Chapter 1

Inductive

- What are we going to do
- First create functor
- Proof functor is Monotone
- Apply fixpoint to functor
- Proof unfold lemma's for fixpoint
- Proof iter lemma for fixpoint
- Proof induction lemma for fixpoint

1.1 Functor

1.1.1 Theory

- Take a function and apply it under one level of the inductive statement
- Maybe draw the diagram

1.1.2 Elpi

- We can also make commands
- What do we get as input for our commands
- What do we need to turn it in to
- Show example for is_list

1.2 Monotone

1.2.1 Theory

- What are we proving, what is Monotone
- This can be seen as a proper
- We have to transform proper to Iris Propositions
- Respectfull, pointwise, persistent
- Define Proper instances for connectives
- How to find proper instance
- IProperTop
- Example of Proper proof

1.2.2 Elpi

Proper

- Write tactic for solving IProper proofs
- We write small tactics for different possibel steps
- Simple steps, for respectfull, pointwise, persistent
- Finishing steps for assumption and reflexive implication
- Apply other proper instance
- Find how many arguments to add to connective
- Lemma to get Iproper instance from IProperTop instance
- Apply Lemma IProper
- Compose till all goals proven

Induction for proper

- Create Proper Type for Fixpoint
- Add pointwise for every constructor using fold-map
- Add this to left and right of Respectfull with a persistent around left-hand side
- Apply proper solver

1.3 Least fixpoint

1.3.1 Theory

- Intuition about what a least fixpoint is
- The least fixpoint of a functor holds for a value if for all fixpoints of the functor the value holds
- What does our fixpoint function create
- Example for is_list

1.3.2 Elpi

- The basic structure is this …
- We recurse over the type of the fixpoint to introduce lambda's and foralls
- As the last step we add lambda's for any parameters we have

1.4 Unfold lemma

1.4.1 Theory

- Allows for more easy reasoning about fixpoints by using the functor
- Essential in following proofs

1.4.2 Elpi

1.5 Induction scheme

1.5.1 Theory

1.5.2 Elpi

1.6 `iConstructor`, `iDestruct` abd `iInductive`