

(5, true)

int * bool

let t = (1, 2) in

t[0] := false

Advantages to typechecking

- * Helps compiler (performance)
- * Helps programmer (debugging)

Disadvantages

- * Limits programs

With new values of 8 bytes in size, what problems arise? (Assume 32-bit word size.)

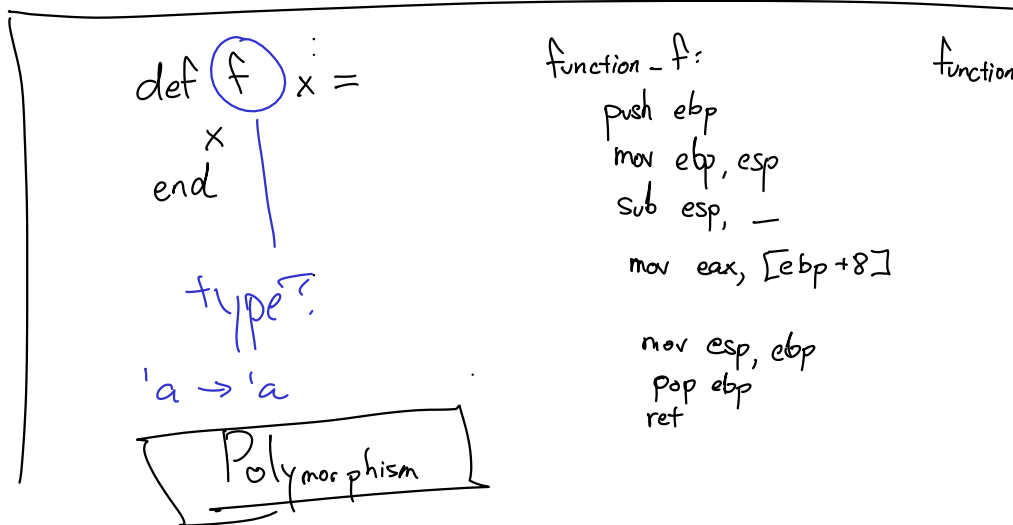
- Arithmetic ops?
- Storage? `eax, ecx`
- Stack allocation
- Heap allocation

$$\underbrace{2^{32}-1 + 2^{32}-1}_{\substack{\text{int?} \\ \text{long?}}} \stackrel{?}{=} 2^{32}-2 \stackrel{?}{=} -2$$



```
def f x =
  x + 1
end
def g y =
  f
end
```

$f : \text{int} \rightarrow \text{int}$



```
def f x =
  x
end
```

type?

'a -> a'

Polymorphism

```
function-f:
  push ebp
  mov ebp, esp
  sub esp, -
  mov eax, [ebp+8]

  mov esp, ebp
  pop ebp
  ret
```

function-f-for-big-things:

Compiling Polymorphism

- * Make one copy for each concrete type (templating)
- * Make every value be the same size (generic)