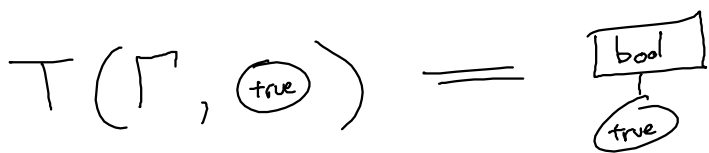
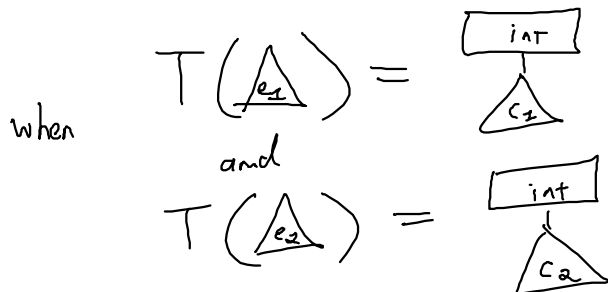
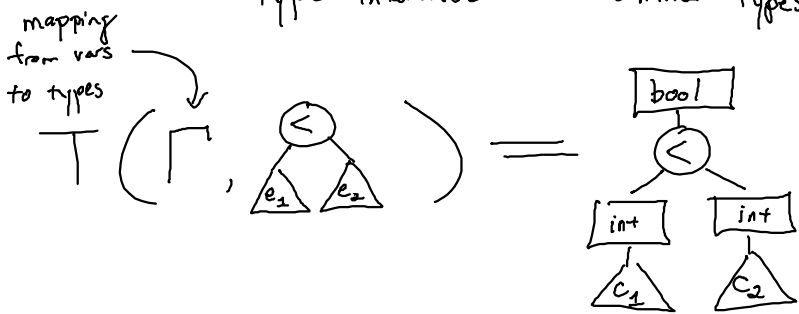
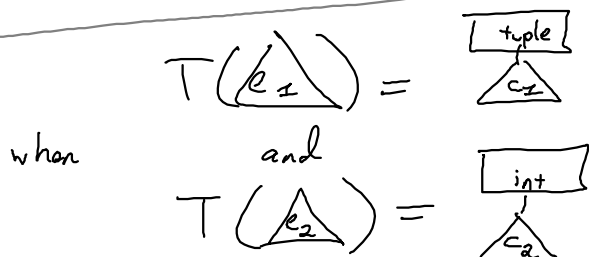
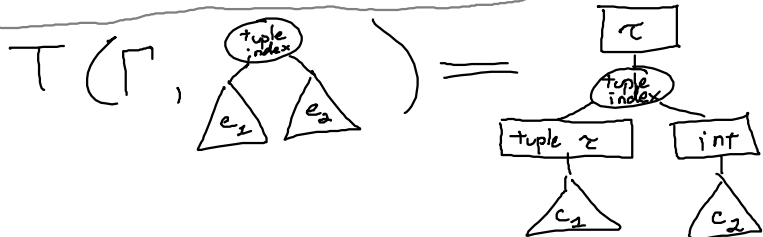
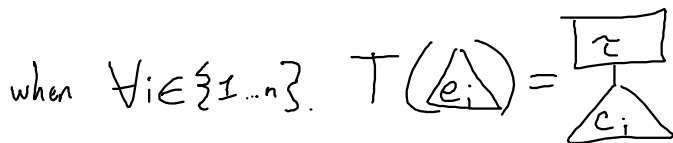
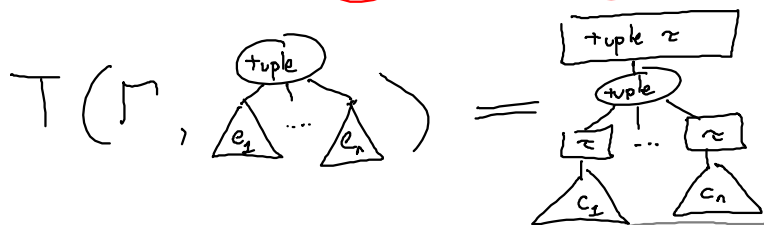
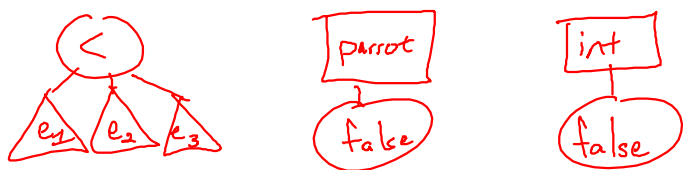


Type Systems

- Type — set of a values *
- Type checker — verifies types of each part of a program
- type inference — determines types



$\langle \text{expr} \rangle ::= \text{true} \mid \text{false} \mid \langle \text{expr} \rangle < \langle \text{expr} \rangle \mid$
 $\text{after}(\langle \text{expr} \rangle) \mid \dots$
 $\langle \text{type} \rangle ::= \text{bool} \mid \text{int} \mid \text{tuple } \tau$



OCaml

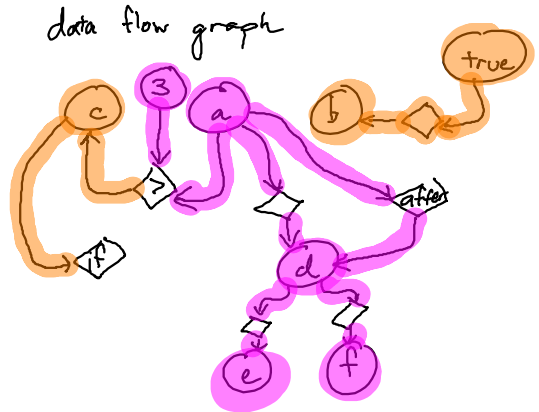
let (a,b) = pair in

Generally

fixed size (heterogeneous type)
fixed type (varying size)

let a = 4 in
 let b = true in
 let c = a > 3 in
 let d = if c then a else after(a) in
 let e = d in
 let f = d in

int
 bool

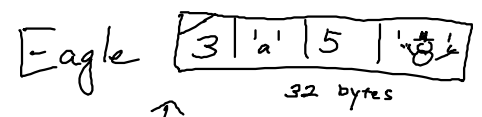
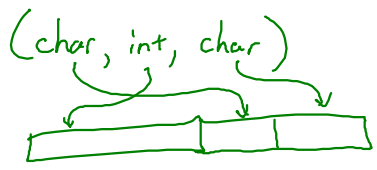


$\langle \text{type} \rangle ::= \text{int} \mid \text{bool} \mid \text{char} \mid (\langle \text{type} \rangle, \dots, \langle \text{type} \rangle)$
 (8 bytes for int, 4 bytes for char, UTF-32 for char)

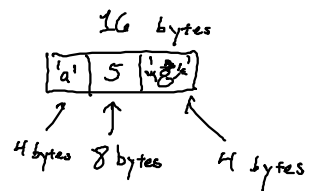
('a', 5, 'k') : (char, int, char)

programmer wants to store this data in this order
 (C)

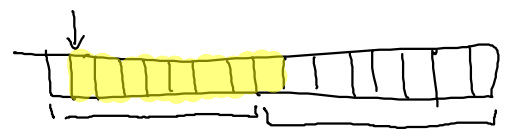
programmer just wants a triple
 (OCaml, ...)



Tongay

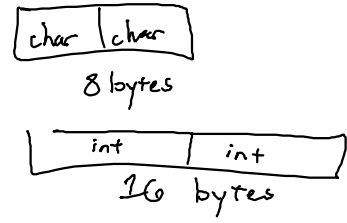


Processor's access to memory is by word



Polymorphism

↑ many ↑ shapes/forms



C don't
 C++ templization — make copies
 Java/Python/.. generics — one copy, universal form

```
def f(n)
  (n, n)
end
```

how do we compile this
 to allow $f(5)$ and
 $f(\frac{1}{8}, '')$