

Type Systems

What is a "type system"?

What is a "type"?

What is a "set"?

Static Program Analysis

A system which performs static checks to prove something about your program:

A type is a kind of data.

A type is a set of values.

A set is an unordered collection of elements.

{ 4, biscuit, ☁ }

C: `int x = 5;` } `int x;` ← "Create a variable called 'x'.
Find enough memory for it.
I promise I will only ever put an int in here."

`<expr> ::= ...`

| `let <identifier> : <type> = <expr> in <expr>`

```

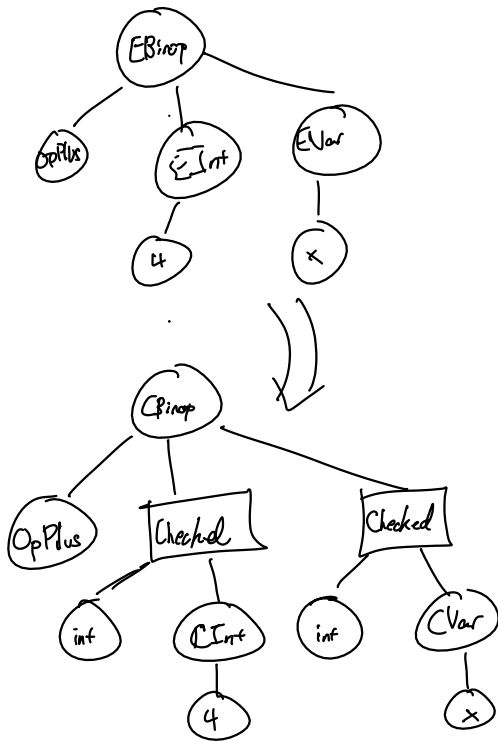
let x : int = 4 in
let y : int = 8 in
x + y

```

```

mov eax, 8
mov [ebp-4], eax
mov eax, 16
mov [ebp-8], eax
mov eax, [ebp-4]
add eax, [ebp-8]

```



```
def f(x, y) :  
  if x then y+1 else (if y then false else true)  
end
```

```
let a = f (true, 3) in  
let b = f (false, true) in  
...
```

Cannot type using
the simple types we
discussed

```
let x : unknown = ... in  
  ⋮  
let y : int = x in  
  ⋮
```

Gradual typing

```
x = 5;  
while (b) {  
  f(x);  
  x = true;  
  ⋮  
}
```