Chapter 3

Fixpoints for representation predicates

3.1 Finite predicates and functors

- The logic described here is embedded in Coq.
- This has as result that any recursive predicate has to terminate.
- The type for locations is defined as $Loc = \mathbb{Z}$
- Thus we can have infinite locations
- Our predicates like, isMLL when they are recursively defined, as in section 2.5, don't necessarily terminate
- Since we cannot define recursive predicates, we instead need to find another way.
- The approach first introduced in [**?**] will be to take the least fixpoint of a monotone functor
- Take this simple recursive definition, that does not always terminate

$$\mathsf{isList}\ hd \triangleq hd = \mathbf{none} \vee hd = \mathbf{some}\ \ell * \ell \mapsto (v, tl) * \mathsf{isList}\ tl$$

- This predicate simply checks if a pointer points to a linked list.
- It checks if either the pointer is a null pointer, or it points to a node containing a value and a pointer to the rest of the list.
- We can transform this definition into a functor acting on possible isList predicates, by replacing the recursive call with an argument

Question: In logic it is not really a call, what is it?

$$\mathsf{isList_F}\,\Phi\,hd \triangleq hd = \mathbf{none} \lor hd = \mathbf{some}\,\ell * \ell \mapsto (v, tl) * \Phi\,tl$$

- Here, Φ is a predicate of type $Val \to iProp$
- And, $\mathsf{isList_F}$ is a predicate of type $(Val \to iProp) \to (Val \to iProp)$
- Any fixpoint Φ of $\mathsf{isList_F}$ would have the following property after filling in the definition of $\mathsf{isList_F}$

$$\forall hd.\, (hd = \mathbf{none} \lor hd = \mathbf{some}\,\ell * \ell \mapsto (v, tl) * \Phi\,tl) \mathrel{-\!\!*} \Phi\,hd$$

- Thus, if $\mathsf{F}\,hd$ holds we know at depth one, hd is a linked list.

$$\begin{array}{rcl} & & hd = \mathbf{none} * \vec{v} = [] \\ \mathsf{isMLL}\,hd\,\vec{v} = & \lor & hd = \mathbf{some}\,l * l \mapsto (v, \mathbf{true}, tl) * \mathsf{isMLL}\,tl\,\vec{v} \\ & \lor & hd = \mathbf{some}\,l * \vec{v} = [v'] + \vec{v}'' * l \mapsto (v', \mathbf{false}, tl) * \mathsf{isMLL}\,tl\,\vec{v}'' \end{array}$$

- We first turn our desired predicate into a functor
- It transforms a predicate Φ into a predicate that applies Φ to the tail of the MLL if it exists

TODO: write more correct

$$\begin{array}{rcl} & & hd = \mathbf{none} * \vec{v} = [] \\ \mathsf{isMLL_F}\,\Phi\,hd\,\vec{v} \triangleq & \lor & hd = \mathbf{some}\,l * l \mapsto (v', \mathbf{true}, tl) * \Phi\,tl\,\vec{v} \\ & \lor & hd = \mathbf{some}\,l * \vec{v} = [v'] + \vec{v}'' * l \mapsto (v', \mathbf{false}, tl) * \Phi\,tl\,\vec{v}'' \end{array}$$

3.2 Monotone predicates

-

Definition 3.1 (*Monotone predicate*)

We define a monotone predicate for two different arities, however it can be expanded to any arity monotone predicate and arity argument, as long as the arity of the predicate is at most one higher than that of the argument.

Arity one predicate, arity zero arguments: Any $\Phi\colon A \to iProp$ is monotone when for any $P, Q\colon iProp$, it holds that

$$\vdash \square(P \mathrel{-\!\!*} Q) \mathrel{-\!\!*} \Phi P \mathrel{-\!\!*} \Phi Q$$

Arity two predicate, arity one arguments: Any $\mathsf{F}\colon (A \to iProp) \to A \to iProp$ is monotone when for any $\Phi, \Psi\colon A \to iProp$, it holds that

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

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

$$\Box\,(P \mathrel{-\!\!*} Q) * \Phi P \vdash \Phi Q$$

- 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
- We want to eventually prove certain predicates are monotone
- To accomplish this we first prove a few different connectives monotone
- We show the full prove for the separating conjunction
- The other connectives are proven in similar ways.

Lemma 3.2 (*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. expading the definition of monotone for arity one we get the following statement.

$$\vdash \Box(P \mathrel{-\!\!*} Q) \mathrel{-\!\!*} P * R \mathrel{-\!\!*} Q * R$$

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

Lemma 3.3 (*Magic wand monotone*)

The magic wand is monotone in its conclusion.

$$\vdash \Box\,(P \mathrel{-\!\!*} Q) \mathrel{-\!\!*} (R \mathrel{-\!\!*} P) \mathrel{-\!\!*} (R \mathrel{-\!\!*} Q)$$

Lemma 3.4 (*Disjunction monotone*)

The disjunction is monotone in its left and right argument.

$$\vdash \Box\,(P \mathrel{-\!\!*} Q) \mathrel{-\!\!*} (P \vee R) \mathrel{-\!\!*} (Q \vee R)$$

$$\vdash \Box\,(P \mathrel{-\!\!*} Q) \mathrel{-\!\!*} (R \vee P) \mathrel{-\!\!*} (R \vee Q)$$

Lemma 3.5 (*Universal quantification monotone*)

The universal quantification is monotone.

$$\Box\,(\forall x.\,\Phi(x) \mathrel{-\!\!*} \Psi(x)) \mathrel{-\!\!*} (\forall x.\,\Phi(x)) \mathrel{-\!\!*} (\forall x.\,\Psi(x))$$

Lemma 3.6 (*Existential quantification monotone*)

The existential quantification is monotone.

$$\Box\,(\exists x.\,\Phi(x) \mathrel{-\!\!*} \Psi(x)) \mathrel{-\!\!*} (\exists x.\,\Phi(x)) \mathrel{-\!\!*} (\exists x.\,\Psi(x))$$

- The last two lemmas, about the two quantification, both used the arity one predicate and arity one argument version of monotone
- They thus had equal arity, whereas up until now all monotone predicates used a predicate with arity one higher than the argument.
- A predicate is also monotone in any arguments in which it is constant
- Thus, a predicate $\Phi(x) = \mathsf{False}$ is monotone in x.

Lemma 3.7 (*Constant monotone*)

A predicate $\Phi\colon A \to iProp$ which is constant in respects to its first argument, thus $\Phi(x) = P$, where x does not occur in P, is monotone in that argument.

$$\Box\,(Q \mathrel{-\!\!*} R) \mathrel{-\!\!*} P \mathrel{-\!\!*} P$$

- The magic wand is not monotone in its first argument, however, we can still say something useful about it.
- The magic wand is downwards monotone in its first argument.

Definition 3.8 (*Downward monotone*)

We define a downward monotone predicate for two different arities, however it can be expanded to any arity monotone predicate and arity argument, as long as the arity of the predicate is at most one higher than that of the argument.

Arity one predicate, arity zero arguments: Any $\Phi\colon A \to iProp$ is monotone when for any $P, Q\colon iProp$, it holds that

$$\vdash \Box(Q \mathrel{-\!\!*} P) \mathrel{-\!\!*} \Phi P \mathrel{-\!\!*} \Phi Q$$

Arity two predicate, arity one arguments: Any $\mathsf{F}\colon (A \to iProp) \to$

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

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

- The definition of downward monotone is exactly the same as upwards monotone, just with its first condition flipped.
- A monotone predicate is sometimes also called an upwards monotone predicate
- Using the downwards monotone property we can give a useful lemma about the assumption of the magic wand.

Lemma 3.9 (*Magic wand downward monotone*)

The magic wand is downwards monotone in its assumption.

$$\vdash \Box\,(Q \mathrel{-\!\!*} P) \mathrel{-\!\!*} (P \mathrel{-\!\!*} R) \mathrel{-\!\!*} (P \mathrel{-\!\!*} R)$$

Question: I don't have an example using downwards monotone, but it is used in for example the twp proof, thus should I include it?

- Any composite predicate made from the connectives we have lemmas about are now easily proven monotone.

Example 3.10 ($\mathsf{isMLL_F}$ *is monotone*)

The predicate $\mathsf{isMLL_F}$ is monotone in its first argument.

$$\Box\,(\forall hd\, \vec{v}.\, \Phi\, hd\, \vec{v} \mathrel{-\!\!*} \Psi\, hd\, \vec{v}) \mathrel{-\!\!*} \forall hd\, \vec{v}.\, \mathsf{isMLL_F}\, \Phi\, hd\, \vec{v} \mathrel{-\!\!*} \mathsf{isMLL_F}\, \Psi\, hd\, \vec{v}$$

Proof. We assume $\Box\,(\forall hd\, \vec{v}.\, \Phi\, hd\, \vec{v} \mathrel{-\!\!*} \Psi\, hd\, \vec{v})$ 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

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

We apply lemma 3.4 twice in order to get three predicates we need to prove are monotone □

3.3 Fixpoints of functors

- This gets rid of the possible infinite nature of the statement
- but not strong enough
- we want to find a Φ such that

$$\forall hd\ \vec{v}.\ \mathsf{isMLLPre}\,\Phi\ hd\ \vec{v} \mathrel{-\!\!*\!\!*} \Phi\ hd\ \vec{v}$$

- This is the fixpoint of isMLLPre
- Use Knaster-Tarski Fixpoint Theorem to find this fixpoint [Tar55]
- Specialized to the lattice on predicates

Theorem 3.11 (*Knaster-Tarski Fixpoint Theorem*)

Let $\mathsf{F}\colon (A \to iProp) \to (A \to iProp)$ *be a monotone predicate, then*

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

defines the least fixpoint of F

Question: Where to introduce *iProp*?

- Monotone is defined as
- In general F is monotone if all occurrences of its Φ are positive
- This is the case for isMLL
- We can expand theorem 3.11 to predicates of type $\mathsf{F}\colon (A \to B \to iProp) \to (A \to B \to iProp)$
- Thus the fixpoint exists and is

$$\mathsf{lfp}\,\mathsf{isMLLPre}\ hd\ \vec{v} = \forall \Phi.\ (\forall hd'\ \vec{v}'.\ \mathsf{isMLLPre}\,\Phi\ hd'\ \vec{v}' \mathrel{-\!\!*} \Phi\ hd'\ \vec{v}') \mathrel{-\!\!*} \Phi\ hd\ \vec{v}$$

- We can now redefine isMLL as

$$\mathsf{isMLL}\ hd\ \vec{v} \triangleq \mathsf{lfp}\,\mathsf{isMLLPre}\ hd\ \vec{v}$$

- Using the least fixpoint we can now define some additional lemmas

Lemma 3.12 (lfp F *is the least fixpoint on* F)

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

$$\forall x.\ \mathsf{F}\,(\mathsf{lfp}\,\mathsf{F})\ x \mathrel{-\!\!*\!\!*} \mathsf{lfp}\,\mathsf{F}\ x$$

3.4 Induction principle

Lemma 3.13 (*least fixpoint induction principle*)

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

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

Lemma 3.14 (*least fixpoint strong induction principle*)

Given a monotone $\mathsf{F} : (A \to \mathit{iProp}) \to (A \to \mathit{iProp})$*, it holds that*

$$\Box(\forall x.\, \mathsf{F}\,(\lambda y.\, \Phi\, y \wedge \mathsf{lfp}\,\mathsf{F}\, y)\, x \mathrel{-\!\!*} \Phi\, x) \mathrel{-\!\!*} \forall x.\, \mathsf{lfp}\,\mathsf{F}\, x \mathrel{-\!\!*} \Phi\, x$$