Fixed Points for Markov Decision Processes

Johannes Hölzl *
TU München
hoelzl@in.tum.de

Abstract

The goal of this paper is to advertise the application of fixed points and ω-complete partial orders (ω-cpos) in the formalization and analysis of probabilistic programming languages. The presented work is formalized in the Isabelle theorem prover.

By applying ω-cpos to the analysis of MDPs we get a nice theory of fixed points. This allows us to transfer least and greatest fixed points through expectation on Markov chains and maximal and minimal expectation on MDPs. One application is to define the operational semantics of pGCL by MDPs, e.g. relating the denotational and operational semantics of pGCL is now a matter of fixed point equations and induction.

1. Introduction

It is standard to use fixed points to describe solutions for probabilistic model checking [2], or to give semantics for probabilistic programming languages as fixed points [5]. This is similar to the work by Monniaux [7]. Both are fixed points on complete lattices (either $[0,1]$ or $\overline{\mathbb{R}}^+$, the non-negative reals with ∞). We are concerned with a more general class, namely ω-complete partial orders (ω-cpos). Here fixed points are only well-defined for continuous functions. A function F is sup-continuous (and inf-continuous) if for all increasing (and decreasing) sequences X, the function F commutes with supremum (and infimum) of X:

$$F(\bigsqcup_i X_i) = \bigsqcup_i F(X_i) \quad \text{and} \quad F(\bigsqcap_i X_i) = \bigsqcap_i F(X_i). \tag{1}$$

The least (and greatest) fixed point of F is equal to the supremum (and infimum) of the iterations of F, when F is continuous:

$$\operatorname{lfp} F = \bigsqcup_i F^i(\bot) \qquad \operatorname{gfp} F = \bigsqcap_i F^i(\top) \tag{2}$$

Hence for continuous functions F, G, α on ω-cpos we derive a *transfer rule* similar to the fusion rule in [4]:

$$\frac{\alpha(\bot) = \bot \qquad \alpha \circ G = F \circ \alpha}{\alpha(\operatorname{lfp} G) = \operatorname{lfp} F} \tag{3}$$

We get a dual rule for greatest fixed points.

We need ω-cpos as they are compatible with measure theory: the Borel-measurable functions $\mathcal{M}\left(\mathcal{A}, \mathcal{B}\left(\overline{\mathbb{R}}^+\right)\right)$ are not a complete lattice, but a ω-cpo, for each measurable spaces $\mathcal{A}$. We know that least (and greatest) fixed points $\operatorname{lfp} F$ (and $\operatorname{gfp} F$) are measurable assuming the functional F is continuous and measurable:

$$F \in \mathcal{M}\left(\mathcal{A}, \mathcal{B}\left(\overline{\mathbb{R}}^+\right)\right) \to \mathcal{M}\left(\mathcal{A}, \mathcal{B}\left(\overline{\mathbb{R}}^+\right)\right) \tag{4}$$

In the rest of this paper we always assume that functions are measurable. From Lebesgue's monotone convergence theorem follows that Lebesgue integration is a continuous function. There is a similar monotone convergence theorem for decreasing sequences. However, this rules requires that the integral is finite starting at some point i in the sequence X_i. We derive the following transfer rule on the Lebesgue integral:

$$\frac{\begin{array}{c}F, G \text{ sup-continuous} \\ \forall s, f \le \operatorname{lfp} F.\ \int F\ f\ dM_s = G\ (\lambda s.\ \int\ f\ dM_s)\end{array}}{\int \operatorname{lfp} F\ dM_s = \operatorname{lfp} G\ s} \tag{5}$$

Compared to Rule (3) we provide it with a stronger equation (restricted to $f \le \operatorname{lfp} F$), as this is necessary for certain proofs. Also adding the index s to the family of probability measures M_s will be very handy when working on the trace spaces of Markov chains and MDPs. We also assume measurability of F as the one in Rule (4). The rule for the greatest fixed point is very similar, however it requires that the integral is always finite.

$$\frac{\begin{array}{cc}F, G \text{ inf-continuous} & \forall f, s.\ \int F\ f\ dM_s < \infty \\ \multicolumn{2}{c}{\forall s, f \ge \operatorname{gfp} F.\ \int F\ f\ dM_s = G\ (\lambda s.\ \int\ f\ dM_s)}\end{array}}{\int \operatorname{gfp} F\ dM_s = \operatorname{gfp} G\ s} \tag{6}$$

When solving integrals of fixed points we can now apply Rule (5) or Rule (6) and then continue with fixed point equations like rolling, diagonalisation, iteration, or induction [4]. In the following sections we apply these rules on Markov chains, MDPs and for the semantics of pGCL.

2. Markov chains

A Markov chain is given by its Markov kernel $K : A \to \mathcal{D}(A)$, where $\mathcal{D}(A)$ is a discrete probability measure on A. Given a kernel K we construct the trace space T_s (e.g. by the Ionescu-Tulcea extension theorem [8] or similar) starting in state s with the iteration property:

$$\int f\, dT_s = \int_t \left(\int_\omega f(t{\cdot}\omega) d\, T_t\right) d\, K_s \tag{7}$$

Here $t{\cdot}\omega$ prepends t to the sequence ω. Using least and greatest fixed points we define the usual LTL operators, extending the work from Blanchette et al. [3], but also first hitting time or cumulated rewards. For example eventually $\Diamond$ and first hitting time h are defined as follows:

$$\begin{array}{rcl}\Diamond(X) & = & \operatorname{lfp}\ (\lambda F.\ \lambda(t{\cdot}\omega).\ \text{if } t \in X \text{ then } 1 \text{ else } F(\omega)) \\ h(X) & = & \operatorname{lfp}\ (\lambda F.\ \lambda(t{\cdot}\omega).\ \text{if } t \in X \text{ then } 0 \text{ else } 1 + F(\omega))\end{array}$$

As the functionals in the definitions are continuous and measurable, it follows that $\Diamond$ and h are also measurable functions. With the transfer rule (3) we get the linear equation system known from probabilistic model checking [2].

The transfer rules are also used in proofs, for example to prove fairness (oiginally from [1]) we show that the not fairness set is expressed as gfp, and bound from above by 0. Similarly, we prove that hitting time $h(X)$ is finite, assuming a finite Markov chain, and that the set X is reachable.

* Supported by the DFG Projekt NI 491/15-1

3. Markov decision processes

A Markov decision process (MDP) is similarly given as a kernel $K : A \to \mathcal{P}(\mathcal{D}(A))$. This time the trace space is parameterized by a scheduler $\sigma(s)$ starting in s.

$$\int f \, d\,T_{\sigma(s)} = \int_t \left(\int_\omega f(t{\cdot}\omega) d\,T_{\sigma(s,t)} \right) d\,K_{\sigma(s)} \tag{8}$$

Here $\sigma(s,t)$ is the continuation of the scheduler $\sigma(s)$ going to t, and $K_{\sigma(s)}$ is a distribution in K_s chosen by the scheduler $\sigma(s)$. Now the minimal and maximal expectation is defined as:

$$E^{\min}_s(f) = \bigsqcap_{\sigma(s)} \int f \, d\,T_{\sigma(s)} \quad E^{\max}_s(f) = \bigsqcup_{\sigma(s)} \int f \, d\,T_{\sigma(s)} \tag{9}$$

For these we get also iterative rules similar to Rule (7):

$$E^{\max}_s(f) = \bigsqcup_{\mu \in K\ s} \int_t E^{\max}_t(\omega.\ f(t{\cdot}\omega))\ d\mu \tag{10}$$

For $E^{\min}$ we get transfer rule for lfp, and for $E^{\max}$ we get the transfer rule only when the sets K_s are finite for all s. From this again we get directly the fixed point we can use to get the linear equation system used for model checking.

4. pGCL Semantics

The probabilistic guarded command language (pGCL) is a imperative language with probabilistic and non-deterministic choice. Gretz et al. [5] compare the usual (denotational) expectation transformer semantics with operational semantics. As operational semantics they define for each pGCL program a MDP. Then finally they show that the reward of the MDP equals the expectation of the expectation transformer. The proof works by representing the expected reward as the sum over all possible paths. In this section we want to show an alternative proof, representing the reward as a fixed point. First we present the pGCL semantics similar to [5]:

$$\begin{array}{rcl} \mathit{pgcl} & := & \mathit{Skip} \mid \mathit{Abort} \mid \mathit{Assign}\ (\sigma \to \sigma) \\ & \mid & \mathit{Seq}\ \mathit{pgcl}\ \mathit{pgcl} \mid \mathit{Par}\ \mathit{pgcl}\ \mathit{pgcl} \\ & \mid & \mathit{If}\ (\sigma \to \mathrm{bool})\ \mathit{pgcl}\ \mathit{pgcl} \\ & \mid & \mathit{Prob}\ [0,1]\ \mathit{pgcl}\ \mathit{pgcl} \\ & \mid & \mathit{While}\ (\sigma \to \mathit{bool})\ \mathit{pgcl}\ \mathit{pgcl} \end{array}$$

The weakest pre-expectation $wp\ c\ f$ is an expectation transformer recursively computed by the program structure of c:

$$\begin{array}{l} wp : \mathit{pgcl} \to \left(\sigma \to \overline{\mathbb{R}}^+\right) \to \left(\sigma \to \overline{\mathbb{R}}^+\right) \\ wp\ \mathit{Skip}\ f = f \\ wp\ \mathit{Abort}\ f = \bot \\ wp\ (\mathit{Assign}\ u)\ f = f \circ u \\ wp\ (\mathit{Seq}\ c_1\ c_2)\ f = wp\ c_1\ (wp\ c_2\ f) \\ wp\ (\mathit{Par}\ c_1\ c_2)\ f = wp\ c_1\ f \sqcap wp\ c_2\ f \\ wp\ (\mathit{If}\ b\ c_1\ c_2)\ f = \\ \quad \lambda s.\ \text{if } b\ s \text{ then } wp\ c_1\ f\ s \text{ else } wp\ c_2\ f\ s \\ wp\ (\mathit{Prob}\ p\ c_1\ c_2)\ f = \\ \quad \lambda s.\ p \cdot wp\ c_1\ f\ s + (1-p) \cdot wp\ c_2\ f\ s \\ wp\ (\mathit{While}\ b\ c)\ f = \\ \quad \mathrm{lfp}\ (\lambda X\ s.\ \text{if } b\ s \text{ then } wp\ c\ X\ s \text{ else } f\ s) \end{array}$$

The operational semantics are K used to construct the MDP:

$$\begin{array}{l} K : (\mathit{pgcl} \times \sigma) \to \mathcal{P}(\mathcal{D}(\mathit{pgcl} \times \sigma)) \\ K\ (\mathit{Skip}, s) = \ll \mathit{Skip}, s \gg \\ K\ (\mathit{Abort}, s) = \ll \mathit{Abort}, s \gg \\ K\ (\mathit{Assign}\ u, s) = \ll \mathit{Skip}, u\ s \gg \\ K\ (\mathit{Seq}\ c_1\ c_2, s) = \\ \quad [K(c_1, s)]\ (\lambda(c_1', s').\ \text{if } c_1' = \mathit{Skip} \text{ then } c_2 \text{ else } \mathit{Seq}\ c_1'\ c_2, s') \\ K\ (\mathit{Par}\ c_1\ c_2, s) = \ll c_1, s \gg \cup \ll c_2, s \gg \\ K\ (\mathit{If}\ b\ c_1\ c_2, s) = \text{if } b\ s \text{ then } K(c_1, s) \text{ else } K(c_2, s) \\ K\ (\mathit{Prob}\ p\ c_1\ c_2, s) = \{\{p \mapsto (c_1, s), (1-p) \mapsto (c_2, s)\}\} \\ K\ (\mathit{While}\ b\ c, s) = \\ \quad \mathit{if}\ b\ s\ \mathit{then} \ll \mathit{Seq}\ c\ (\mathit{While}\ b\ c), s \gg\ \mathit{else} \ll \mathit{Skip}, s \gg \end{array}$$

Here $\ll c, s \gg$ is the singleton set, containing the Dirac distribution over (c, s). The syntax $[K]f$ maps the elements of K over f, i.e. if K has type $\mathcal{P}(\mathcal{D}(A))$ and $f : A \to B$, then the result type is $\mathcal{P}(\mathcal{D}(B))$.

Next we define the reward of f on a trace as a least fixed point:

$$r\ f = \mathrm{lfp}\ (\lambda F\ ((c, s){\cdot}\omega).\ \text{if } c = \mathit{Skip} \text{ then } f(s) \text{ else } F(\omega))$$

Using the transfer rule on $E^{\min}$ we get the following equation:

$$\begin{array}{l} E^{\min}_x\ (r\ f) = \mathrm{lfp}\ (\lambda F\ x. \\ \quad \bigsqcap_{\mu \in K\ x} \int_{(c,s)} \text{if } c = \mathit{Skip} \text{ then } f(s) \text{ else } F(c, s)\ d\mu)\ x \end{array} \tag{11}$$

With this we finally prove:

$$E^{\min}_{(c,s)}(r\ f) = wp\ c\ f\ s \tag{12}$$

Before we start the proof by induction, we prove the Seq-case:

$$E^{\min}_{(\mathit{Seq}\ c_1\ c_2, s)}(r\ f) = E^{\min}_{(c_1, s)}\left(r\ \left(\lambda s'.\ E^{\min}_{(c_2, s')}(r\ f)\right)\right)$$

This is proved by rewriting with Eq. (11) and then applying fixed point induction in both directions.

Now we prove Eq. (12), by induction over c and generalizing in s and f. The only interesting case is the $(\mathrm{While}\ g\ c)$-case. That wp is less or equal $E^{\min}$ follows from the fact that $E^{\min}$ fullfills already the fixed point equation. For the other direction we unroll the while-loop to the following functional:

$$\begin{array}{l} w\ F\ s = \bigsqcap_{\mu \in K\ x} \int_{(d,s)} \text{if } d = \mathit{Skip} \text{ then if } g\ s \text{ then } F(c, s) \\ \quad \text{else } f(s) \text{ else } F(d, s)\ d\mu \end{array}$$

We finish with $E^{\max}_{(\mathit{While}\ g\ c, s)}(r\ f) \leq \mathrm{lfp}\ w\ (c, s) = \mathrm{wp}\ (\mathit{While}\ g\ c)\ s$, where first part is proved by induction and the second by loop unrolling and the diagonal rule for least fixed points.

5. Formalization in Isabelle

The theory presented in this paper is formalized in Isabelle, and can be found in the AFP entry `Markov_Models` [6]. Compared to the presentation in this paper, many statements in Isabelle are more complicated, as we need to explicitly state measurability. Otherwise, the fixed point theorems on Markov chains and MDPs simplified the formalization considerable as many ε-proofs are now reduced to induction on fixed points, or even direct rewriting by fixed point equations. Proofs like fairness on Markov chains or finite hitting time on finite Markov chains were quite technical before and got more concise by using fixed point equations.

References

[1] C. Baier. *On the Algorithmic Verification of Probabilistic Systems*. Habilitation, Universität Mannheim, 1998.

[2] C. Baier and J.-P. Katoen. *Principles of Model Checking*. The MIT Press, Cambridge, Massachusetts, 2008.

[3] J. C. Blanchette, A. Popescu, and D. Traytel. Unified classical logic completeness. In S. Demri, D. Kapur, and C. Weidenbach, editors, *Automated Reasoning*, volume 8562 of *LNCS*, pages 46–60. 2014.

[4] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, second edition, 2002.

[5] F. Gretz, J. Katoen, and A. McIver. Operational versus weakest pre-expectation semantics for the probabilistic guarded command language. *Performance Evaluation*, 73:110–132, 2014.

[6] J. Hölzl and T. Nipkow. Markov models. *The Archive of Formal Proofs*, Jan 2012. `http://afp.sf.net/entries/Markov_Models.shtml`.

[7] D. Monniaux. Abstract interpretation of programs as Markov decision processes. In *SAS 2003*, volume 58 of *Science of Computer Programming*, pages 179–205. 2005.

[8] D. Pollard. *A Users's Guide to Measure Theoretic Probability*. Cambridge Series in Statistical and Probabilistic Mathematics. 2002.