Natural Numbers, or the best way to enumerate anything

Marc Coiffier

Contents

```
'utils require import
```

• Required module: utils

```
'Nat_context [ { Type '.Nat } { .Nat '.zero } { .Nat 'n -> .Nat ? '.succ } ] def 'Nat Nat_context { .Nat } prods "Natural" defconstr 'zero Nat_context { .zero } funs "0" defconstr Nat 'n -> 'succ Nat_context { .succ ( n ( .Nat .zero .succ ) ) } funs "S n" defconstr ! The Nat type is defined to Natural . \lambda(n : Natural).\mu(n) has type \forall(n : Natural) (Nat^P : Natural \rightarrow Set_1), Nat^P0 \rightarrow (\forall(n_0 : Natural), Nat^Pn_0 \rightarrow Nat^P(Sn_0)) \rightarrow Nat^Pn .
```

 $\lambda(\mathbf{n}:Natural).\,S\,\mathbf{n}\,$ has type $Natural\to Natural$.