

DSL

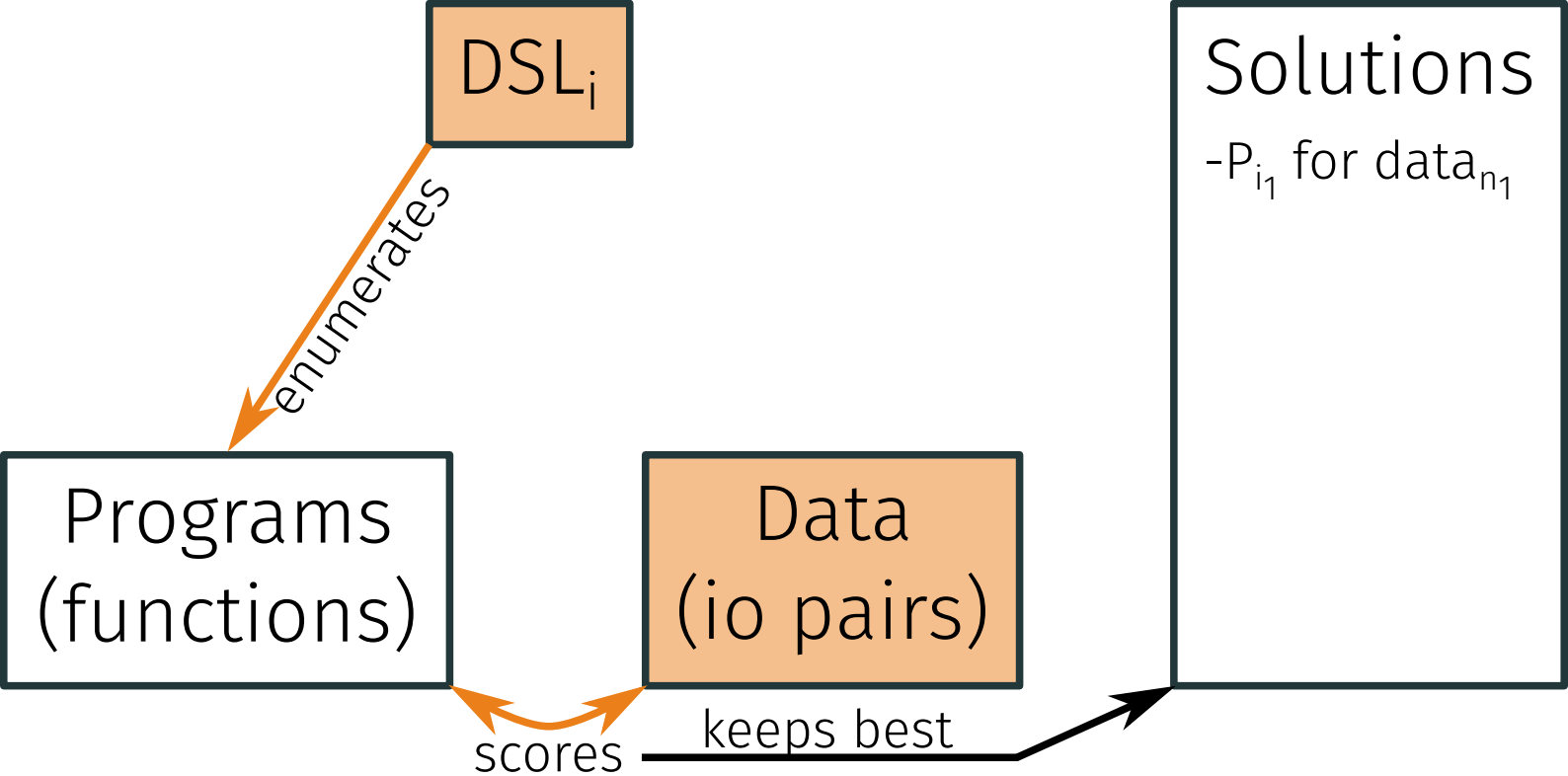
```
f0(l,r) = foldr r l cons  
f1(l,p) = foldr r nil  
          (λ (x a)  
            (if (p x)  
                (cons x a)  
                a))
```

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

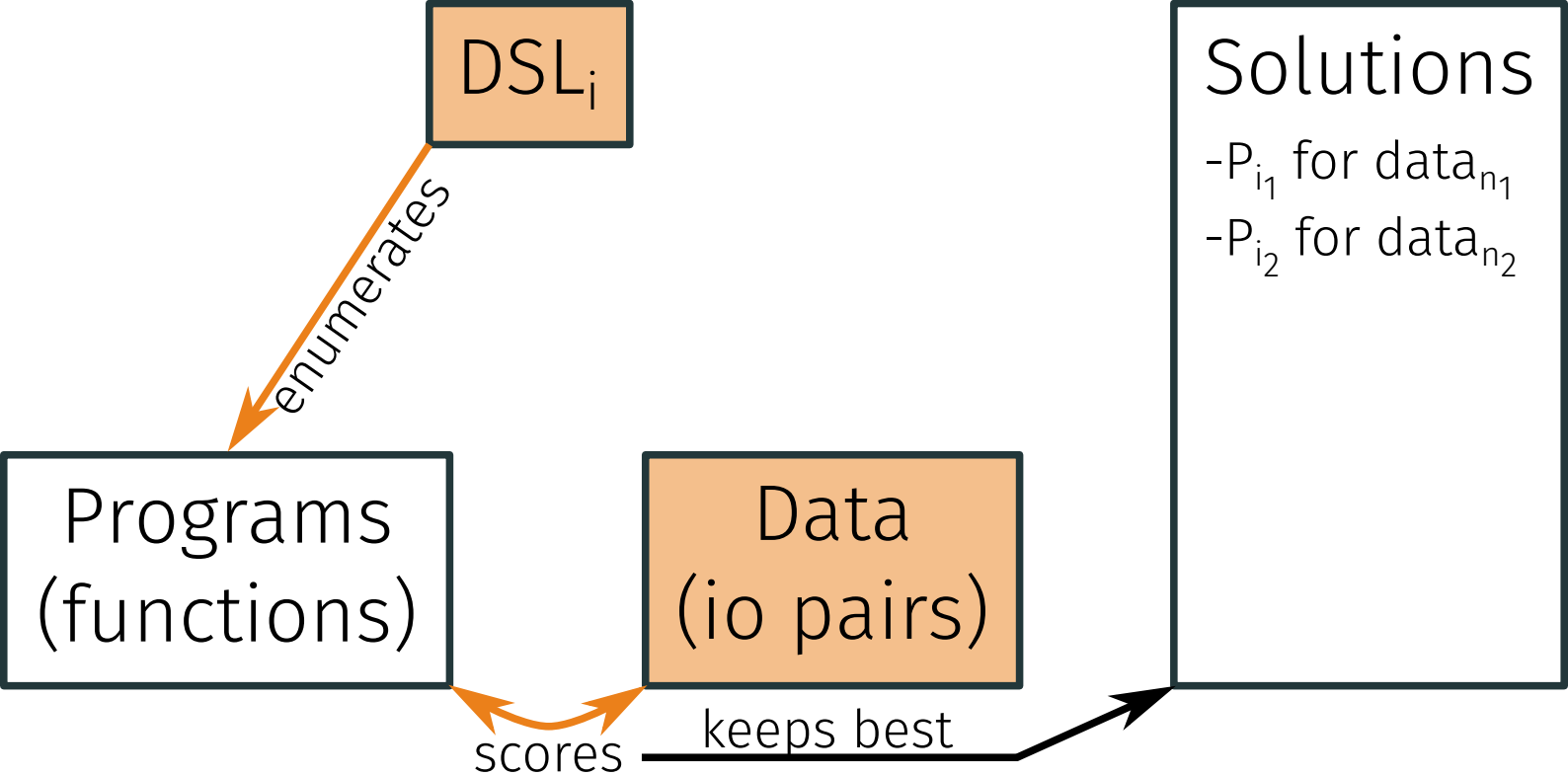
```
f0(l,r) = foldr r l cons  
f1(l,p) = foldr r nil  
          (λ (x a)  
            (if (p x)  
                (cons x a)  
                a))
```

### Programs

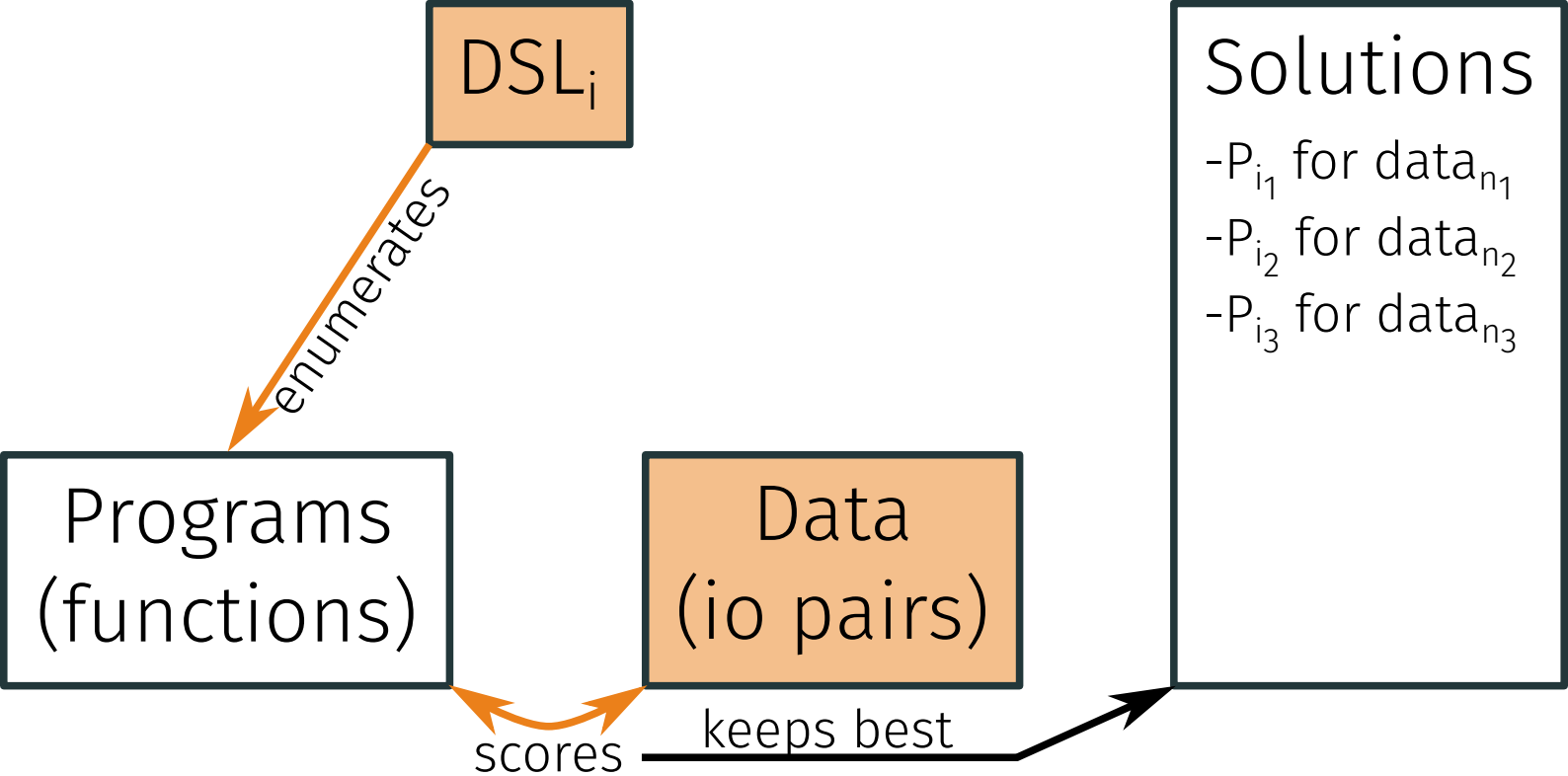
```
f(l) = (f0 l l)  
  
f(l) = (f1 l (λx. > x 2))
```

### 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	Programs	Data
<pre>f<sub>0</sub>(l,r) = foldr r l cons f<sub>1</sub>(l,p) = foldr r nil            (λ (x a)              (if (p x)                  (cons x a)                  a))</pre>	<pre>f(l) = (f<sub>0</sub> l l)  f(l) = (f<sub>1</sub> l (λx. &gt; x 2))</pre>	<pre>[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]</pre>



### DSL

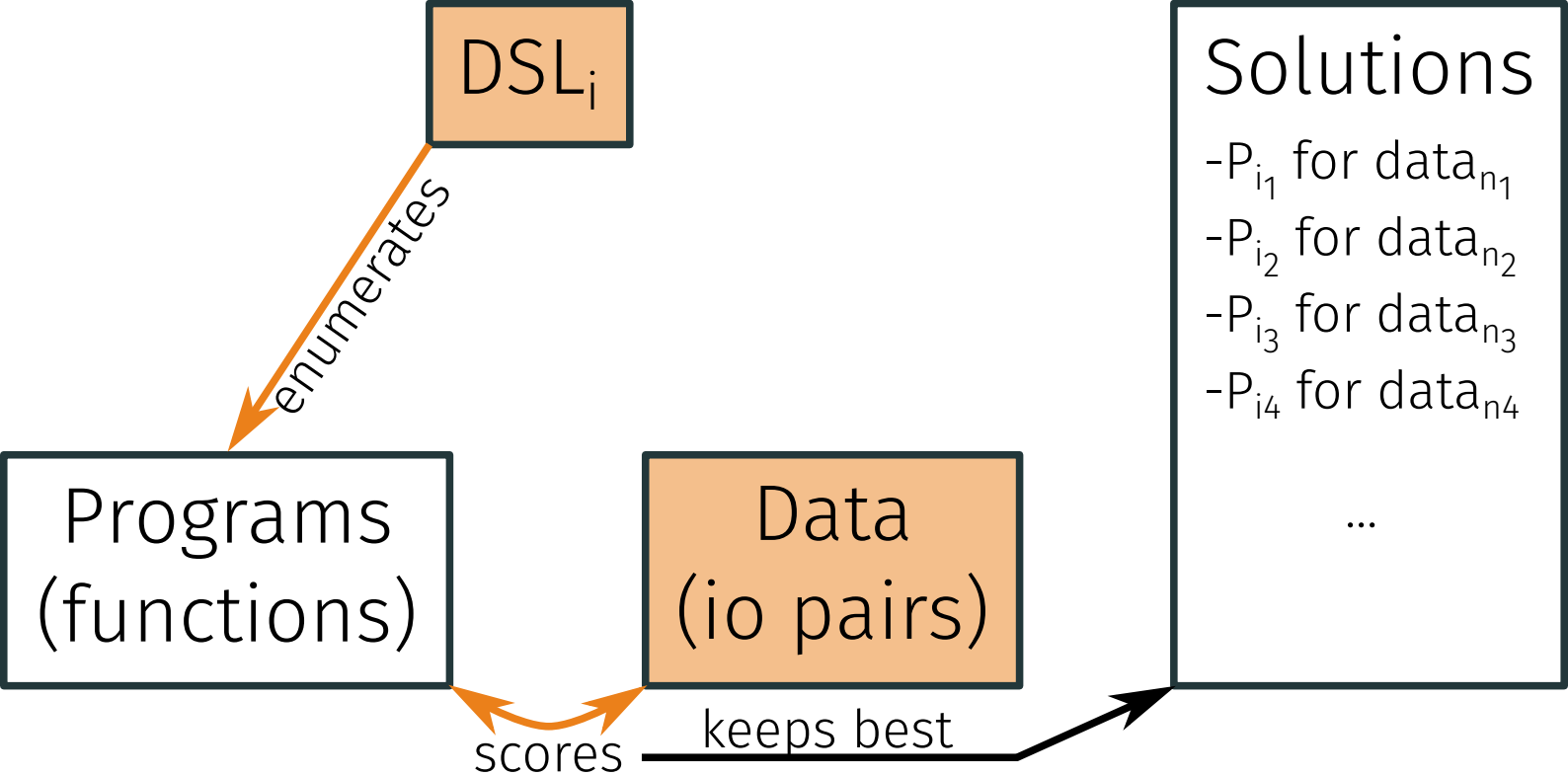
```
f0(l,r) = foldr r l cons  
f1(l,p) = foldr r nil  
          (λ (x a)  
            (if (p x)  
                (cons x a)  
                a))
```

### 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	Programs	Data
<pre>f<sub>0</sub>(l,r) = foldr r l cons f<sub>1</sub>(l,p) = foldr r nil            (λ (x a)             (if (p x)                 (cons x a)                 a))</pre>	<pre>f(l) = (f<sub>0</sub> l l)  f(l) = (f<sub>1</sub> l (λx. &gt; x 2))</pre>	<pre>[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]</pre>