

Type Systems

- ▶ related to typechecking
- ▶ ascribes to each part of a program a property called a "type"
- ▶ types include descriptions of data representations (type is a set of values)*
- ▶ a set is an unordered collection of unique values

$\{1, 2\}$ $\{3, 4, 5\}$

$\{3\}$ $\{6, 11\}$

$\{3, 3\} = \{3\}$

$\{3, 4\} = \{4, 3\}$

→ $\text{bool} = \{\text{true}, \text{false}\}$

$\text{int} = \{0, 1, -1, 2, -2, \dots, 2^{62}-1, -2^{62}\}$

bipartite tree

type cexpr =

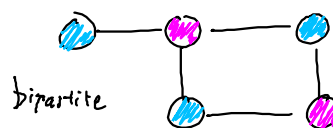
| CInt of int

| CAfter of checked

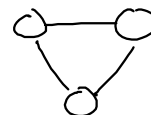
.....

and checked =

| Checked of cexpr * typ



not



$T: \text{expr} \rightarrow \text{checked}$

$T(5) =$

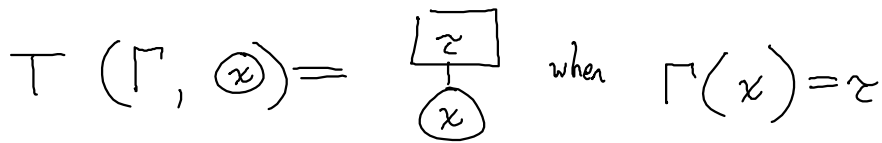
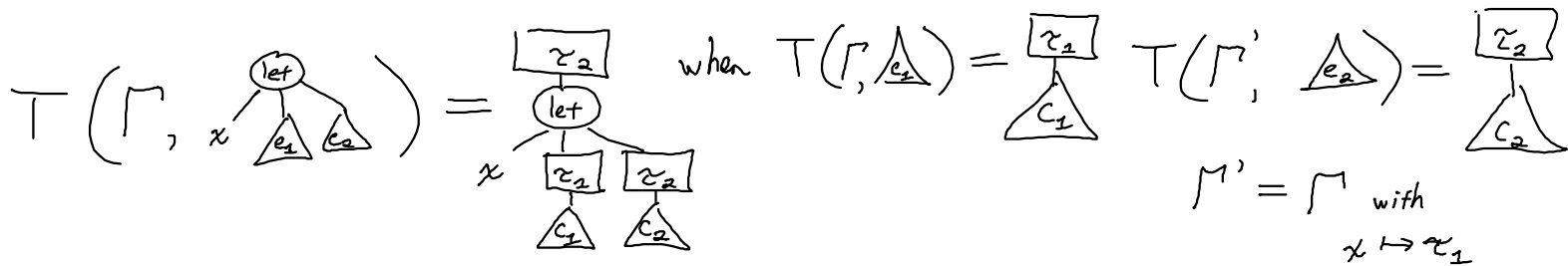
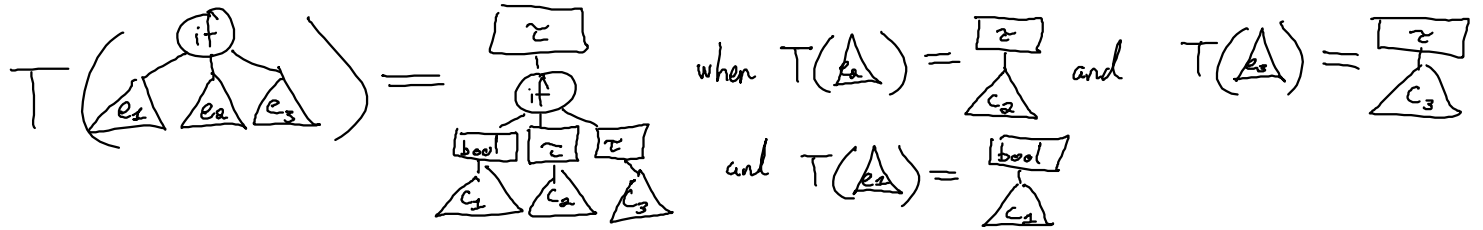
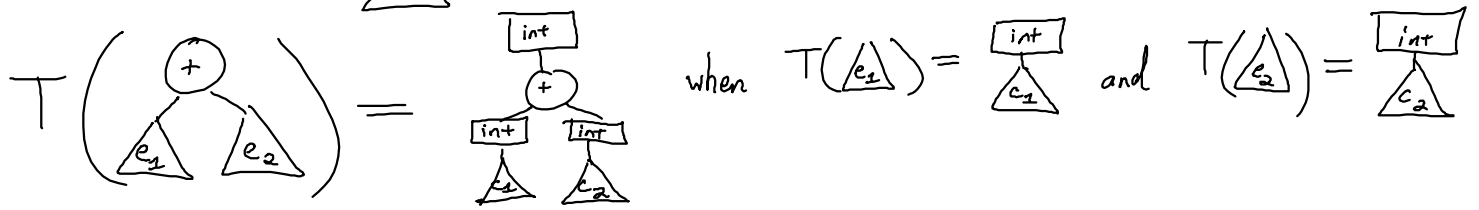
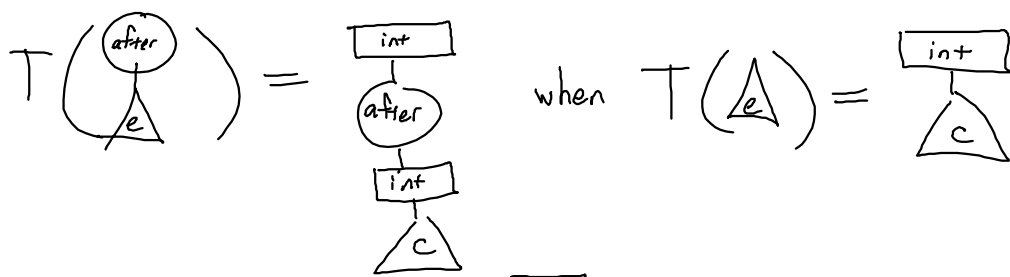
$T(EInt_n) =$

$T(\text{after } 3) =$

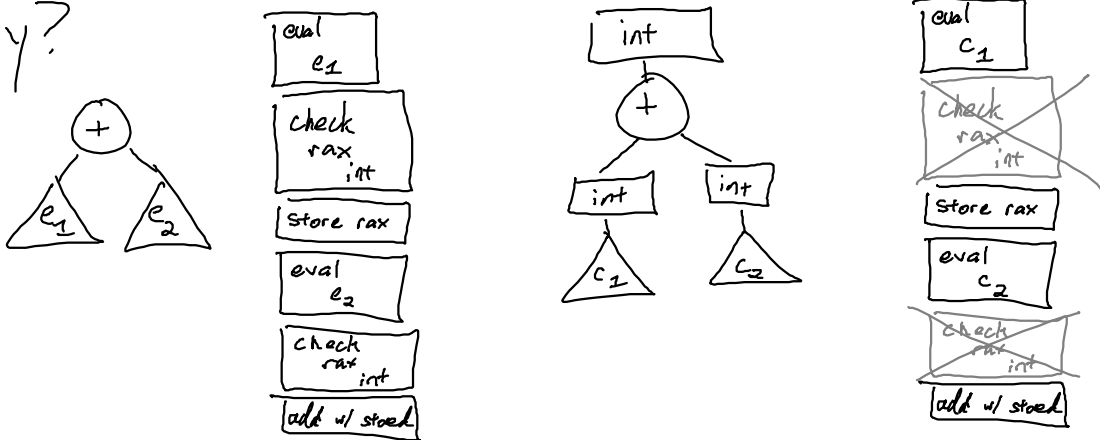
$T(\text{after } e) =$

when $T(\triangle_e) =$

a checked ast will, when evaluated, produce a value in the set that is the indicated type (if it evaluates)



Why?



1. Help programmer understand code & find mistakes
2. Help software (like compiler) understand code (optimization)