Chapter 1

Showcase of eIris Inductive

- Explain that we will show an example of the new iInductive system
- First we define the inductive and show the created lemma's and definitions
- Next we use the induction schema in our new tactic to proof a lemma

1.1 Defining a new inductive

- Able to use standard Coq inductive syntax
- Show example
- Show fix-point and explain
- Show other Lemmas and explain as well

```
EI.ind
Inductive is_list (q : Qp) : val → iProp :=
  | empty_is_list : is_list q NONEV
  | cons_is_list l v tl :
    l ↦{#q} (v,tl) -∗ is_list q tl
    -∗ is_list q (SOMEV #l).
```

1.2 Proving a lemma using the inductive

- Explain Lemma we will prove
- Show steps of proof

```
Lemma ind_test_1 (q q' : Qp) (v : val) :
    is_list q v ∗ is_list q' v ∗-∗ is_list (q+q') v.
Proof.
    iSplit.
    - eiIntros "[Hq Hq']".
    iRevert "Hq'".
    eiInduction "Hq" as "[IH | (%l' & %v' & %tl' & Hl' & IH & %Hy)]"; eiIntr
    + iApply is_list_unfold_2.
        iLeft.
        iFrame.
    + simplify_eq.
        iApply is_list_unfold_2.
        iRight.
        iExists l', v', tl'.
        eiDestruct "Hq'" as "[%Hl | (%l'' & %v'' & %tl'' & Hl & Hilq' & %Hv)
        iCombine "Hl' Hl" as "Hl" gives %[_ ?]; simplify_eq.
        iFrame.
        iDestruct "IH" as "[IH _]".
        iSplitL.
        * iApply ("IH" with "[$]").
        * by iPureIntro.
    - eiIntros "Hi".
    eiInduction "Hi" as "[%Ha | (%l & %v' & %tl & [Hq Hq'] & [[Hiq Hiq'] _]
    + simplify_eq.
        iSplitL.
        * iApply is_list_unfold.
        iLeft.
        by iPureIntro.
        * iApply is_list_unfold.
        iLeft.
        by iPureIntro.
    + iSplitL "Hq Hiq".
        * iApply is_list_unfold.
        iRight.
        iExists l, v', tl.
        iFrame.
        by iPureIntro.
        * iApply is_list_unfold.
        iRight.
        iExists l, v', tl.
        iFrame.
        by iPureIntro.
Qed.
```