CMPS 231 HW5

***Part A***

1. VA = Variable Axiom

Γ = {x: bool, y:int} FA = Function Application

-------------------------------------- **(VA)** AF = Anonymous Function Rule

x: bool ∈ Γ Sub = Subtyping

LC /Or = Logical Connectives Rule

Γ = {x: bool, y:int} ICA = Integer Constant Axiom

-------------------------------------- **(VA)** BCA = Boolean Constant Axiom

y: int ∈ Γ

Γ = {x: bool, y:int}

-------------------------------------- **(Sub)**

y - 5: int ∈ Γ

Γ = {x: bool, y:int}

-------------------------------------- **(Sub)**

y - 5 < 7: bool ∈ Γ

Γ = {x: bool, y:int}

-------------------------------------- **(Or)**

x || (y - 5 < 7): bool ∈ Γ

(2)

Γ = {}

--------------------------- **(AF)**

f: α1 -> α2 ∈ Γ |- f: α1 -> α2

Γ = {f: α1 -> α2}

--------------------------- **(AF)**

x: bool ∈ Γ, f: α1 -> α2 |- x: bool

Γ = {f: α1 -> α2}

--------------------------- **(VA)**

x: bool ∈ Γ, f: α1 -> α2 |- if x then f true else 19: α2

Γ = {f: bool -> int}

--------------------------- **(AF)**

f: bool -> int ∈ Γ |- f: bool -> int

Γ = {f: bool -> int}

--------------------------- **(FA)**

(f true): int ∈ Γ

Γ = {f: bool -> int}

--------------------------- **(VA)**

19: int ∈ Γ

Γ = {}

--------------------------- **(AF)**

x: bool ∈ Γ |- (fn f => fn x => if x then f true else 19): (bool -> int) -> bool -> int

(3)

Γ = {}

--------------------------- **(AF)**

f: α1 -> α2 ∈ Γ |- f: α1 -> α2

Γ = {f:int -> bool}

--------------------------- **(AF)**

x: int ∈ Γ, f: int -> bool |- 5 < x: bool

Γ = {f:int -> bool}

--------------------------- **(Let)**

f: int -> bool ∈ Γ |- let fun f x = 5 < x in fn y => (f 3) || y end: bool -> bool

Γ = {}

--------------------------- **(AF)**

y: bool ∈ Γ |- y: bool

Γ = {}

--------------------------- **(FA)**

f 3: bool ∈ Γ

Γ = {}

--------------------------- **(Or)**

(f 3) || y: bool ∈ Γ

Γ = {}

--------------------------- **(AF)**

fn y => (f 3) || y: bool -> bool ∈ Γ

1. fn f => if true then fn y => f (y + 1) else fn z => false

f: bool -> u

true: bool

y: v

--------------------------------------

f: bool -> u |- y + 1: int **(ICA)**

--------------------------------------

f: bool -> u |- y + 1: v **(VA)**

--------------------------------------

f: bool -> u |- f (y + 1): u **(FA)**

--------------------------------------

fn y: int => f (y + 1): bool -> u

--------------------------------------

false: bool

--------------------------------------

z: v

--------------------------------------

f: bool -> u |- false: bool **(BCA)**

--------------------------------------

f: bool -> u |- false: v **(VA)**

--------------------------------------

f: bool -> u |- fn z => false: bool -> v **(AF)**

--------------------------------------

f: bool -> u |- fn z => false: v -> bool **(LC)**

--------------------------------------

f: bool -> u |- if true then fn y => f (y + 1) else fn z => false: v -> bool **(IF)**

--------------------------------------

f: bool -> (v -> bool) => if true then fn y => f (y + 1) else fn z => false: bool -> (v -> bool) -> bool

1. fn f => f (f && true)

f: bool -> u

--------------------------------------

f: bool -> u |- f && true: bool -> u **(LC)**

--------------------------------------

fn f: bool -> u => f (f && true): bool -> u