Chapter 3

Fixpoints for representation predicates

- We will show how one can apply the Tarski Fixpoint theorem to create an inductive predicate and how we can create the induction principle from it.

3.1 Problem statement

- The logic described here is embedded in Coq.
- We are only allowed to do structural recursion
- The recursive formulation of $\mathsf{isMLL}$is not structurally recursive
- We need another way to define this predicate
- Iris already has a way of defining fixpoints that would be applicable
- Least fixpoints
- Inspired by the Tarski Fixpoint theorem on lattices and ?
-

3.2 Least fixpoint in Iris

Definition 3.1 (*Monotone predicate*)

Predicate $\mathsf{F}\colon (A \to \mathit{iProp}) \to A \to \mathit{iProp}$ is monotone when for any $\mathit{\Phi}, \Psi\colon A \to \mathit{iProp}$, it holds that

$$\vdash \Box(\forall x.\, \mathit{\Phi} x \twoheadrightarrow \Psi x) \twoheadrightarrow \forall x.\, \mathsf{F}\mathit{\Phi} x \twoheadrightarrow \mathsf{F}\Psi x$$

- Note that there would have been a similar way we could have written the property of a monotone predicate.

$$\square\,(\forall x.\, \Phi x \mathrel{-\!\!*} \Psi x) * \mathsf{F}\Phi x \vdash \mathsf{F}\Psi x$$

- This would be more inline with the way they are written in chapter 2
- However, these rules are a lot more strict in what the context is in which they are used, thus making them a lot harder to use.
- Also, it is the way they are written and used in Iris
- We thus write these like in the definition from now on
- Using this definition of monotone we can define the least fixpoint theorem.

Theorem 3.2 (*Least fixpoint*)

Given a monotone predicate $\mathsf{F}\colon (A \to iProp) \to A \to iProp$, there exists a least fixpoint $\mu\mathsf{F}\colon A \to iProp$ such that

1.

$$\mu\mathsf{F}\, x \dashv\vdash \mathsf{F}\,(\mu\mathsf{F})\, x$$

2.

$$\vdash \square(\forall y.\, \mathsf{F}\,\Phi\, y \mathrel{-\!\!*} \Phi\, y) \mathrel{-\!\!*} \forall x.\, \mu\mathsf{F}\, x \mathrel{-\!\!*} \Phi\, x$$

Proof. Given a monotone predicate $\mathsf{F}\colon (A \to iProp) \to A \to iProp$ we define $\mu\mathsf{F}$ as

$$\mu\mathsf{F}\, x \triangleq \forall \Phi.\ \square(\forall y.\, \mathsf{F}\,\Phi\, y \mathrel{-\!\!*} \Phi\, y) \mathrel{-\!\!*} \Phi\, x$$

We now prove the two properties of the least fixpoint

1. We start with proving this right to left, then using the result, prove left to right.

   R-L We first unfold the definition of $\mu\mathsf{F}\, x$.

   $$\mathsf{F}\,\mu\mathsf{F}\, x \vdash \forall \Phi.\ \square(\forall y.\, \mathsf{F}\,\Phi\, y \mathrel{-\!\!*} \Phi\, y) \mathrel{-\!\!*} \Phi\, x$$

   Next we introduce Φ and the wand.

   $$\mathsf{F}\,\mu\mathsf{F}\, x * \square(\forall y.\, \mathsf{F}\,\Phi\, y \mathrel{-\!\!*} \Phi\, y) \vdash \Phi\, x$$

   We now apply $\square(\forall y.\, \mathsf{F}\,\Phi\, y \mathrel{-\!\!*} \Phi\, y)$ to $\Phi\, x$.

   $$\mathsf{F}\,\mu\mathsf{F}\, x * \square(\forall y.\, \mathsf{F}\,\Phi\, y \mathrel{-\!\!*} \Phi\, y) \vdash \mathsf{F}\,\Phi\, x$$

We can now use the monotonicity of F with the assumption $\mathsf{F}\,\mu\mathsf{F}\,x$

$$\square(\forall y.\, \mathsf{F}\,\Phi\,y \twoheadrightarrow \Phi\,y) \vdash \mu\mathsf{F}\,x \twoheadrightarrow \Phi\,x$$

After unfolding the definition of $\mu\,x$ and introducing the wand we get

$$(\forall \Phi.\ \square(\forall y.\, \mathsf{F}\,\Phi\,y \twoheadrightarrow \Phi\,y) \twoheadrightarrow \Phi\,x) * \square(\forall y.\, \mathsf{F}\,\Phi\,y \twoheadrightarrow \Phi\,y) \vdash \Phi\,x$$

This statement holds by application of the first assumption.

L-R We again first unfold the definition of $\mu\mathsf{F}\,x$.

$$\forall \Phi.\ \square(\forall y.\, \mathsf{F}\,\Phi\,y \twoheadrightarrow \Phi\,y) \twoheadrightarrow \Phi\,x \vdash \mathsf{F}\,\mu\mathsf{F}\,x$$

We apply the assumption with $\Phi = \mathsf{F}\,\mu\mathsf{F}$ resulting in the following statement after introductions

$$\mathsf{F}\,(\mathsf{F}\,\mu\mathsf{F})\,x \vdash \mathsf{F}\,\mu\mathsf{F}\,x$$

This holds because of monotonicity of F and the above proved property.

2. This follows directly from unfolding the definition of $\mu\mathsf{F}$.

□

- The second property of the least fixpoint is the normal induction property.
- However, it is often useful to make it stronger

Lemma 3.3 (*least fixpoint strong induction principle*)

Given a monotone predicate $\mathsf{F}\colon (A \to iProp) \to (A \to iProp)$, it holds that

$$\square(\forall x.\, \mathsf{F}\,(\lambda y.\, \Phi\,y \wedge \mu\mathsf{F}\,y)\,x \twoheadrightarrow \Phi\,x) \twoheadrightarrow \forall x.\, \mu\mathsf{F}\,x \twoheadrightarrow \Phi\,x$$

- We now show how this can be applied to create the isMLLpredicate

Example 3.4 (*Iris least fixpoint of* isMLL)

- We want to transform the non-structurally recursive definition of isMLLinto a least fixpoint

$$\mathsf{isMLL}\,hd\,\vec{v} = \begin{array}{ll} & hd = \mathbf{none} * \vec{v} = [] \\ \vee & \exists \ell, v', tl.\ hd = \mathbf{some}\,l * l \mapsto (v', \mathbf{true}, tl) * \mathsf{isMLL}\,tl\,\vec{v} \\ \vee & \exists \ell, v', \vec{v}'', tl.\ hd = \mathbf{some}\,l * l \mapsto (v', \mathbf{false}, tl) * \\ & \vec{v} = v' :: \vec{v}'' * \mathsf{isMLL}\,tl\,\vec{v}'' \end{array}$$

- We start by ?ing any recursive calls in the definition in order to create a functor?

$$\mathsf{isMLL_F}\,\Phi\,hd\,\vec{v} \triangleq \begin{array}{ll} & hd = \textbf{none} * \vec{v} = [] \\ \vee & \exists \ell, v', tl.\, hd = \textbf{some}\, l * l \mapsto (v', \textbf{true}, tl) * \Phi\, tl\, \vec{v} \\ \vee & \exists \ell, v', \vec{v}'', tl.\, hd = \textbf{some}\, l * l \mapsto (v', \textbf{false}, tl) * \\ & \vec{v} = [v'] + \vec{v}'' * \Phi\, tl\, \vec{v}'' \end{array}$$

- Predicate $\mathsf{isMLL_F}$ now has type $(Val \rightarrow \overrightarrow{Val} \rightarrow iProp) \rightarrow Val \rightarrow \overrightarrow{Val} \rightarrow iProp$
- However, the least fixpoint only works for functors of type $(A \rightarrow iProp) \rightarrow A \rightarrow iProp$
- We solve this by currying $\mathsf{isMLL_F}$ into $\mathsf{isMLL'_F}$: $((Val, \overrightarrow{Val}) \rightarrow iProp) \rightarrow (Val, \overrightarrow{Val}) \rightarrow iProp$

$$\mathsf{isMLL'_F}\,(hd, \vec{v}) \triangleq \mathsf{isMLL_F}\, hd\, \vec{v}$$

- In order to apply the fixpoint theorem, we need $\mathsf{isMLL'_F}$ to be monotone

Proof. To prove $\mathsf{isMLL'_F}$ is monotone, we need the following to hold.

$$\Box(\forall x.\, \Phi\, x \twoheadrightarrow \Psi\, x) \twoheadrightarrow \forall x.\, \mathsf{isMLL'_F}\, \Phi\, x \twoheadrightarrow \mathsf{isMLL'_F}\, \Psi\, x$$

... $\Box$

- Given that $\mathsf{isMLL'_F}$ is monotone, we now know from theorem 3.2 that the least fixpoint exists of $\mathsf{isMLL'_F}$
- We can now define $\mathsf{isMLL'_F}$ as

$$\begin{aligned} \mathsf{isMLL'}\,(hd, \vec{v}) &\triangleq \mu(\mathsf{isMLL'_F})\,(hd, \vec{v}) \\ &= \forall \Phi.\ \Box(\forall y.\, \mathsf{isMLL'_F}\, \Phi\, y \twoheadrightarrow \Phi\, y) \twoheadrightarrow \Phi\, x \end{aligned}$$

- To finish the definition of isMLLwe uncurry the created fixpoint

$$\mathsf{isMLL}\, hd\, \vec{v} \triangleq \mathsf{isMLL'}\,(hd, \vec{v})$$

Question: Also highlight the strong induction already or not?

3.3 Changing arities

- We modify the definitions as described in Iris to allow for multiple arity functors.

- The first step in automating creation of fixpoints is to deal with predicates with more than one argument

- In example 3.4 we solved this by currying the predicate before taking the fixpoint

- When automating the process we solved this somewhat differently

- We change the definitions of and theorems used to match the arity of the predicate we want to take the fixpoint of

Definition 3.5 (*Monotone predicate*)

For any $n \in \mathbb{N}$, predicate $\mathsf{F}\colon (A_1 \to \cdots \to A_n \to iProp) \to A_1 \to \cdots \to A_n \to iProp$ is monotone when for any $\Phi, \Psi\colon A_1 \to \cdots \to A_n \to iProp$, it holds that

$$\vdash \begin{array}{l} \Box(\forall x_1, \ldots, x_n.\, \Phi\, x_1\, \ldots\, x_n \twoheadrightarrow \Psi\, x_1\, \ldots\, x_n) \twoheadrightarrow \\ \forall x_1, \ldots, x_n.\, \mathsf{F}\, \Phi\, x_1\, \ldots\, x_n \twoheadrightarrow \mathsf{F}\, \Psi\, x_1\, \ldots\, x_n \end{array}$$

- This definition also applies for $n = 0$

- For example, we can prove the separating conjunction monotone in both its arguments

Lemma 3.6 (*Seperation conjuction is monotone*)

The separation conjunction is monotone in its left and right argument.

Proof. We only prove monotonicity in its left argument, the proof for the right side is identical. We thus need to prove $\Phi_R P = P * R$ is monotone. expanding the definition of monotone for arity one we get the following statement.

$$\vdash \Box(P \twoheadrightarrow Q) \twoheadrightarrow P * R \twoheadrightarrow Q * R$$

We introduce the wands and persistence modalities giving us the assumptions, $P \twoheadrightarrow Q$, P and R. We then use $*$-MONO using the first two assumptions for proving P and using the last assumption for proving R. That $P \twoheadrightarrow Q * P \vdash Q$ holds follows from $\twoheadrightarrow$I-E, and $R \vdash R$ holds directly. $\Box$

- In the same way we also modify the least fixpoint theorem

Theorem 3.7 (*Least fixpoint*)

Given an $n \in \mathbb{N}$ *and a monotone predicate* $\mathsf{F}\colon (A_1 \to \cdots \to A_n \to \mathit{iProp}) \to A_1 \to \cdots \to A_n \to \mathit{iProp}$, *there exists a least fixpoint* $\mu\mathsf{F}\colon A_1 \to \cdots \to A_n \to \mathit{iProp}$ *such that*

1.

$$\mu\mathsf{F}\, x_1 \,\ldots\, x_n \dashv\vdash \mathsf{F}\,(\mu\mathsf{F})\, x_1 \,\ldots\, x_n$$

2.

$$\vdash \begin{array}{l} \Box(\forall y_1, \ldots, y_n.\, \mathsf{F}\,\Phi\, y_1 \,\ldots\, y_n \twoheadrightarrow \Phi\, y_1 \,\ldots\, y_n) \twoheadrightarrow \\ \forall y_1, \ldots, y_n.\, \mu\mathsf{F}\, x_1 \,\ldots\, x_n \twoheadrightarrow \Phi\, x_1 \,\ldots\, x_n \end{array}$$

- The proof follows the same steps as the proof for theorem 3.2

3.4 Monotone proof search

- We create a system for syntactically finding proofs of monotonicity
- Based on generalized rewriting system in coq by Sozeau [Soz09].
- Define monotonicity of connectives in separation logic using proper elements of relations

TODO: This is not sufficient but stuck on it

Definition 3.8 (*Proper element of a relation*)

Given a relation $R\colon A \to A \to \mathit{iProp}$ and an element $x \in A$, x is a proper element of R if $R\, x\, x$

- When the relation is reflexive, all possible elements are Proper
- For example if we take the magic wand as relation, all propositions are proper.
-

Definition 3.9 (*Respectful relation*)

The respectful relation $R \hookrightarrow R'\colon (A \to B) \to (A \to B) \to \mathit{iProp}$ of two relations $R\colon A \to A \to \mathit{iProp}$, $R'\colon B \to B \to \mathit{iProp}$ is defined as

$$R \hookrightarrow R' \triangleq \lambda f, g.\, \forall x, y.\, R\, x\, y \twoheadrightarrow R'\,(f\, x)\,(g\, x)$$

Definition 3.10 (*Persistent relation*)

The persistent relation $\Box\, R\colon A \to A \to \mathit{iProp}$ for a relation $R\colon A \to$

$A \to iProp$ is defined as

$$\square\, R \triangleq \lambda x, y.\ \square(R\, x\, y)$$

- We can rewrite lemma 3.6 using the relations we described above

Lemma 3.11 (*Separating conjuction monotone*)

The separating conjunction is a proper element of the relation $(\square \twoheadrightarrow\!\!* \hookrightarrow \square \twoheadrightarrow\!\!* \hookrightarrow \twoheadrightarrow\!\!*)$

- Writing out the above statement gives

$$\vdash \forall P, Q.\ \square(P \twoheadrightarrow\!\!* Q) \twoheadrightarrow\!\!* \forall P', Q'.\ \square(P' \twoheadrightarrow\!\!* Q') \twoheadrightarrow\!\!* P * Q \twoheadrightarrow\!\!* P' * Q'$$

- This is monotonicity on the left and right side of the separating conjunction at the same time

Definition 3.12 (*Pointwise relation*)

The pointwise relation $\bullet R$ is a special case of a respectful relation defined as

$$\bullet R \triangleq (= \hookrightarrow R)$$

Lemma 3.13 (*Existential quantification monotone*)

The existential quantification is a proper element of the relation

$$(\square \bullet \twoheadrightarrow\!\!* \hookrightarrow \twoheadrightarrow\!\!*)$$

Example 3.14 ($\mathsf{isMLL_F}$ *is monotone*)

The predicate $\mathsf{isMLL_F}$ is monotone in its first argument. Thus, $\mathsf{isMLL_F}$ is a proper element of

$$\square \bullet\bullet \twoheadrightarrow\!\!* \hookrightarrow \bullet\bullet \twoheadrightarrow\!\!*$$

$$\square\, (\forall hd\, \vec{v}.\, \Phi\, hd\, \vec{v} \twoheadrightarrow\!\!* \Psi\, hd\, \vec{v}) \twoheadrightarrow\!\!* \forall hd\, \vec{v}.\, \mathsf{isMLL_F}\, \Phi\, hd\, \vec{v} \twoheadrightarrow\!\!* \mathsf{isMLL_F}\, \Psi\, hd\, \vec{v}$$

Proof. We assume $\square\, (\forall hd\, \vec{v}.\, \Phi\, hd\, \vec{v} \twoheadrightarrow\!\!* \Psi\, hd\, \vec{v})$ holds and for arbitrary hd and $\vec{v}$, $\mathsf{isMLL_F}\, \Phi\, hd\, \vec{v}$ holds. After applying the definition of $\mathsf{isMLL_F}$ we need to prove

$$\mathsf{isMLL_F}\, \Psi\, hd\, \vec{v}$$

$\square$