CS 494 SF – Software Foundations

William Mansky
Writing Definitions in Coq

- Part of Coq is a functional programming language called Gallina
- A language for writing mathematical definitions
- Has *inductive types, pattern matching, and recursion*

- See Basics.v
Inductive Definitions

Inductive day :=
  | monday
  | tuesday
  | wednesday
  | thursday
  | friday
  | saturday
  | sunday.

Types are sets!

\{ \text{monday, tuesday, ..., saturday, sunday} \}

day is a type
monday : day
tuesday : day
...
saturday : day
sunday : day

day is a set
monday ∈ day
tuesday ∈ day
...
saturday ∈ day
sunday ∈ day
Inductive Definitions

How would you define the natural numbers?
Inductive Definitions

Inductive nat :=
| O : nat
| S : nat -> nat.

New type: nat
New values: O, S O, S (S O), ...

Each value is distinct, and there are no other values of type nat.
Fixpoint pred (n : nat) :=
    match n with
    | O => O
    | S n => n
  end.

S (S O) and pred (S O) are both nats
but pred (S O) computes a value, while S (S O) is a value!
HW1: Basics.v

• Complete all the exercises in Basics.v (you may skip the one marked optional)
• You can run BasicsTest.v to make sure you’ve gotten all of them
• Due Friday 1/25 at 2 PM
• Submit via Gradescope