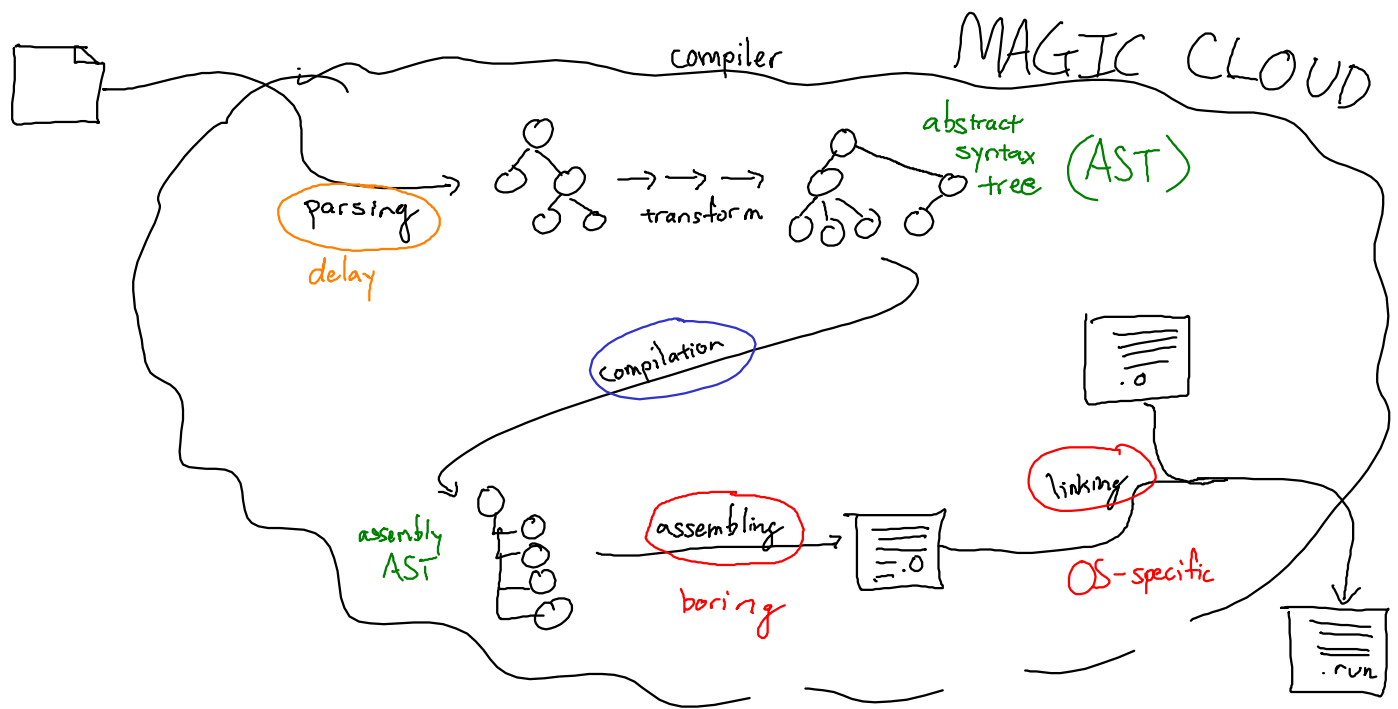




Auklet

1. Syntax

English

✓ There is snow outside.
✗ Snow there is is is snow i's

O Caml

let n=1 in n+2
= let + + in 2 1 let

semi-formal ~EBNF

$\langle \text{expr} \rangle ::= 0 \mid 1 \mid -1 \mid 2 \mid -2 \mid \dots$

$\mid \langle \text{expr} \rangle + \langle \text{expr} \rangle \mid \langle \text{expr} \rangle - \langle \text{expr} \rangle \mid \langle \text{expr} \rangle * \langle \text{expr} \rangle$
 $\mid (\langle \text{expr} \rangle) \mid \langle \text{ident} \rangle \mid \text{let } \langle \text{ident} \rangle = \langle \text{expr} \rangle \text{ in } \langle \text{expr} \rangle$
 $\mid \text{after } (\langle \text{expr} \rangle) \mid \text{before } (\langle \text{expr} \rangle)$

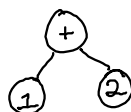
not in Auklet

function	division
lists	comparators
pointers	printing
types	
if	
boolean	
float	
loops	
imports	

Auklet source

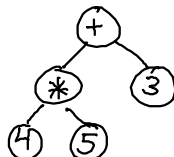
1+2

AST



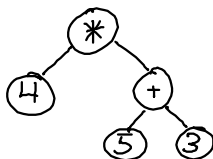
3

4*5+3



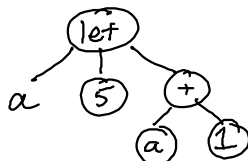
23

4*(5+3)



32

let a=5 in a+1



6

after (4)



5

2. Semantics

What a program means.

Compiling is translating to a different language while preserving meaning.
 ↙ there is convenient for computer

→ mov %rbx, %rcx

mov 8(%rbx), %rax

→ mov rcx, rbx

mov rcx, [rbx+8]

AST

assembly

Compilation will translate
Autdef AST into a sequence
of instructions that leave the
result in RAX

4

```
mov rax, 4
```

after

```
mov rax, 4
```

4

add rax, 1

after

```
add rax, 1
```

type expr =
| EInt of int
| EAfter of expr
variant types

CCam

let rec compile_expression (e : expr) : instruction list =

match e with

④ $EI_{nt} \quad 4$

$$|E|_{\text{Int}}(n) \rightarrow$$

[Asm Mov (Arg Register (RAX), Arg Constant (n))] ← mov rax, 4

1) $E_{After}(e^i) \rightarrow$

compile_expression e' @ [AsmAdd (ArgRegister(RAX),
ArgConstant(1))]



after

T