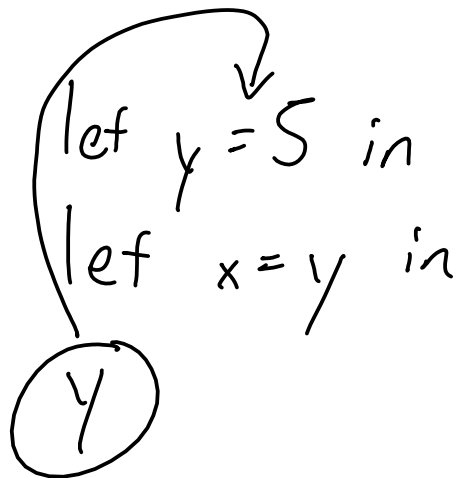


let $x = e_1$ in
 e_2

let $x =$
let $y = 2$ in $y + 1$
in
 $x + 2$

let $x =$
let $y = 5$ in y
in
 y

let $y = 2$ in
let $x = y + 1$ in
 $x + 2$



Cobra

$\langle \text{expr} \rangle ::=$
if $\langle \text{expr} \rangle$ then $\langle \text{expr} \rangle$ else $\langle \text{expr} \rangle$.

- 0 is false
- booleans are different

if cond then else

if safe() and obj.sound() then

Cobra: Binary Representation

let x:bool = 5 in

type i_expr =
| IVar of string
| IInt of int
| IBool of bool

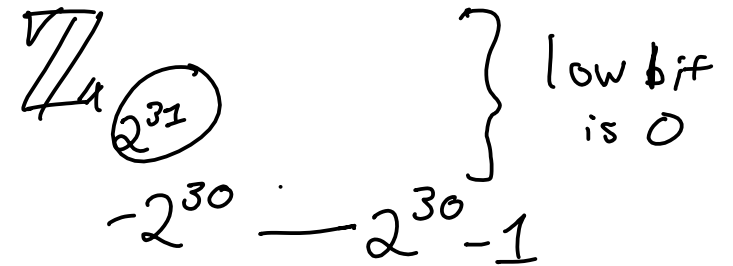
compile_i_expr ie =
match ie with
| IVar x →env.....
| IInt n →
| IBool b →

0x00000000

0x00000001 false } low bit
0x80000001 true } is 1

Use more registers!

- eax is # register
- ebx is bool register



0x00000004

2 0000 0000 0000 0000 0000 0000 0100

CAdd(ie1, ie2) →

let a1 = compile_i_expr ie1 in
let a2 = compile_i_expr ie2 in

[XMov (XRegister EAX), a1;
XAdd (XRegister EAX), a2]

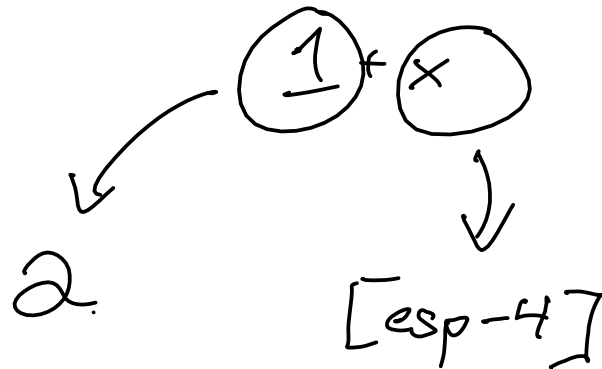
mov eax, 2
add eax, [esp+4]

In
Cobra

$x+y \Rightarrow z$

In
Repr

$2x+2y \Rightarrow 2z$



Subtraction: add 1?

okay? change negation?

Multiplication: div 2?

Division: mult 2?

1000000 * 1000

2000000 * 2000

$I_n^{\text{Cobra}} \quad x * y \Rightarrow z$

$I_n^{\text{Repr}} \quad \left(\frac{2x}{2}\right) * 2y \Rightarrow 2z \quad \text{shr eax, 1}$

$I_n^{\text{Cobra}} \quad x - y \Rightarrow z$

$I_n^{\text{Repr}} \quad 2x - 2y \Rightarrow 2z$

$\langle \text{expr} \rangle ::= \dots$

| true

| false

| $\langle \text{expr} \rangle < \langle \text{expr} \rangle$

if $x < y$ then e_1 else e_2

let $\$var1 = x < y$ in

if $\$var1$ then e_1 else e_2

```
cmp eax, [esp-4]
```

```
jge else4
```

```
mov eax, 0x80000001
```

```
jmp end5
```

else4:

```
mov eax, 0x00000001
```

end5:

0x80000001

0x00000001

```
mov eax, first arg
```

```
sub eax, second arg
```

```
and eax, 0x80000000
```

```
or eax, 0x00000001
```