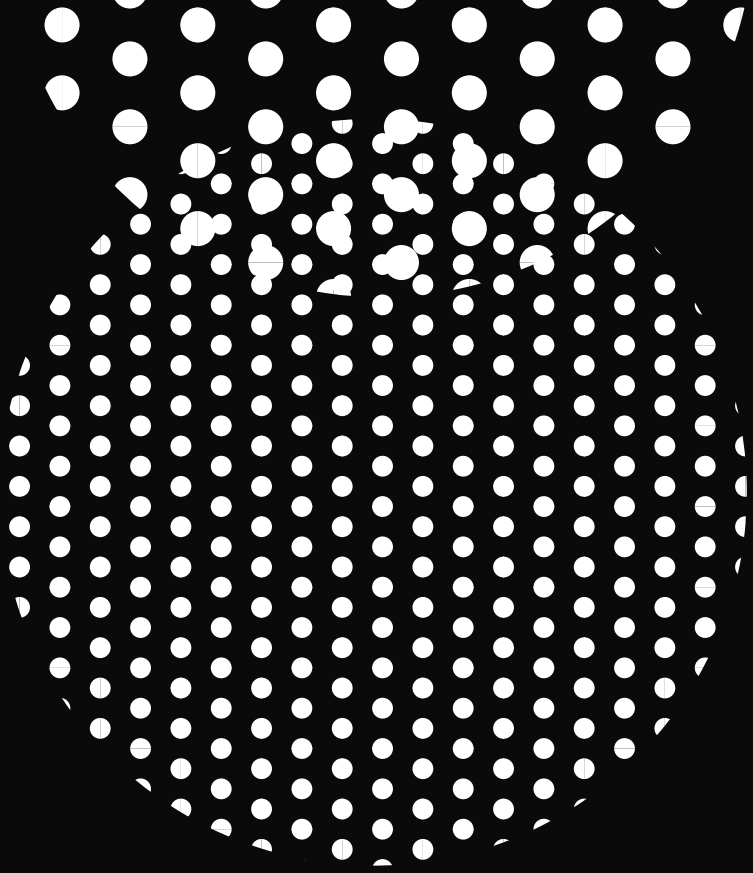

1.1.

1) Hyp: $concat(l_1, Nil) = l_1$ $\qquad$ $concat(Nil, Nil) = Nil$

$$concat(\overbrace{Cel(x, l_1)}^{l_1 :=}, Nil) = Cel(x, concat(l_1, Nil))$$

$$= Cel(x, l_1) = l \quad HR$$

2) $l = Cel(x, l_1)$ $\qquad$ hyp : $lo(concat(l, l)) = 2 \cdot lo(l)$

$$concat(l, l) = concat(Cel(x, l_1), l)$$

$$= Cel(x, concat(l_1, l))$$

$$lo(Cel(x, concat(l_1, l))) = 1 + lo(concat(l_1, l))$$

Mq $lo(concat(l_1, l_2)) = lo(l_1) + lo(l_2)$: HYP REC

$$lo(Cel(x, l_1), l_2) = lo(Cel(x, concat(l_1, l_2))) = 1 + lo(concat(l_1, l_2))$$

$$= \underbrace{1 + lo(l_1)} + lo(l_2)$$

$$= lo(Cel(x, l_1)) + lo(l_2)$$

donc par rec $lo(l_1, l_2) = lo(l_1) + lo(l_2)$

alors $lo(l, l) = lo(l) + lo(l) = 2 \cdot lo(l)$

1.2

1. $(a \mid G_1 G_2$

$(B \mid G_2$

$(C \mid G_1, G_2$

2. $(abb \mid b)^*$

2.1

1.

pile	INT ID SET CST ADD ID ADD ID
ε	int x := 1 + y + z
INT	
Typ Id	
typed_id	
typed_id SET	
typed_id : SET CST	
expr ADD Id	
expr ADD expr	
expr ADD Id	
expr ADD expr	
expr	
var_decl : decls → ε	
decls	
prog	

2.

2.2

1. int f(int x, int g) := x+g

int g(T -> int x, T g) := f(x(g), 1)

2.

$$\frac{\Gamma \vdash e_0 : (T_1, \ldots, T_n) \to T \qquad (\Gamma \vdash e_i : T_i)_{i=1,\ldots,n}}{\Gamma \vdash e_0(e_1, \ldots, e_n) : T}$$

3.

(a)

$$\frac{\dfrac{}{1 : int} \qquad \dfrac{\cancel{f : (int, int) \to int} \quad \dfrac{}{x : int} \quad \dfrac{}{2 : int}}{f(x,2) : int}}{1 + f(x,2) : int}$$

(b)

$$\frac{\dfrac{}{1 : int} \qquad \dfrac{\dfrac{\dfrac{}{f : (T, int) \to (int \to int)} \quad \dfrac{}{x : T} \quad 2 : int}{f(x,2) : int \to int} \quad \dfrac{}{3 : int}}{f(x,2)(3) : int}}{1 + (f(x,2))(3) : int}$$

donc $T = int$

4.

```
let rec type_exp e env =
  match e with
  | Add(e1, e2) ->
      if type_exp e1 = Int && type_exp e2 = Int then Int
      else failwith "type_error"

  | Call(e1, el) -> match type_exp e env with
      | Int -> failwith "type error"
      | Function(t1, ts) -> List.iter2 (fun ei ti -> if type_exp ei env <> ti
          then failwith "type_error") el ts; t

Function (type_exp e1, List.map (fun e2 -> type_exp e2) e)
```

5. récurrence sur e