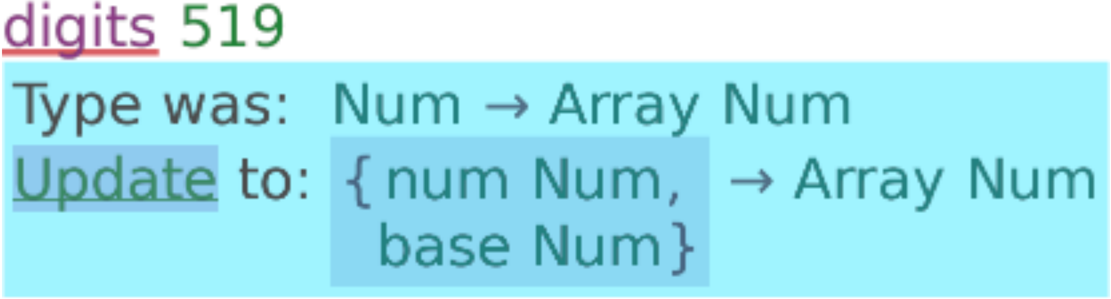


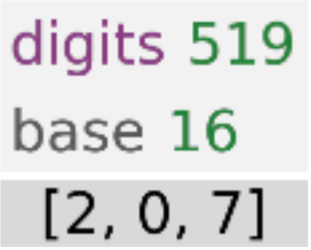
```
digits 519
Type was:   Num → Array Num
Update to: { num Num,
             base Num } → Array Num
```

Blame Assignment Across Definitions

Within each definition, Lamdu stores the types of its dependencies. When they change, Lamdu tracks both the new type and the previously used type.

Until accepting the updated type, the old type is used for type inference, preserving coherency.

When updating, local type mismatches may be created.



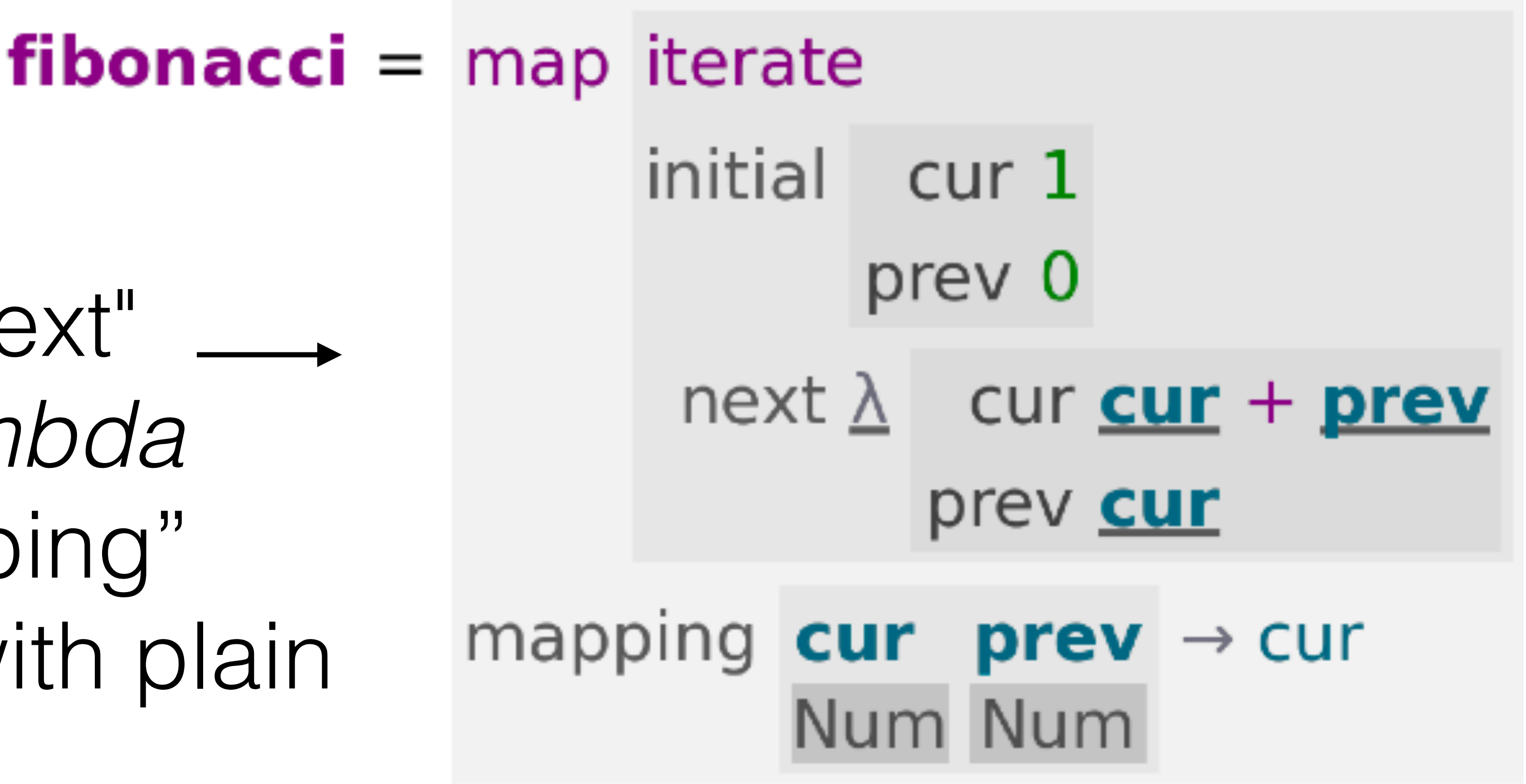
```
digits 519
base 16
```

```
[2, 0, 7]
```

Projectional Syntactic Sugar

Lamdu automatically presents
code with syntactic sugars

Lamdu displays the “next" function using *light lambda* syntax while the “mapping” function is displayed with plain lambda syntax



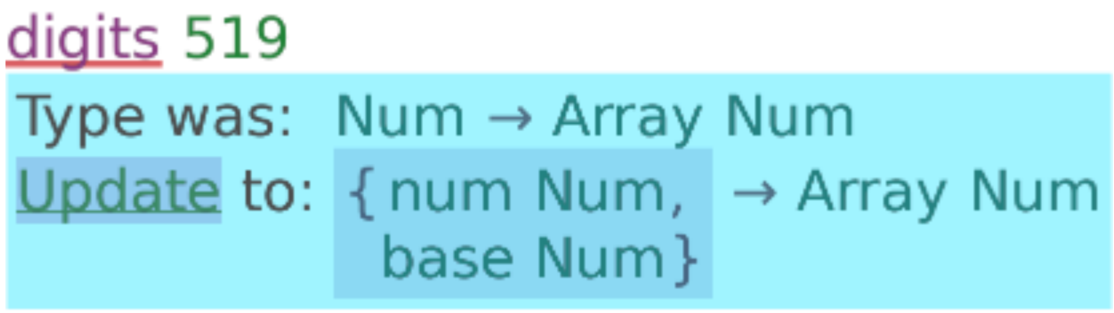
```
fibonacci = map iterate
                    initial  cur 1
                            prev 0
                     next λ   cur cur + prev
                             prev cur
            mapping cur  prev → cur
                    Num  Num
```

Blame Assignment Across Definitions

Within each definition, Lamdu stores the types of its dependencies. When they change, Lamdu tracks both the new type and the previously used type.

Until accepting the updated type, the old type is used for type inference, preserving coherency.

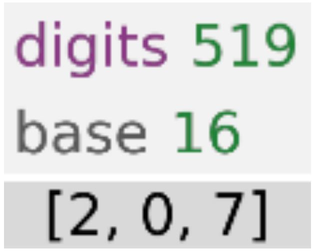
When updating, local type mismatches may be created.



digits 519

Type was: Num → Array Num

Update to: {num Num, base Num} → Array Num



digits 519

base 16

[2, 0, 7]