# Chapter 8

# Conclusion

In this thesis we showed how to create inductive predicates automatically in the Iris logic in Coq using Elpi. To accomplish this, we created a command, `eiInd`, which, given a standard Coq inductive statement on the Iris separation logic, defines the inductive predicate with its associated lemmas. Next, we created tactics that allow one to easily eliminate the inductive predicate, apply constructors, and perform induction. These tactics were integrated into a novel partial reimplementation of the IPM in Elpi to allow the inductive predicates to be tightly integrated. Lastly, we showed that the system created for defining inductive predicates can define complicated predicates like the total weakest precondition, as originally defined by the IPM.

## 8.1 Future work

We see three possible directions for feature work. Implement more advanced tactics and definitions related to inductive predicates in the Elpi implementation of the IPM. Add the non-expansive property to relevant definitions and lemmas. Generalize the fixpoint generation to coinductive predicates and the Banach fixpoint.

**Advanced inductive definitions and tactics** The Coq inductive command has support for mutually defined inductive types. These are two inductive predicates that are mutually dependent on each other, i.e., you cannot define one before you define the other. Creating these types of inductive predicates is not currently possible in the system we developed. Adding this to our system might require features from Elpi that are not yet implemented. Another feature of Coq inductive predicates is the `inversion` tactic. This tactic derives the possible constructors with which an inductive predicate was created given its arguments [CT96]. This tactic is an essential part of many Coq proofs about inductive predicates and could be interesting to implement for Iris inductive predicates.

**Non-expansive inductive predicates** The Iris definitions for the fixpoint included a non-expansive requirement for the pre fixpoint function. A pre fixpoint function, $\mathsf{F}$, is non-expansive if $\forall \Phi, \Psi.\ (\forall x.\ \Phi\, x \iff \Psi\, x) \mathrel{-\!\!*} (\forall x.\ \mathsf{F}\, \Phi\, x \iff \mathsf{F}\, \Psi\, x)$. This property has to be expanded to any arity as was done for monotonicity and automatically solved. This would allow for a larger class of inductive predicates to be defined, namely ones where the fixpoint properties depend on the fixpoint being non-expansive.

**Coinductive and Banach inductive predicates** Besides the inductive predicates we defined based on the least fixpoint, there are two other non-automated classes of (co)inductive predicates available in Iris. These are the inductive predicates based on the Banach fixpoint, and coinductive predicates based on the greatest fixpoint. These allow for more types of (co)inductive predicates and other notions of (co)induction on these predicates.