Syntactic categories:

- Arithmetic expressions, a
- Boolean expressions, b
- Lists, l

$$l ::= l_1 ++ l_2$$
$$\mid [e_1; e_2; \cdots; e_k]$$

$$[[3;4;5];[6;7]][0] + 8$$

$$\Rightarrow [3;4;5] + 8 \Rightarrow !!!$$

---

a $\quad$ b

al $\quad$ bl $\quad$ all $\quad$ bll

al^3 $\quad$ bl^3

$\vdots$ $\quad$ $\vdots$

- Language L_3 (Ints, Bools, Lists)

$$e ::= \underbrace{0 \mid 1 \mid 2 \mid \dots}_{c \in Int} \mid \underbrace{true \mid false}_{Bool} \mid [e_1; e_2; \dots; e_k]$$

$$\mid e_1 \pm e_2 \mid e_1 \text{ and/or } e_2 \mid \text{not } e_1$$

$$\mid e_1[e_2] \mid e_1 \mathbin{++} e_2$$

$$\mid \text{if } e_1 \text{ then } e_2 \text{ else } e_3$$

$$[3; 4; 5][1] + 2 \Rightarrow 4 + 2 \Rightarrow 6$$

$$[true; false; false][1] + 2 \Rightarrow$$

$$false + 2 \Rightarrow !!!$$

Program source code → Parser → AST → Type checker → Y (permissible) / N (impermissible)

[8; [1; 2; 3]] [0] + 9

How to make sense of expressions in L_3

- Map ~~each~~ expression to a "type"

$$T ::= \text{int} \mid \text{bool} \mid \text{list}\langle T \rangle$$

$$\text{list}\langle \text{list}\langle \text{int} \rangle\rangle \quad \ldots$$

$$e ::= \underbrace{0 \mid 1 \mid 2 \mid \dots}_{c \in Int} \mid \underbrace{true \mid false}_{Bool} \mid [e_1; e_2; \dots; e_k]$$

$$\mid e_1 \pm e_2 \mid e_1 \text{ and/or } e_2 \mid \text{not } e_1$$

$$\mid e_1[e_2] \mid e_1 \mathbin{++} e_2$$

$$\mid \text{if } e_1 \text{ then } e_2 \text{ else } e_3$$

- If e is an integer literal $(0, 1, 2, \dots)$
  then it is of type int
- If boolean literal (true | f)
  then it is of type bool

- If e_1 is of type int &
  e_2 is of type int
  then $e_1 + e_2$ is of type int
  $e_1 - e_2$ is of type int

If e_1 is a list<T>
& e_2 is a list<T>
then e_1 ++ e_2 is of type
list<T>.

$$\frac{e_1 : \text{list}\langle T\rangle \qquad e_2 : \text{list}\langle T\rangle}{e_1 \mathbin{++} e_2 : \text{list}\langle T\rangle}$$

Typing judgment

$$\frac{e_1 : \text{bool} \qquad e_2 : T \qquad e_3 : T}{\text{if } e_1 \text{ then } e_2 \text{ else } e_3 : T}$$

$$\frac{e_1 : T \qquad e_2 : T \qquad \dots \qquad e_k : T}{[e_1 ; e_2 ; \dots ; e_k] : \text{list}\langle T\rangle}$$

$$(2+3) - 8 : int$$

$(2+3) : int$ $\qquad$ $8 : int$

$2 : int$

$3 : int$

<u>Property 1</u> : Progress

If $e : T$ then either:

$e_1 \Rightarrow e_2$, or

e is a value

<u>Property 2</u> : Preservation

~~If $e : T$~~ & $e_1 \Rightarrow e_2$

then $e_2 : T$

- Whenever e_1 is of type bool & e_2 is of type bool

e_1 and e_2, e_1 or e_2, not e_1 : all these are also of type bool.

- If e_1 is a list<T>, for some T, & e_2 is of type int, then $e_1[e_2]$ is of type T.

- Language L_4 (Integers, functions)

Simply-Typed Lambda Calculus

$$e ::= \underbrace{0 \mid 1 \mid 2 \mid \cdots}_{\text{Int}} \;\Big|\; \underbrace{x, y, z, \ldots}_{\text{vars}}$$

Expressions

$$\mid e_1 \pm e_2 \mid e_1\ e_2 \quad \text{(Fun app)}$$

$$\mid \text{fun } (x:?) \to e \quad \text{(Fun abstraction)}$$

$$(\text{fun } (x:\text{Int}) \to x+8) \;\; 2$$

$$(\text{fun } (x:\text{Int}) \to x+8) \; (\text{fun } (x:\text{Int}) \to x)$$

$$\Rightarrow \; !!!$$

Types : $T ::= \text{Int} \mid T_1 \to T_2$

$$\frac{\text{fun } (f : \text{Int} \to \text{Int}) \to (f\ 5) + 3 \quad : (\text{Int} \to \text{Int}) \to \text{Int}}{}$$

$: \text{Int} \to (\text{Int} \to \text{Int})$

fun $(x : \text{Int}) \to$
$($ fun $(y : \text{Int}) \to$
$x + y)$

$$\lambda x : \text{Int} . \lambda y : \text{Int} . x + y$$

Typing judgments for the STLC

$$e ::= \underbrace{0 \mid 1 \mid 2 \mid \cdots}_{\text{Int}} \mid \underbrace{x, y, z, \cdots}_{\text{vars}}$$

$$\mid e_1 \pm e_2 \mid e_1 \; e_2 \quad \text{(Fun app)}$$

$$\mid \text{fun}\,(x:T) \rightarrow e \quad \text{(Fun abstraction)}$$

$$\frac{c \in \text{Int}}{\Gamma \vdash c : \text{Int}} \qquad \frac{\Gamma \vdash e_1 : \text{Int} \quad \Gamma \vdash e_2 : \text{Int}}{\Gamma \vdash e_1 \pm e_2 : \text{Int}}$$

$$\frac{\Gamma \vdash e_1 : T_1 \rightarrow T_2 \quad \Gamma \vdash e_2 : T_1}{\Gamma \vdash e_1 \; e_2 : T_2}$$

$\lambda(x:\text{Int}) \, . \, x$

$\lambda(x:\text{Int} \rightarrow \text{Int}) \, . \, x$

$$\frac{x : T \in \Gamma}{\Gamma \vdash x : T}$$

"Context"

Γ

Turnstile

$$\frac{\Gamma, x:T_1 \vdash e : T_2}{\Gamma \vdash \text{fun}\,(x:T_1) \rightarrow e : T_1 \rightarrow T_2}$$