DOMAIN: LIST PROCESSING

Dreamcoder takes in list processing **tasks**...

Reverse

```
[9 2 4]→[4 2 9]
[4 3]→[3 4]
[3 4 5 6]→ [6 5 4 3]
```

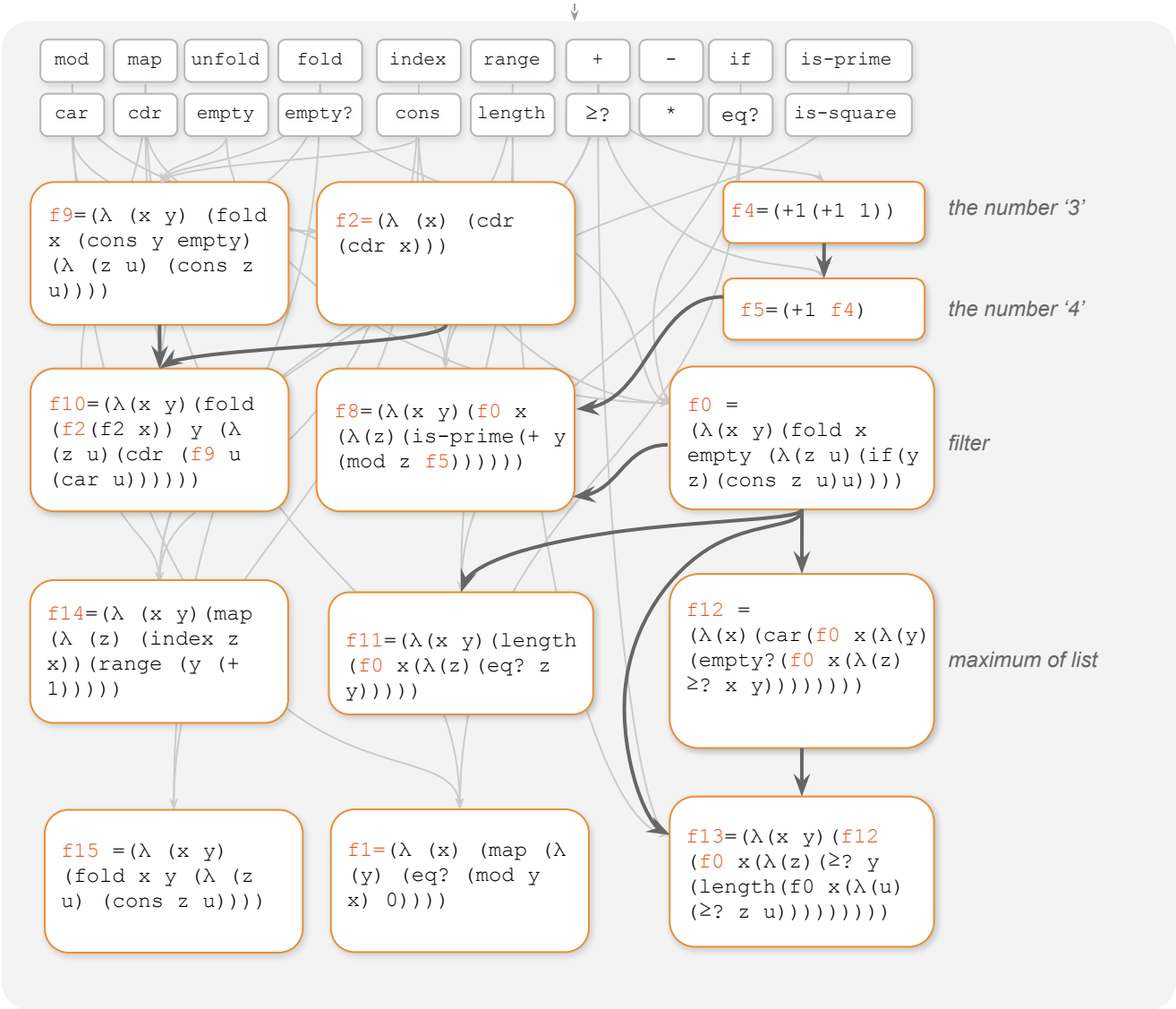
Count how many 3s are in list

```
[3 1 3 3 1 9 2 3]→ 4
[3 3 2 3]→ 3
[3 1 2 3]→  2
```

Sort

```
[9 2 3] → [2 3 9]
[8 7 9 2 5] → [2 5 7 8 9]
[9 8 9] → [8 9 9]
```

From a set of starting primitives...



...to output **programs** in its learned DSL that solve the tasks.

```
(fold(empty(λ(x)(λ(y)
(f9(x y)))
```

```
(λ(x)(length(f0(x)(λ
(y)(eq? f4 y))))
```

```
(λ(map(λ(f13(x y) z (+ u
1)))(range(length u))))
```

It invents a DSL of new complex functions to **filter**, take the **maximum**, and others that **build on its primitives** and **learned functions**.