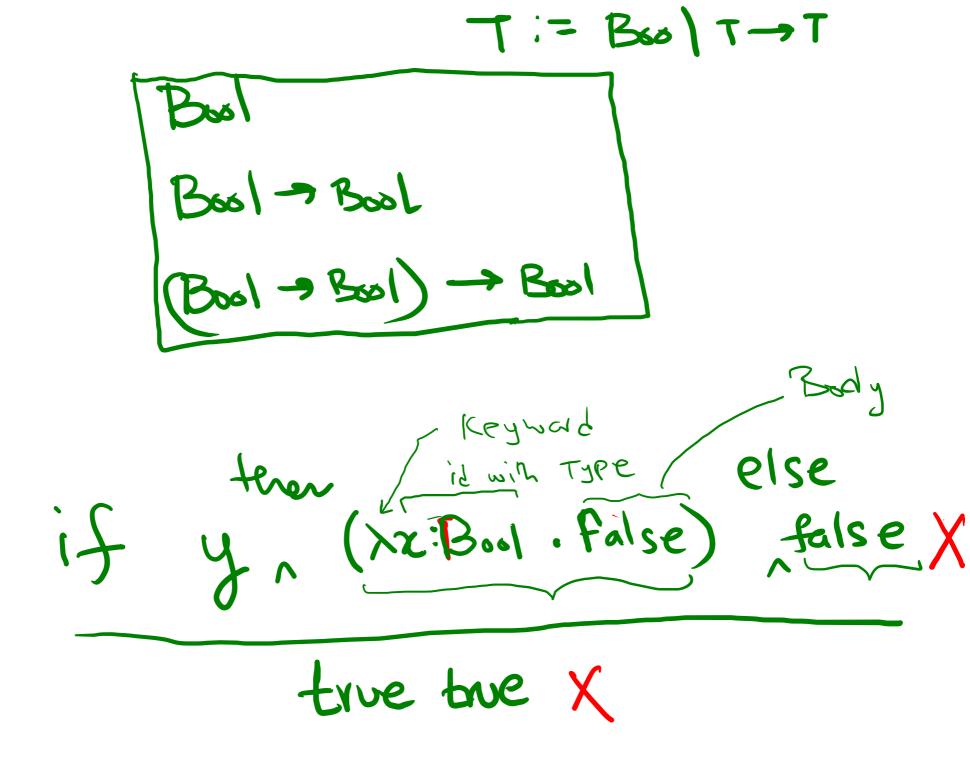
t:= z/\x:T.t/tt/b

b := true | false

と、て

XX: Bool. X: Bool-Bool

Tr: Bool (y) 7:



Type Environments ((ontexts) 「二 を3 「「, ス: 丁 T is a map from identifiers to Types.

y: Bool→Bool - Ax: Bool yx: Bool y Bool Bool

「一七:T

Well Typing Judgements

An inductive defin of well-typings for STLC (with if)

Typing Rules:

x:TET (T-VAR)
Thz:T

T- b: Bool

Abstraction:

y & dom(F)

The titz: Tiz

application

The tail to the titz to th

THE to Bool THE2: T TI-t3:T (T-IF)

THE LI Ken to else to: T

expression

Type

supluse: P= x: Bool, y: Bool -> Bool

Mr + false : Bool

using T-Boul 1. M- false: Book

(2) 1 - z: Bool

1. 25:13 od, 2:13001 - 2:13001 Using (T-VAR)

alternatively: 2: Bool ET 3 :. P+x: Bool 3 - (TUAF)

Ttstily x: Bool

1. 7 + y: Bool→Bool

using (T.VAR)

: y: Bool = Bool

2. M + x: Bool

USING (TVAR) on x.

3. M/ yz: Bool

hom 1, 2, using (TAPP)

Prove

M- λz: Bool. x: Bool → Bool

1. [7, Z:Bool - x:Bool Using (T-VAR)

2. M- \Z: Bool. z: Bool - Boil

orgme veryp.

Show M- if x then (yx) else x: Bool

1. The Z: Bool Using T-VAR

2. Thy: Bool - Bool using T-VAR

3. 17 - 7x: Bool

4. if x then yx) else z: Bool from 1,3,1 using
T-IF