

# Chapter 7

## Related work

### 7.1 Other projects using Elpi

- [Tas19] Derive
- [Blo+23] Trakt
- [CST20] Hierarchy builder
- Troq

### 7.2 Fixpoints

- [Bro07; BBC08] Defines inductive predicates in a BI logic . Follows similar steps, given a pre fixpoint function, called an operator. Proves it monotone and then takes the least fixpoint.

Question: What is this exactly? I know Iris is one?

Question: I don't understand how it generates the function, because it seems to do so.

### 7.3 Inductive predicates in other separation logic proof systems

- [Cao+18] mentions representation predicates and shows how to define simple variants. Only works when it has a strictly decreasing recursive call.
- [BCO05] is a tool for checking a subset of separation logic specifications. It includes several built in representation predicates. You are not able to define more.
- [Lei10] uses an SMT based automatic program verifier, with added hints if necessary. It automatically can handle recursive data types and do induction on them.

- [Fil13] They support inductive predicates. But doesn't use separation logic I think.
- [MSS16; SM18] Viper, is based on separation logic. It allows for automatic definition of inductive predicates. Not clear how, never explained as far as I can tell.
- [Jac+11] is based on separation logic.

### **Embedded things**

- [AR] Uses Coq inductive directly. This only works if the PROP you are working in is a concrete sort, often heap to Prop. iProp uses a record as PROP, thus does not work. But we even define it for the BI logic, which is not concrete at all.
- [Chl11] Bedrock, uses Coq fixpoint I think
- [BJB12] Charge, also Coq fixpoint, but need to read further.
- [App14] Need to read
- [RKV21] Also concrete Prop, and uses inductive, but in Agda

## **7.4 Generalized rewriting, Propers**

- [Soz09] Generalized rewriting. They backtrack in their proof search, we do not.

## **7.5 Other IPM implementations**

- MTac2
- LTac2
- Lean