Proofs are Programs

Induction

Short Recap

Inductive Datatypes

```
Inductive nat : Type :=
  | O
  | S (n : nat).
```

Recursive Functions

```
Fixpoint plus (n : nat) (m : nat) : nat :=
  match n with
  | O => m
  | S n' => S (plus n' m)
  end.
```

Proofs by

```
Theorem plus_O_n :… .
Proof. … Qed.
```

Simplification

```
simpl. reflexivity.
```

Rewriting

```
rewrite -> H.
```

Case Analysis

```
destruct n as [| n'] eqn:E.
```

Quizzes

Introduction Patterns

```
Inductive rgb : Type :=
  | red
  | green
  | blue.

Inductive color : Type :=
  | black
  | white
  | primary (p : rgb).
```

```
c: color
————————————————————
```

Introduction Patterns

```
Inductive rgb : Type :=
  | red
  | green
  | blue.

Inductive color : Type :=
  | black
  | white
  | primary (p : rgb).
```

```
c: color
________________________

destruct c.
```

opens three subgoals
(for all constructors of color)

Introduction Patterns

```
Inductive rgb : Type :=
  | red
  | green
  | blue.

Inductive color : Type :=
  | black
  | white
  | primary (p : rgb).
```

```
c: color
________________________

destruct c.

destruct c as [].
```

opens three subgoals
(for all constructors of color)

opens three subgoals

Introduction Patterns

```
Inductive rgb : Type :=
  | red
  | green
  | blue.

Inductive color : Type :=
  | black
  | white
  | primary (p : rgb).
```

```
c: color
________________________________

destruct c.

destruct c as [].

destruct c as [ | | ].
```

opens three subgoals (for all constructors of color)

opens three subgoals

opens three subgoals

Introduction Patterns

```
Inductive rgb : Type :=
  | red
  | green
  | blue.

Inductive color : Type :=
  | black
  | white
  | primary (p : rgb).
```

```
c: color
```

```
destruct c.
```

opens three subgoals
(for all constructors of color)

```
destruct c as [].
```

opens three subgoals

```
destruct c as [ | | ].
```

opens three subgoals

```
destruct c as [ | | p ].
```

opens three subgoals
+ names argument of primary (p)

Introduction Patterns

```
Inductive rgb : Type :=
  | red
  | green
  | blue.

Inductive color : Type :=
  | black
  | white
  | primary (p : rgb).
```

```
c: color
________________________

destruct c.
destruct c as [].
destruct c as [ | | ].
destruct c as [ | | p ].
destruct c as [ | | [] ].
```

`destruct c.` — opens three subgoals (for all constructors of color)

`destruct c as [].` — opens three subgoals

`destruct c as [ | | ].` — opens three subgoals

`destruct c as [ | | p ].` — opens three subgoals + names argument of primary (p)

`destruct c as [ | | [] ].` — opens 5 subgoals:
- black
- white
- primary red
- primary green
- primary blue

We reach limits easily...

```
Theorem plus_O_n : forall n:nat,
  0 + n = n.
```

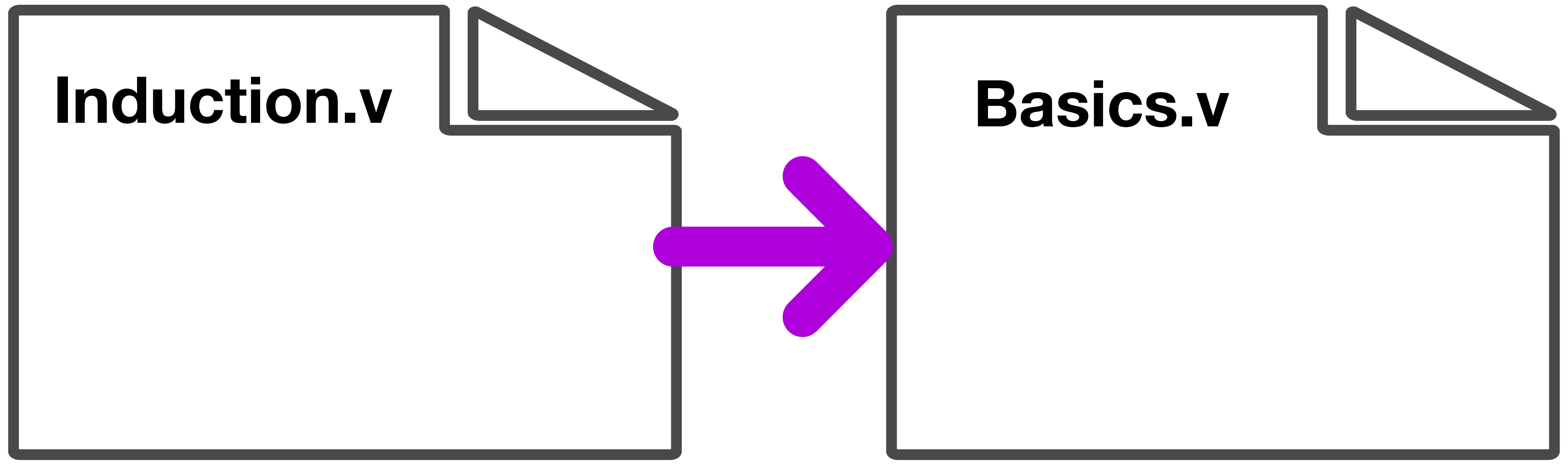
✓

```
Theorem plus_0_r : forall n:nat,
   n + 0 = n.
```

?

```
Fixpoint plus (n : nat) (m : nat) : nat :=
  match n with
  | O => m
  | S n' => S (plus n' m)
  end.
```

Building upon old files

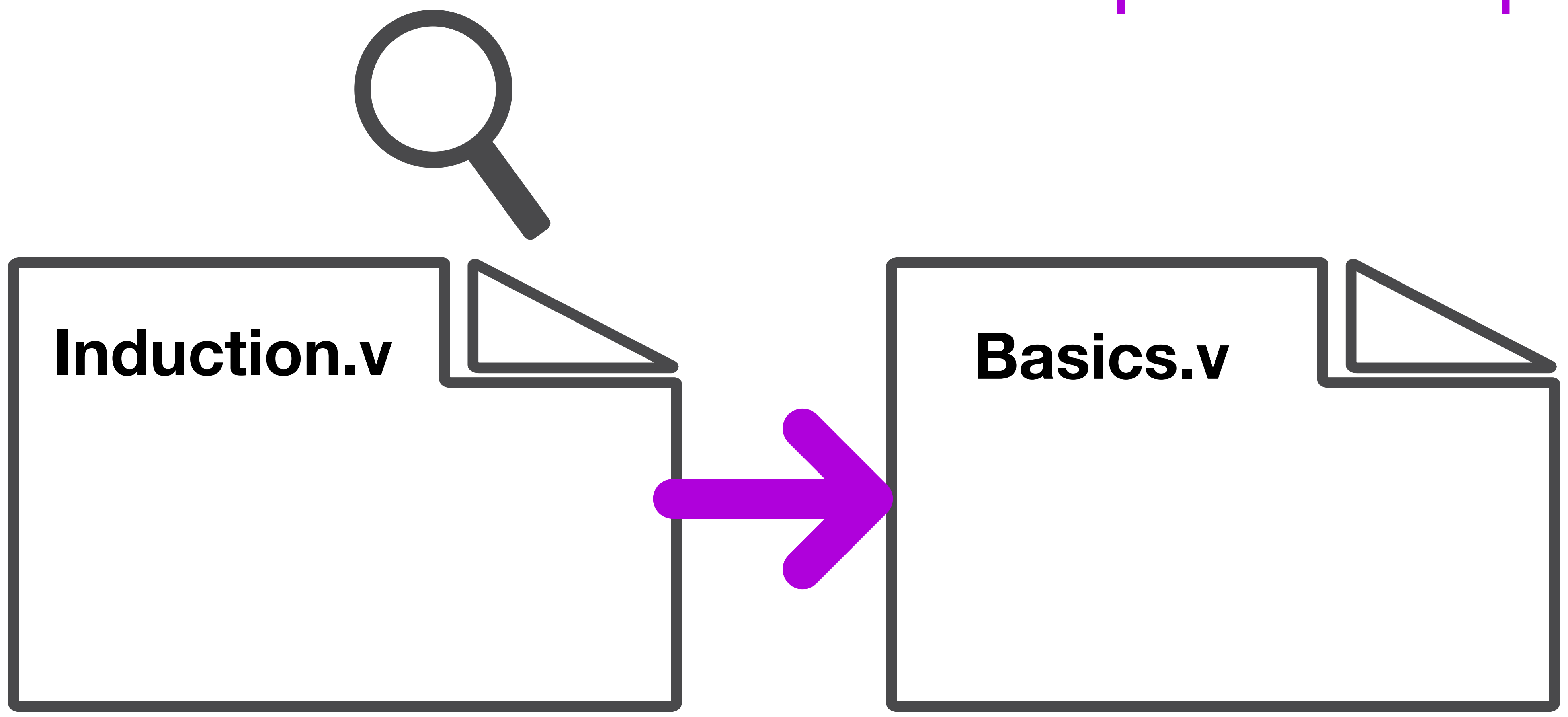


Building upon old files

Prefix associated to directory

File in this directory

```
From LF Require Export Basics.
```

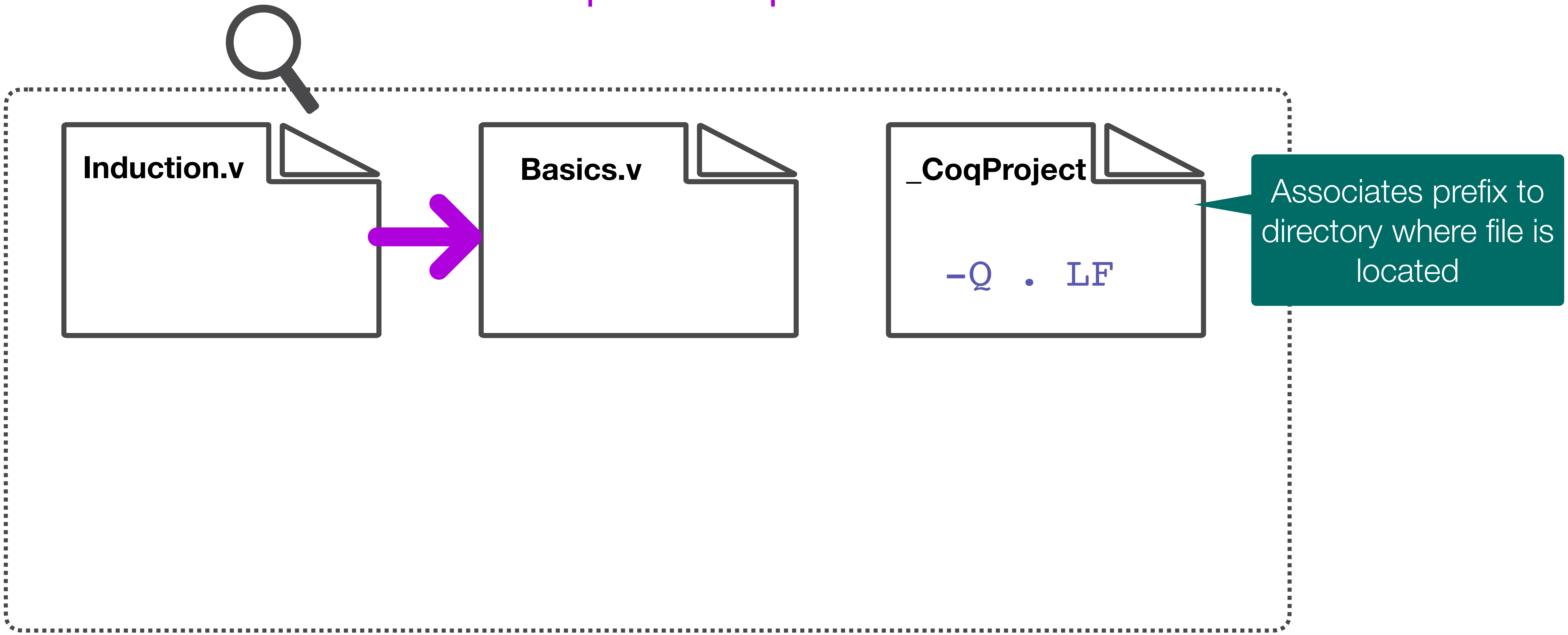


Building upon old files

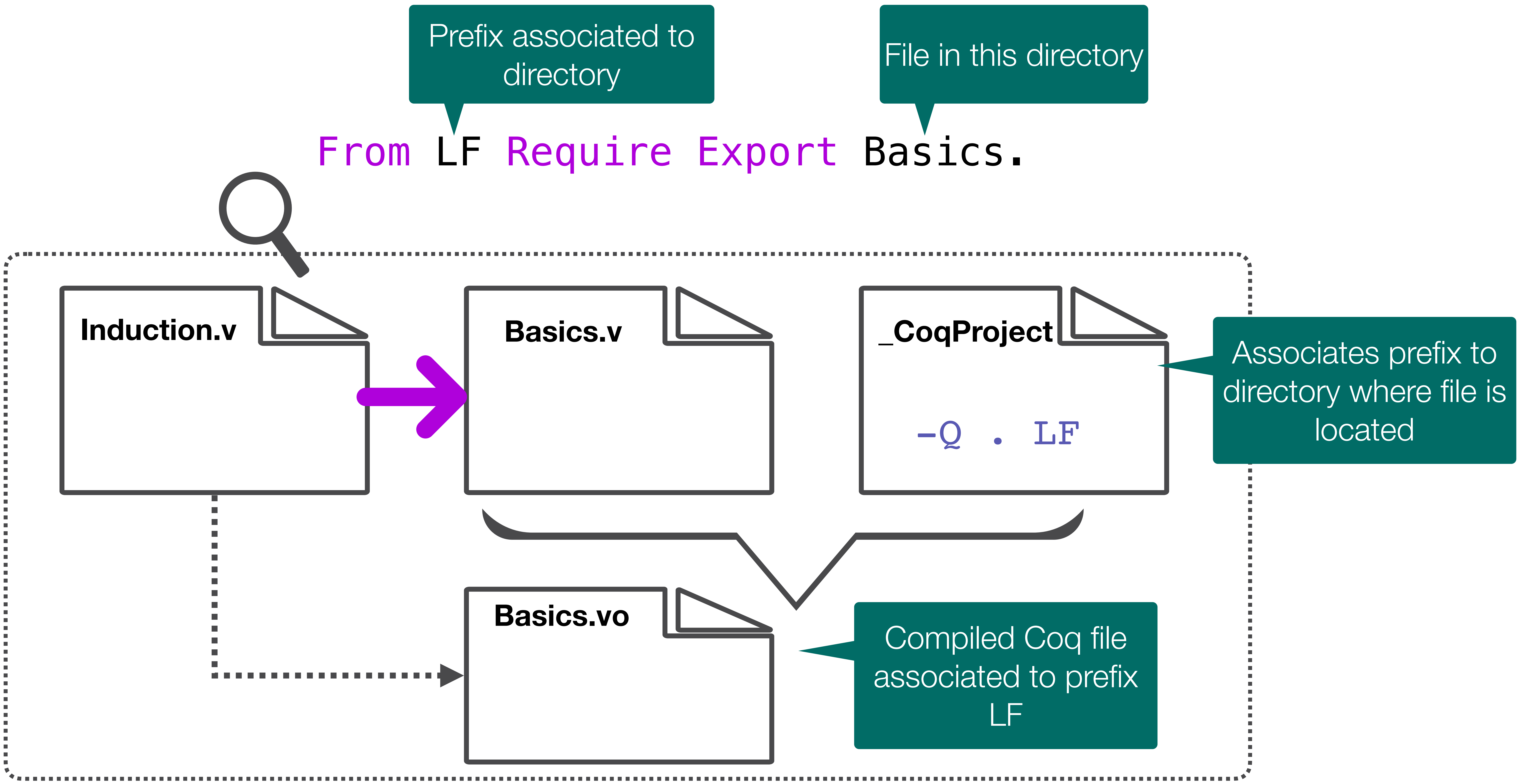
Prefix associated to directory

File in this directory

```
From LF Require Export Basics.
```



Building upon old files



Induction

- General principle to show that $\forall n \in \mathbb{N}\,.\ P(n)$
  - show $P(0)$
  - show that for any $n > 0$ if $P(n-1)$ holds, then so does $P(n)$
    - Alternatively show that for any n' if $P(n')$ holds, then so does $P(S\ n')$
- Typical example $P(n) := \sum_{i=0}^{n} i = \frac{n\ (n+1)}{2}$

Proof by Induction

Proof by Induction

```
Theorem add_0_r : forall n:nat,
  n + 0 = n.
```

Theorem: For any n, n + 0 = n.

Proof by Induction

```
Theorem add_0_r : forall n:nat,
  n + 0 = n.
Proof.
  intros n.
  induction n as [| n' IHn'].
```

Theorem: For any n, n + 0 = n.

Proof: by induction on n.

Proof by Induction

```
Theorem add_0_r : forall n:nat,
  n + 0 = n.
Proof.
  intros n.
  induction n as [| n' IHn'].
  - (* n = 0 *)
```

Theorem: For any n, n + 0 = n.

Proof: by induction on n.

First, suppose n = 0. We must show that 0 + 0 = 0.

Proof by Induction

```
Theorem add_0_r : forall n:nat,
  n + 0 = n.
Proof.
  intros n.
  induction n as [| n' IHn'].
  - (* n = 0 *)
    reflexivity.
```

Theorem: For any n, n + 0 = n.

Proof: by induction on n.

First, suppose n = 0. We must show that 0 + 0 = 0.

This follows directly from the definition of +.

Proof by Induction

```
Theorem add_0_r : forall n:nat,
  n + 0 = n.
Proof.
  intros n.
  induction n as [| n' IHn'].
  - (* n = 0 *)
    reflexivity.
  - (* n = S n' *)
```

Theorem: For any n, n + 0 = n.

Proof: by induction on n.

First, suppose n = 0. We must show that 0 + 0 = 0.

This follows directly from the definition of +.

Next, suppose n = S n’, where n’ + 0 = n’ (IHn’). We must show that (S n’) + 0 = S n’

Proof by Induction

```
Theorem add_0_r : forall n:nat,
  n + 0 = n.
Proof.
  intros n.
  induction n as [| n' IHn'].
  - (* n = 0 *)
    reflexivity.
  - (* n = S n' *)
    simpl.
```

Theorem: For any n, n + 0 = n.

Proof: by induction on n.

First, suppose n = 0. We must show that 0 + 0 = 0.

This follows directly from the definition of +.

Next, suppose n = S n', where n' + 0 = n' (IHn'). We must show that (S n') + 0 = S n'

By the definition of + this follows from S (n' + 0) = S n'.

Proof by Induction

```
Theorem add_0_r : forall n:nat,
  n + 0 = n.
Proof.
  intros n.
  induction n as [| n' IHn'].
  - (* n = 0 *)
    reflexivity.
  - (* n = S n' *)
    simpl.
    rewrite -> IHn'.
```

Theorem: For any n, n + 0 = n.

Proof: by induction on n.

First, suppose n = 0. We must show that 0 + 0 = 0.

This follows directly from the definition of +.

Next, suppose n = S n', where n' + 0 = n' (IHn'). We must show that (S n') + 0 = S n'

By the definition of + this follows from S (n' + 0) = S n'.

By the inductive hypothesis (IHn') this is equivalent to S n' = S n'.

Proof by Induction

```
Theorem add_0_r : forall n:nat,
  n + 0 = n.
Proof.
  intros n.
  induction n as [| n' IHn'].
  - (* n = 0 *)
    reflexivity.
  - (* n = S n' *)
    simpl.
    rewrite -> IHn'.
    reflexivity.
```

Theorem: For any n, n + 0 = n.

Proof: by induction on n.

First, suppose n = 0. We must show that 0 + 0 = 0.

This follows directly from the definition of +.

Next, suppose n = S n’, where n’ + 0 = n’ (IHn’). We must show that (S n’) + 0 = S n’

By the definition of + this follows from S (n’ + 0) = S n’.

By the inductive hypothesis (IHn’) this is equivalent to S n’ = S n’.

This concludes the proof.

Proof by Induction

```
Theorem add_0_r : forall n:nat,
  n + 0 = n.
Proof.
  intros n.
  induction n as [| n' IHn'].
  - (* n = 0 *)
    reflexivity.
  - (* n = S n' *)
    simpl.
    rewrite -> IHn'.
    reflexivity.
Qed.
```

Theorem: For any n, n + 0 = n.

Proof: by induction on n.

First, suppose n = 0. We must show that 0 + 0 = 0.

This follows directly from the definition of +.

Next, suppose n = S n’, where n’ + 0 = n’ (IHn’). We must show that (S n’) + 0 = S n’

By the definition of + this follows from S (n’ + 0) = S n’.

By the inductive hypothesis (IHn’) this is equivalent to S n’ = S n’.

This concludes the proof.

Paper Proofs (Informal Proofs)

Paper Proofs (Informal Proofs)

```
Theorem add_assoc'' : forall n m p : nat,
  n + (m + p) = (n + m) + p.
```

Theorem: For any n m p,
n + (m + p) = (n + m) + p

Paper Proofs (Informal Proofs)

```
Theorem add_assoc'' : forall n m p : nat,
  n + (m + p) = (n + m) + p.
Proof.
  intros n m p.
  induction n as [| n' IHn'].
```

Theorem: For any n m p,
n + (m + p) = (n + m) + p

Proof: By induction on n.

Paper Proofs (Informal Proofs)

```
Theorem add_assoc'' : forall n m p : nat,
  n + (m + p) = (n + m) + p.
Proof.
  intros n m p.
  induction n as [| n' IHn'].
  - (* n = 0 *)
```

Theorem: For any n m p,
n + (m + p) = (n + m) + p

Proof: By induction on n.

First, suppose n = 0. We must show that 0 + (m + p) = (0 + m) + p.

Paper Proofs (Informal Proofs)

```
Theorem add_assoc'' : forall n m p : nat,
  n + (m + p) = (n + m) + p.
Proof.
  intros n m p.
  induction n as [| n' IHn'].
  - (* n = 0 *)
    reflexivity.
```

Theorem: For any n m p,
n + (m + p) = (n + m) + p

Proof: By induction on n.

First, suppose n = 0. We must show that 0 + (m + p) = (0 + m) + p.

This follows directly from the definition of +.

Paper Proofs (Informal Proofs)

```
Theorem add_assoc'' : forall n m p : nat,
  n + (m + p) = (n + m) + p.
Proof.
  intros n m p.
  induction n as [| n' IHn'].
  - (* n = 0 *)
    reflexivity.
  - (* n = S n' *)
```

Theorem: For any n m p,
n + (m + p) = (n + m) + p

Proof: By induction on n.

First, suppose n = 0. We must show that 0 + (m + p) = (0 + m) + p.

This follows directly from the definition of +.

Next, suppose n = S n’, where n’ + (m + p) = (n’ + m) + p (IHn’).
We must show that (S n’) + (m + p) = ((S n’) + m) + p

Paper Proofs (Informal Proofs)

```
Theorem add_assoc'' : forall n m p : nat,
  n + (m + p) = (n + m) + p.
Proof.
  intros n m p.
  induction n as [| n' IHn'].
  - (* n = 0 *)
    reflexivity.
  - (* n = S n' *)

    simpl.
```

Theorem: For any n m p,
n + (m + p) = (n + m) + p

Proof: By induction on n.

First, suppose n = 0. We must show that 0 + (m + p) = (0 + m) + p.

This follows directly from the definition of +.

Next, suppose n = S n', where n' + (m + p) = (n' + m) + p (IHn').
We must show that (S n') + (m + p) = ((S n') + m) + p

By the definition of + this follows from S (n' + (m + p)) = S ((n' + m) + p)

Paper Proofs (Informal Proofs)

```
Theorem add_assoc'' : forall n m p : nat,
  n + (m + p) = (n + m) + p.
Proof.
  intros n m p.
  induction n as [| n' IHn'].
  - (* n = 0 *)
    reflexivity.
  - (* n = S n' *)

    simpl.
    rewrite IHn'.
    reflexivity.
```

Theorem: For any n m p,
n + (m + p) = (n + m) + p

Proof: By induction on n.

First, suppose n = 0. We must show that 0 + (m + p) = (0 + m) + p.

This follows directly from the definition of +.

Next, suppose n = S n', where n' + (m + p) = (n' + m) + p (IHn').
We must show that (S n') + (m + p) = ((S n') + m) + p

By the definition of + this follows from S (n' + (m + p)) = S ((n' + m) + p)

which is immediate from the inductive hypothesis (IHn')

Paper Proofs (Informal Proofs)

```
Theorem add_assoc'' : forall n m p : nat,
  n + (m + p) = (n + m) + p.
Proof.
  intros n m p.
  induction n as [| n' IHn'].
  - (* n = 0 *)
    reflexivity.
  - (* n = S n' *)

    simpl.
    rewrite IHn'.
    reflexivity.
Qed.
```

Theorem: For any n m p,
n + (m + p) = (n + m) + p

Proof: By induction on n.

First, suppose n = 0. We must show that 0 + (m + p) = (0 + m) + p.

This follows directly from the definition of +.

Next, suppose n = S n', where n' + (m + p) = (n' + m) + p (IHn').
We must show that (S n') + (m + p) = ((S n') + m) + p

By the definition of + this follows from S (n' + (m + p)) = S ((n' + m) + p)

which is immediate from the inductive hypothesis (IHn')

Summary

Proofs by

Case Analysis

```
destruct n as [| n'] eqn:E.
```

Induction

```
induction n as [| [n’ H]].
```

Introduction Patterns

```
destruct c as [ | | p ].
```

```
intros [ | | [] ].
```

“Informal Proofs”

Proof: by induction on n.

Proofs within Proofs

```
assert (H:...).
    { … }
```