

# CS4022D Principles of Programming Languages

## Lecture #8: Operational Semantics

Saleena N  
NIT Calicut

August 13, 2021

# Language of Arithmetic Expressions

$t ::=$

true

false

if t then t else t

0

succ t

pred t

iszero t

*terms*

*constant true*

*constant false*

*conditional*

*constant zero*

*successor*

*predecessor*

*zero test*

# Language of Arithmetic Expressions: Terms / Programs

```
iszero succ 0  
if (iszero (pred 0)) then true else false  
true  
succ succ succ 0  
pred succ pred succ 0
```

# Language of Booleans

$t ::=$

true

false

if t then t else t

*terms*

*constant true*

*constant false*

*conditional*

# Language of Booleans

$t ::=$

true

false

if t then t else t

*terms*

*constant true*

*constant false*

*conditional*

true

false

if true then false else true

if (if true then true else false) then false else true

# Conditional

if (if true then true else false) then false else true

if  $t_1$  then  $t_2$  else  $t_3$

$t_1$  is ...

$t_2$  is ...

$t_3$  is ...

- Draw the AST representation of the term

# Language of Booleans: A Programming Language

- Each term is a program
  - true
  - if true then false else true
  - if (if true then true else false) then false else true

# Language of Booleans: Evaluation of programs

- Program evaluation in an abstract machine - a state transition machine
  - initial state: the term to be evaluated
  - state transitions: to be specified
  - final state: the result of evaluation



# Operational Semantics (small-step)

- As steps of evaluation in the abstract machine
- A set of evaluation rules of the form  $t \rightarrow t'$
- $t \rightarrow t'$  : machine in state  $t$  transforms state to  $t'$

# Language of Booleans: Values

$v ::=$

true

false

# Language of Booleans: Evaluation Rules

Evaluation:  $t \rightarrow t'$

$\text{if true then } t_2 \text{ else } t_3 \rightarrow t_2$  (E-IFTRUE)

$\text{if false then } t_2 \text{ else } t_3 \rightarrow t_3$  (E-IFFALSE)

$$\frac{t_1 \rightarrow t'_1}{\text{if } t_1 \text{ then } t_2 \text{ else } t_3 \rightarrow \text{if } t'_1 \text{ then } t_2 \text{ else } t_3}$$
 (E-IF)

# Evaluation Rules

*if true then  $t_2$  else  $t_3 \rightarrow t_2$*  (E-IFTRUE)

# Evaluation Rules

*if true then false else true*  $\rightarrow$  *false*

# Language of Booleans: Evaluation Rules

*if false then  $t_2$  else  $t_3 \rightarrow t_3$*  (E-IFFALSE)

# Language of Booleans: Evaluation

*if false then false else true*  $\rightarrow$  *true*

# Language of Booleans: Evaluation

*if false then false else (if true then true else false)*  
 $\rightarrow$  *if true then true else false*

*if true then true else false*  $\rightarrow$  *true*

- Evaluation completes in two steps
- Rules used in each step?



# Language of Booleans: Evaluation

*if false then false else (if true then true else false)*

$E-IFFALSE \longrightarrow$  *if true then true else false*

*if true then true else false*  $E-IFTRUE \longrightarrow$  *true*

- Step1: instance of E-IFFALSE
- Step2: instance of E-IFTRUE

# Rule Schema

- E-IFTRUE, E-IFFALSE and E-IF are rule schema
- Each evaluation step uses an instance of the rule schema

# Evaluate

- if true then (if true then false else true) else true
- if false then true else false
- if (if true then true else false) then true else false

# Evaluate

- if (if true then true else false) then true else false
- *if  $t_1$  then  $t_2$  else  $t_3$*  with  $t_1 \rightarrow t'_1$
- instance of E-IF

# E-IF

$$\frac{t_1 \rightarrow t'_1}{\text{if } t_1 \text{ then } t_2 \text{ else } t_3 \rightarrow \text{if } t'_1 \text{ then } t_2 \text{ else } t_3} \quad (\text{E-IF})$$

*if true then  $t_2$  else  $t_3 \rightarrow t_2$*  (E-IFTRUE)

*if false then  $t_2$  else  $t_3 \rightarrow t_3$*  (E-IFFALSE)

Axioms

# Language of Booleans: Evaluation Rules

$$\frac{t_1 \rightarrow t'_1}{\text{if } t_1 \text{ then } t_2 \text{ else } t_3 \rightarrow \text{if } t'_1 \text{ then } t_2 \text{ else } t_3} \quad (\text{E-IF})$$

- Inference Rule - premise (precondition), conclusion
- Precondition to be satisfied inorder to apply the rule

# Evaluate

- if (if true then true else false) then true else false
- *if  $t_1$  then  $t_2$  else  $t_3$*  with  $t_1 \rightarrow t'_1$
- instance of E-IF
- subterm  $t_1$  to be evaluated using an instance of ...?