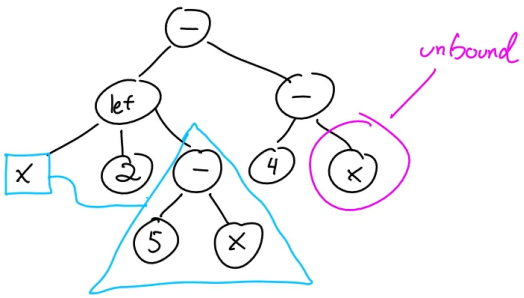
$(\text{let } x=2 \text{ in } 5-x)-(4-x)$

Auklet
Variable
Scoping



Bluebird

ONLY for Bluebird:

we will assume that programmer uses the right types

1 Syntax

```
<expr> ::= .....
         | true | false
         | <expr> && <expr> | <expr> || <expr> | !<expr>
         | <expr> = <expr>
         | if <expr> then <expr> else <expr>
         | isbool(<expr>) | isint(<expr>)
```

2 Semantics

if true then 4 else 3 ⟹ 4

if 2=1 then 7 else false ⟹ false

isbool(true) ⟹ true

isbool(false) ⟹ true

isint(true) ⟹ false

isbool(isint(true)) ⟹ true

3 Representation

x64 — registers contain 64 bits*

2^{64} combinations

sign bit ↓

00000000 00000000 00000000 00000000 00000000 00000000 00000000 000000

#1: Do what C does.

true ≡ 1

false ≡ 0

isbool(1) ⟹ false

mov rax, 1

0x800…01

0b 1000 0000 0000 … 0000 0001

#2: Tagging

Auklet has 64-bit signed integers.

Bluebird has 63-bit signed integers

"binary representation"

2^{62} = 4,611,686,018,427,387,904

- Integers have a tag of ∅
  - BB 4 = machine 0b00 — 01000 = machine 8
  - BB 3 = machine 0b00 — 00110 = machine 6
- Booleans have a tag of 1
  - BB true = machine 0x80 — 01
  - BB false = machine 0x00 — 01

  } lecture-specific
- $2^{63}-2$ unused

Bluebird

AST → ASM ?

after(3) ⟹ mov rax, 6 ≅ [mov rax, 3 / sal rax, 1]

add rax, 2 ≅ add rax, 2

BB machine

$R(n) = n * 2$

```
mov rax, 10   ; BB 5
mov [rsp-8], rax
mov rax, 6    ; BB 3
mov [rsp-16], rax
mov rax, [rsp-8]
add rax, [rsp-16]  ; BB 8
```

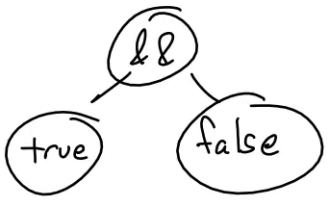
$$R(x) + R(y) = 2x + 2y = 2(x+y) = R(x+y)$$
$$R(x) - R(y) = 2x - 2y = 2(x-y) = R(x-y)$$
$$R(x) * R(y) = 2x * 2y = 4xy = R(xy) \cdot 2$$

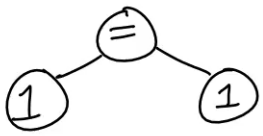
$a \cdot b$ $\quad \frac{R(a)}{2} \cdot R(b)$

$2{,}000{,}000{,}000 \cdot 2{,}000{,}000{,}000$

$2^{62} - 1 = 4{,}611{,}686{,}018{,}427{,}387{,}903$

AST





ASM?

```
mov rax, 0x800.....01
mov [rsp-8], rax
mov rax, 0x00.....01
mov [rsp-16], rax
mov rax, [rsp-8]
and rax, [rsp-16]
```