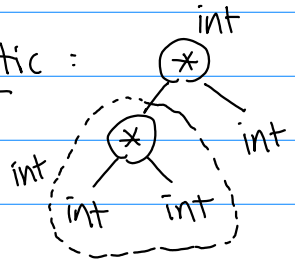
Type-Checking: 2 major categories

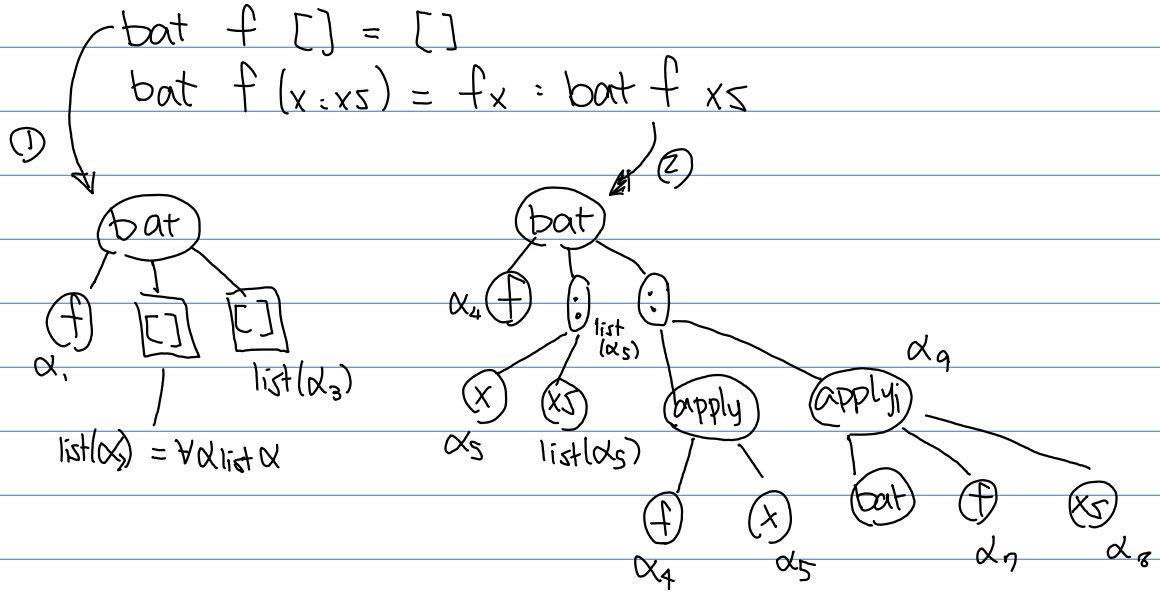
- Semantic:



- influential:
  type inference

Haskell

```
bat f []     = []
bat f (x:xs) = fx : bat f xs
```



':' takes $(\beta, (list(\beta)) \to list(\beta)$

$\alpha_6 \equiv list(\alpha_5)$

'apply' $(\alpha \to \beta, \alpha) \to \beta$

$\alpha_6 = list(\alpha_5)$ $\quad \alpha_7 = \alpha_5 \to \alpha_6$

$\alpha_4 = \alpha \to \beta$ $\quad \alpha_8 = list(\alpha_5)$

$\alpha_5 = \alpha$

$\alpha_4 = \alpha_5 - \alpha_6$

Q: What is the type of bat?

① bat is $\alpha_1 * list(\alpha_2) \Rightarrow list(\alpha_3)$

② bat : $(\alpha_5 \to \alpha_6) \times list(\alpha_5) \to \alpha_9$

$\Rightarrow list(\alpha_5 \to \alpha_6) \times list(\alpha_5) \to list(\alpha_6)$

Substitution: for each type variable in an type expression, consistently substitute a real type or some other type for each variable consistently substitute another type.

$\alpha_5 \to$ int $\alpha_5 \to$ int

$\alpha_6 \to$ float $\alpha_6 \to$ list (β)

Unification: given two type expressions, find a substitution on both that make the two expressions the same.

Substitution

$\alpha_1 \Rightarrow \alpha_1 \to \beta$

$\alpha_2 \Rightarrow \alpha$

$\alpha_3 \Rightarrow \beta$

$\alpha_5 \Rightarrow \alpha$

$\alpha_6 \Rightarrow \beta$

$(\alpha \to \beta) \times$ list $(\alpha) \to$ list (β)

$(\alpha \to \beta) \times$ list $(\alpha) \to$ list (β) ← verification

map e^x [1 2 3 4] $\to [e^1 e^2 e^3 e^4]$

int $\to$ float