So far:

Avklet — arithmetic
Bluebird — binary representations, booleans
Cardinal — C calling conventions, function calls
Dove — declare functions, call functions, static analysis
Eagle —

---

# Hawk

## Syntax

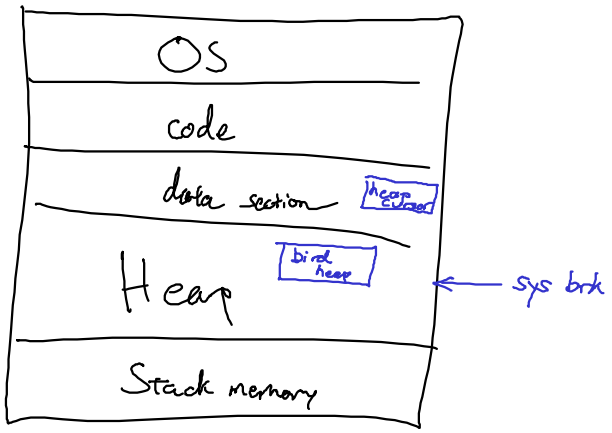
```
<expr> ::= ... | (<expr>, <expr>)
             | fst (<expr>)
             | snd (<expr>)
```

## Semantics

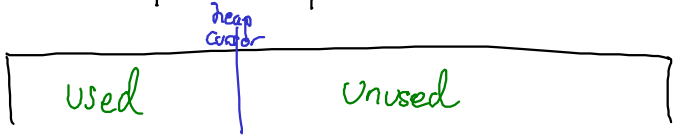
```
(3, 4) ⟹ (3, 4)
(1+2, 3-4) ⟹ (3, -1)
(7-1, 3+2) ⟹ (6, 5)
(true, 7+1) ⟹ (true, 8)
(3, true+1) ⟹ runtime error 1
(1, (2, 3+4)) ⟹ (1, (2, 7))
fst((4,3)) ⟹ 4
snd((1+true, 3)) ⟹ runtime error 1
fst(false) ⟹ runtime error 4
fst(((1,2),3)) ⟹ (1,2)
```

## we don't have

exceptions, error handling
dynamic memory
floats
strings
lists / arrays
static types
imperative loops



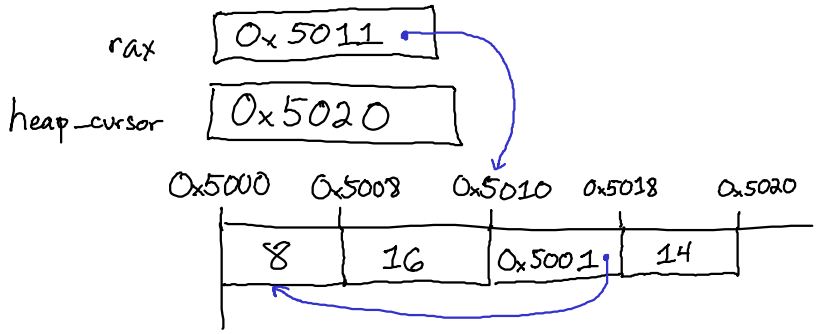

1. Driver allocate memory (4Mb)
2. Store pointer to unused memory in heap-cursor

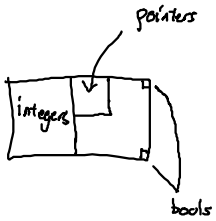



3. We will not clean up memory
4. We will not allocate more memory
5. If we use too much memory, unspecified behavior

bird 10,240

((4, 8), 7)




| bird values | binary repr | |
|---|---|---|
| 0xNNNN NNNN NNNN [nnn] | 0xNNNNNNNN NNNN NNN [nnn0] | integers |
| true | 0xFFFF ..... FF [1111] | booleans |
| false | 0x7FFF ..... FF [1111] | |
| pointer to 0xNNNN ..... NNN [n000] | 0xNNNN ..... NN [n001] | pointers |

```
.section data
    align 8
    heap_cursor:
    dq 0
.section text
bird_main:
    push rbp
    ⋮
    mov [heap_cursor], rdi
```

each is one byte

```
db 7, 8, 2
dw 5
dd
dq 0
```

data : byte
data : word
data : double word
data : quad word