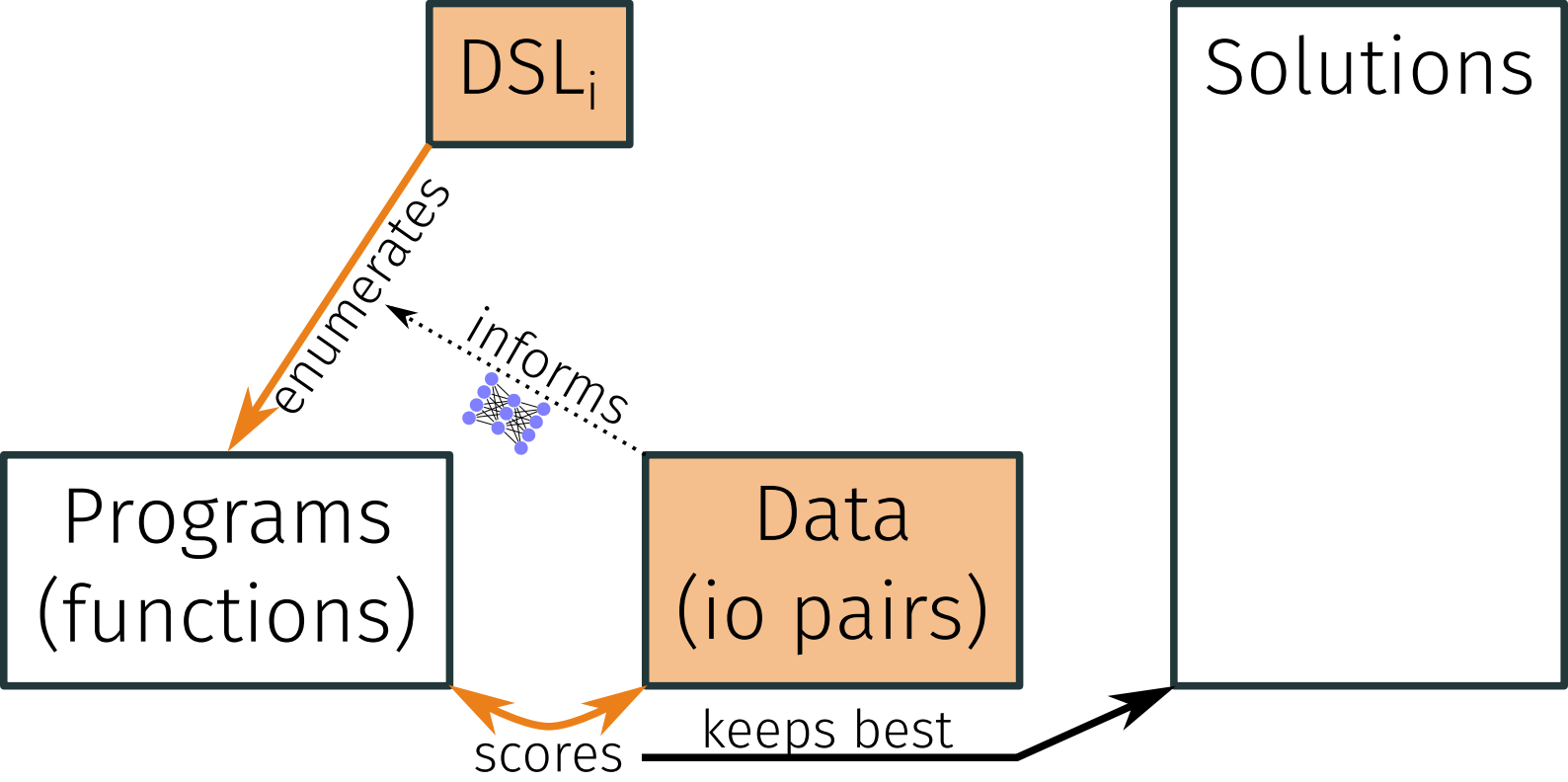


DSL

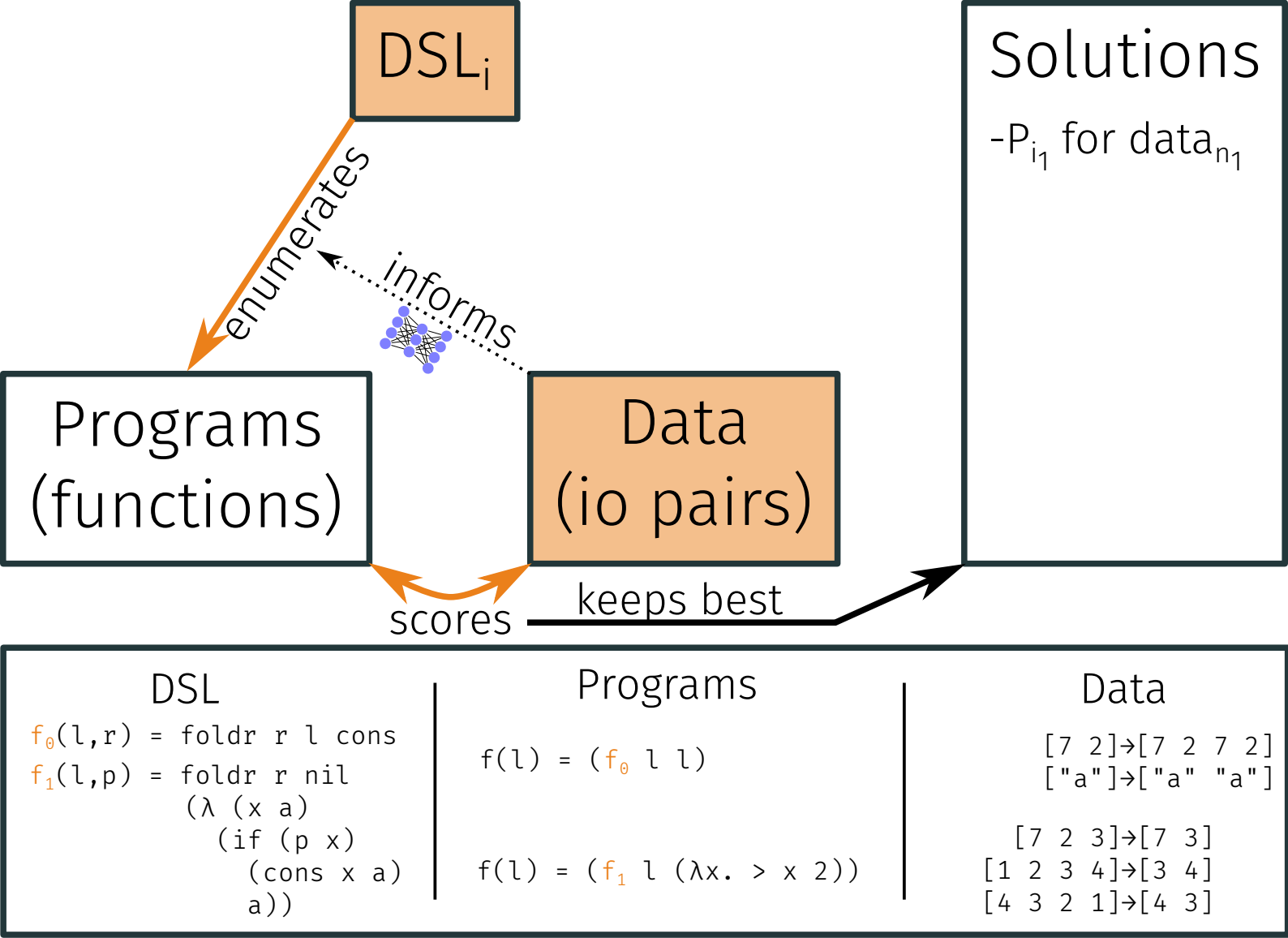
```
f₀(l,r) = foldr r l cons
f₁(l,p) = foldr r nil
            (λ (x a)
              (if (p x)
                (cons x a)
                a))
```

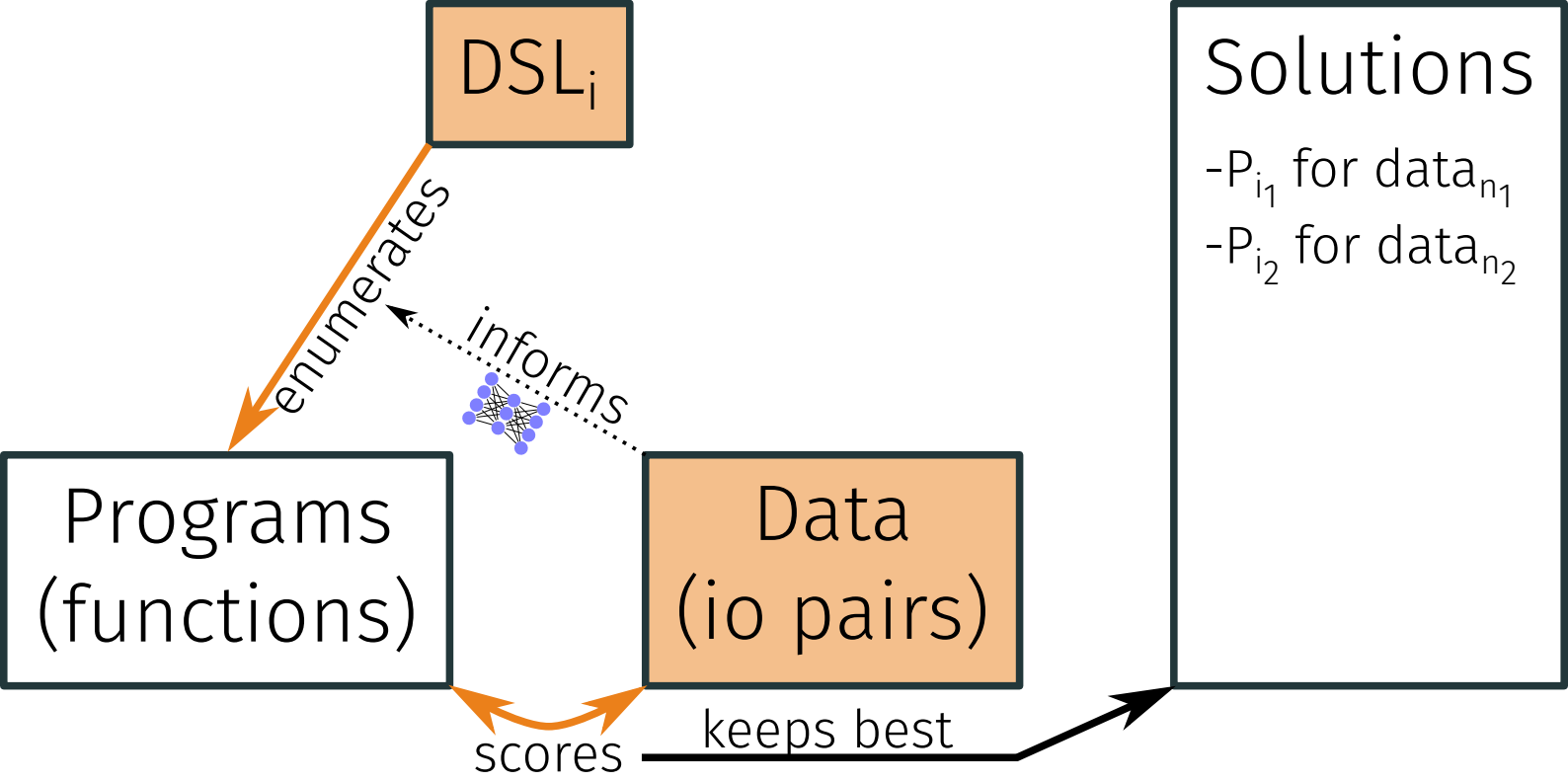
Programs

```
f(l) = (f₀ l l)

f(l) = (f₁ l (λx. > x 2))
```

Data

```
[7 2]→[7 2 7 2]
["a"]→["a" "a"]

[7 2 3]→[7 3]
[1 2 3 4]→[3 4]
[4 3 2 1]→[4 3]
```


DSL$_i$
Solutions
-P_{i_1} for $data_{n_1}$
enumerates
informs
Programs
(functions)
Data
(io pairs)
scores
keeps best
DSL
f_0(l,r) = foldr r l cons
f_1(l,p) = foldr r nil
(λ (x a)
(if (p x)
(cons x a)
a))
Programs
f(l) = (f_0 l l)
f(l) = (f_1 l (λx. > x 2))
Data
[7 2]→[7 2 7 2]
["a"]→["a" "a"]
[7 2 3]→[7 3]
[1 2 3 4]→[3 4]
[4 3 2 1]→[4 3]



DSL

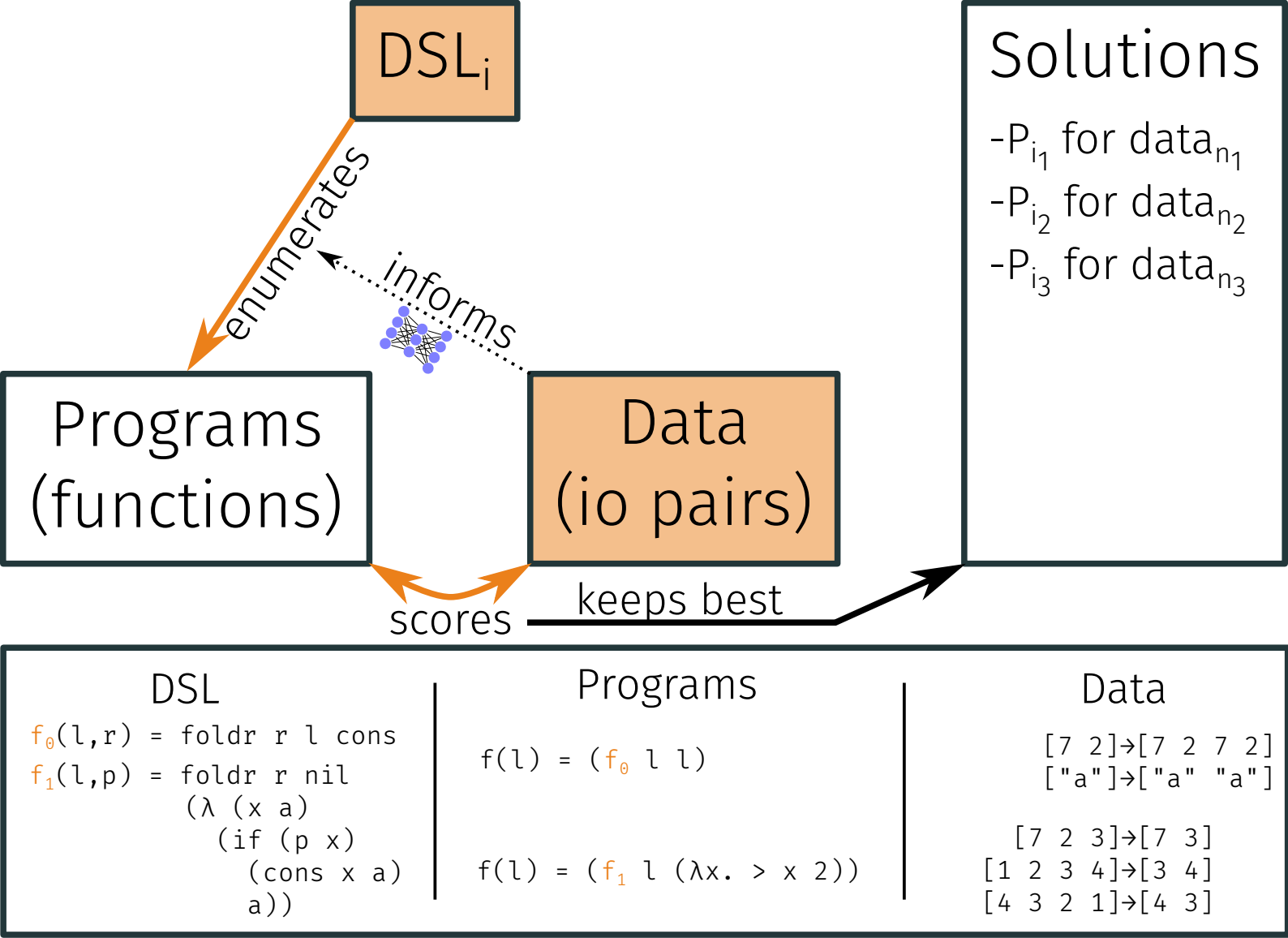
```
f0(l,r) = foldr r l cons
f1(l,p) = foldr r nil
            (λ (x a)
              (if (p x)
                (cons x a)
                a))
```

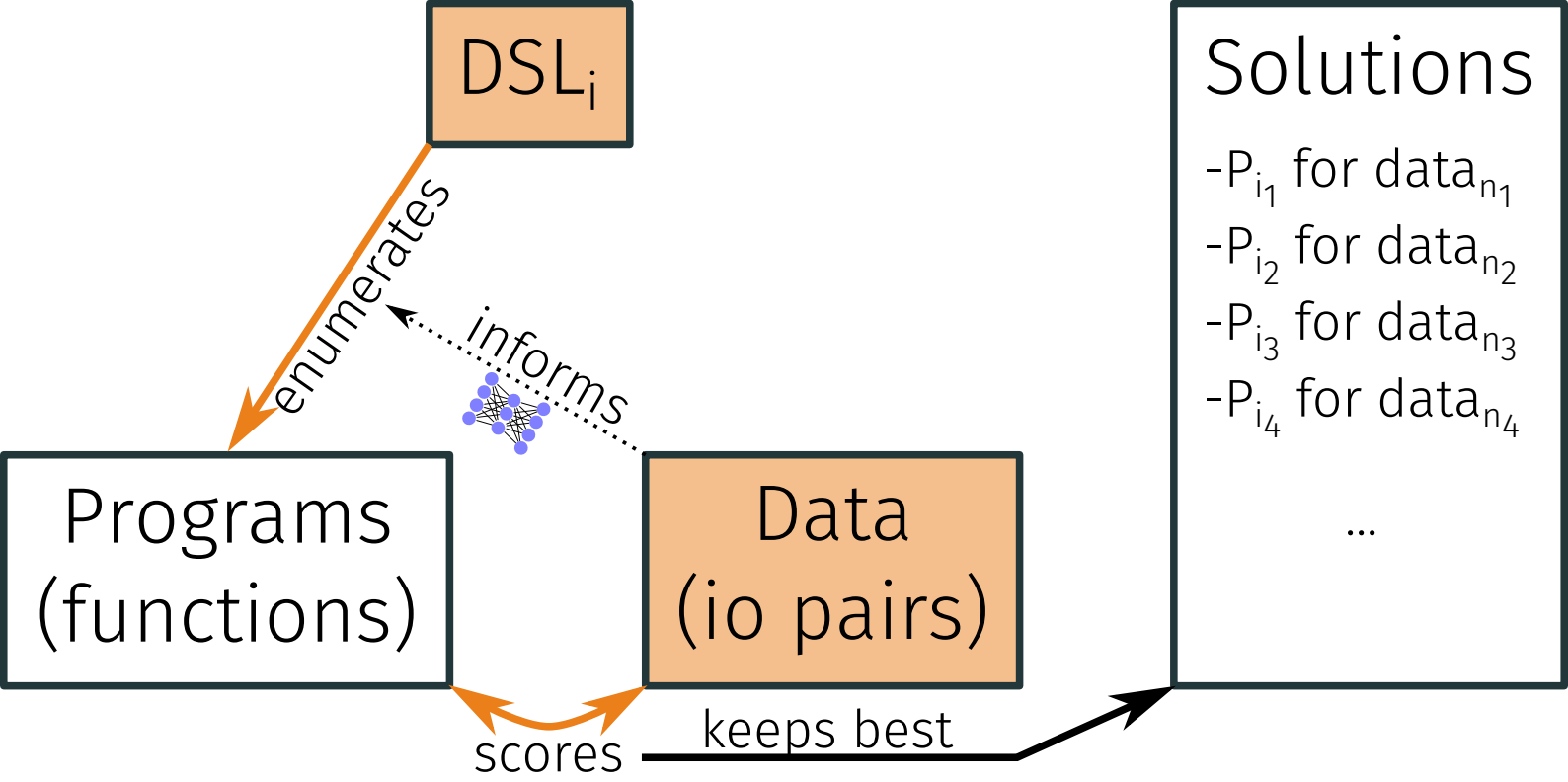
Programs

```
f(l) = (f0 l l)

f(l) = (f1 l (λx. > x 2))
```

Data

```
[7 2]→[7 2 7 2]
["a"]→["a" "a"]

  [7 2 3]→[7 3]
[1 2 3 4]→[3 4]
[4 3 2 1]→[4 3]
```


DSL$_i$
enumerates
informs
Programs
(functions)
Data
(io pairs)
scores
keeps best
Solutions
-P$_{i_1}$ for data$_{n_1}$
-P$_{i_2}$ for data$_{n_2}$
-P$_{i_3}$ for data$_{n_3}$
DSL
f$_0$(l,r) = foldr r l cons
f$_1$(l,p) = foldr r nil
(λ (x a)
(if (p x)
(cons x a)
a))
Programs
f(l) = (f$_0$ l l)
f(l) = (f$_1$ l (λx. > x 2))
Data
[7 2]→[7 2 7 2]
["a"]→["a" "a"]
[7 2 3]→[7 3]
[1 2 3 4]→[3 4]
[4 3 2 1]→[4 3]



DSL

```
f0(l,r) = foldr r l cons
f1(l,p) = foldr r nil
            (λ (x a)
              (if (p x)
                (cons x a)
                a))
```

Programs

```
f(l) = (f0 l l)

f(l) = (f1 l (λx. > x 2))
```

Data

```
[7 2]→[7 2 7 2]
["a"]→["a" "a"]

[7 2 3]→[7 3]
[1 2 3 4]→[3 4]
[4 3 2 1]→[4 3]
```