

My research on constructors that compute

13 May 2021

1 Constructors that compute

The idea is that constructors of inductive types can have an attached pattern matching clause which tells us how they compute. This can be used for many things, among them making quotients or securing nice normal forms for the type.

The type of integers.

```
 $\mathbb{Z} : \mathcal{U} :=$   
|  $z : \mathbb{Z}$   
|  $s : \mathbb{Z} \rightarrow \mathbb{Z}$  with  
|  $s (p \ k) => k$   
|  $p : \mathbb{Z} \rightarrow \mathbb{Z}$  with  
|  $p (s \ k) => k$ 
```

Absolute value.

```
 $\text{abs} : \mathbb{Z} \rightarrow \mathbb{Z} :=$   
|  $z => z$   
|  $s \ k => s \ k$   
|  $p \ k => s \ (\text{abs } k)$ 
```

```
 $\text{abs-abs} : \forall k : \mathbb{Z}, \text{abs } (\text{abs } k) = \text{abs } k$ 
```

```
|  $z => \text{refl}$   
|  $s \ k => \text{refl}$   
|  $p \ k => \text{refl}$ 
```