9-1/ Algebraic Data Types

$$\text{Data Type} = \begin{array}{ll} 1 & \text{— unit (void)} \\ DT \times DT & \text{— pair} \\ DT + DT & \text{— sum /variant} \end{array}$$

$$\text{Bool} = 1 + 1 \qquad 0 \qquad \text{Succ}$$

$$\text{Nat} = 1 + \text{Nat} \quad \text{—unary num}$$

$$\text{Bin} = \underset{\substack{\downarrow \\ \varepsilon}}{1} + \underset{\substack{\downarrow \\ 1s}}{\text{Bin}} + \underset{\substack{\downarrow \\ 0s}}{\text{Bin}}$$

$$\text{List}\langle A\rangle = \underset{\substack{\downarrow \\ \text{mt}}}{1} + \underset{\substack{\nearrow \\ \text{cons}}}{} (A \times \text{List}\langle A\rangle)$$

9-2

BinT<A> 1 + (A × BinT<A> × BinT<A>)

$J_4 \to J_5$:

e := x | v | (e e ...) | if e e e

| case e of (inl x) → e

or (inr x) → e

v := num | bools | prim | λ x (x ...) e

| unit | pair v v | inl v | inr v

prim := .... | pair | inl | inr | fst | snd

9-3/

$$E[\text{fst } (\text{pair } u\ v)] = E[u]$$

$$E[\text{snd } (\text{pair } u\ v)] = E[v]$$

$$E[\text{case } (\text{inl } v) \text{ of } (\text{inl } x_l) \rightarrow e_l$$
$$\text{or } (\text{inr } x_r) \rightarrow er]$$

$$= E[e_l\,[x_l \leftarrow v]]$$

$$E[\text{case } (\text{inr } v) \text{ of } (\text{inl } x_l) \rightarrow e_l$$
$$\text{or } (\text{inr } x_r) \rightarrow e_r]$$

$$= E[er\,[xr \leftarrow v]]$$

9-4) A list is empty or it is
a cons with a thing and then another list

```
empty := inl unit
cons  := λ (data rest) . inr (pair data rest)

length := λ rec (l).
    case l of
        inl _ → 0
        inr p → 1 + rec (snd p)
```

```
9-5/  map := λ rec (f l)
         case l of
              inl _ -> l
              inr p =>
                cons (f (fst p))
                     (rec f (snd p))

     reduce := λ rec (f z l)
        case l of   inl _ -> z
             i      inr p => rec f (f z (fst p))
                                   (snd p)

   reduce (λ (x y) (+ x y)) 0
        (cons 1 (cons 2 (cons 3 empty))) = 6
```

9-6/ $e ::= \ldots \mid \mathrm{obj}\ \{\ x : e,$
$\ldots\ \}$

$\mid\ e.x$

$v ::= \ldots \mid \mathrm{obj}\ \{\ x : v \ldots\ \}$

$$E[\ \mathrm{obj}\ \{x_0 : v_0, \ldots, x_n : v_n\}.\ x_i\ ] = E[v_i]$$

empty% := empty

set o x e = cons (pair $\overbrace{\text{"x"}}^{\text{strings}}$ e) o

obj {} = empty

obj {x_0 : e_0, x_m : e_m ...} = set obj{x_m : e_m} x_0 e_0

e.x = lookup e "x"

```
lookup := λ rec (o k).
        case o of   inl _ → error
                    inr p → if (string=? k
                                   (fst (fst p)))
                               (snd (fst p))
                            o.w.  rec (snd p) k
```

```
9-7/    filter := λ rec (pr l).
          case l of
               inl _ → l
           inr p → let r' = rec pr (snd p)
             if (pr (fst p))
                (cons (fst p) r')
                r'


      append := λ rec (x y).
        case x of
            inl _ → y
            inr p → cons (fst p) (rec (snd p) y)
```