Algebra of Iterative Constructions

Kevin Batz Benjamin Lucien Kaminski Lucas Kehrer
Gerwin Klein Henning Urbat Todd Schmid

May 8, 2026

Abstract

Fixed points are a recurring theme in computer science and are often constructed as limits of suitably seeded fixed point iterations. This entry formalises an instance of the algebra of iterative constructions (AIC) from the paper “The Algebra of Iterative Constructions”.

AIC is a purely algebraic approach to reasoning about fixed point iterations of continuous endomaps on complete lattices. AIC allows derivations of constructive fixed point theorems via equational logic and avoids explicit computations with indices. We demonstrate the applicability of AIC by providing algebraic proofs of several well- and less-well-known fixed point theorems: Among others, we prove the *Tarski-Kantorovich principle* – a generalization of the *Kleene fixed point theorem* – as well as a fixed point-theoretic generalization of k-induction.

We moreover improve upon a recent generalization of the Tarski-Kantorovich principle due to Olszewski for obtaining pre- and postfixed points from lattice-theoretic limit inferiors and limit superiors trough iterating an endomap on an *arbitrary* seed element: We identify sufficient continuity conditions on the endomaps so that these limits become *proper* fixed points.

theory *Iteration*
  imports *Main*
begin

Make Isabelle lattice syntax available

unbundle *lattice-syntax*

1 Operators as defined in section 2 of the paper

Formalise sequences as functions from *nat* to lattice elements

type-synonym $'a\ seq = nat \Rightarrow\ 'a$

We directly reuse Isabelle's $\bot$, $\top$, $\sqcup$, and $\sqcap$ on functions for the corresponding operators on sequences. They do not need a separate definition.

definition *flat* :: *'a seq* ⇒ *'a::complete-lattice seq*
(‹⊙-› *[80] 81*) **where**
$\odot a \equiv \lambda n.\ a\ 0$

definition *next-op* :: *'a seq* ⇒ *'a::complete-lattice seq*
(‹○-› *[80] 81*) **where**
$\bigcirc a \equiv \lambda n.\ a\ (Suc\ n)$

definition *diamond* :: *'a seq* ⇒ *'a::complete-lattice seq*
(‹◇-› *[80] 81*) **where**
$\Diamond a \equiv \lambda n.\ \bigsqcup \{a\ k | k.\ n \leq k\}$

definition *box* :: *'a seq* ⇒ *'a::complete-lattice seq*
(‹□-› *[80] 81*) **where**
$\Box a \equiv \lambda n.\ \bigsqcap \{a\ k | k.\ n \leq k\}$

definition *app* :: (*'a* ⇒ *'a*) ⇒ *'a seq* ⇒ *'a::complete-lattice seq*
(**infix** $ *70*) **where**
$F\ \$\ a \equiv \lambda n.\ F\ (a\ n)$

definition *orbit* :: (*'a* ⇒ *'a*) ⇒ *'a seq* ⇒ *'a::complete-lattice seq*
(**infix** ⋆ *100*) **where**
$F\star\ a \equiv \lambda n.\ (F\widehat{\ }\widehat{\ }n)\ (a\ n)$

term $\Box\Diamond\ a = \Diamond\Box\ a$
term $F\ \$\ a = a$
term $F\star\ a = a$
term $F\ \$\ \Diamond\ F\star\ \bot$

Iterating a function F for n times applied to sequence a:

term $(F\ \widehat{\ }\widehat{\ }\ n)\ \$\ a$

1.1 Paper Section 2.3

We reuse HOL equality, which means the equality axioms in the paper are already available as Isabelle lemmas or axioms, e.g. reflexivity, transitivity, congruence, substitution, etc.

1.2 Paper Section 2.4

The fact that sequences form a lattice is already available in Isabelle and we can reuse the predefined ≤ order in Isabelle.

abbreviation *seq-le* :: *'a seq* ⇒ *'a::complete-lattice seq* ⇒ *bool*
((-/ ⪯ -) *[51, 51] 50*) **where**
$a \preceq b \equiv a \leq b$

term $a \preceq \bigcirc a$

lemma $a \preceq b = (a \sqcup b = b)$ **for** a::$'a$::*complete-lattice seq*
⟨*proof*⟩

2 Paper Section 3

Axioms and derived rules about the main operators, derived as lemmas in Isabelle.

Some of the basic sequence axioms already exist as lattice facts in Isabelle. These are shown here for completeness using the method "fact".

2.1 Bounded Lattice

lemma *sup-com*:
$a \sqcup b = b \sqcup (a :: 'a$::*complete-lattice seq*)
⟨*proof*⟩

lemma *inf-com*:
$a \sqcap b = b \sqcap (a :: 'a$::*complete-lattice seq*)
⟨*proof*⟩

lemma *sup-assoc*:
$(a \sqcup b) \sqcup c = a \sqcup (b \sqcup c)$ **for** $a :: 'a$::*complete-lattice seq*
⟨*proof*⟩

lemma *inf-assoc*:
$(a \sqcap b) \sqcap c = a \sqcap (b \sqcap c)$ **for** $a :: 'a$::*complete-lattice seq*
⟨*proof*⟩

lemma *sup-absorb*:
$a \sqcup (a \sqcap b) = a$ **for** $a :: 'a$::*complete-lattice seq*
⟨*proof*⟩

lemma *inf-absorb*:
$a \sqcap (a \sqcup b) = a$ **for** $a :: 'a$::*complete-lattice seq*
⟨*proof*⟩

lemma *bot*[*simp*, *intro*!]:
$\bot \preceq a$
⟨*proof*⟩

lemma *top*[*simp*, *intro*!]:
$a \preceq \top$
⟨*proof*⟩

Additional Partial Order and Lattice Axioms from Figures 2 and 3 in

Sect 3.

lemma *reflex*:
$a \preceq a$
⟨*proof*⟩

lemma *trans*:
$[\![a \preceq b; b \preceq c]\!] \Longrightarrow a \preceq c$
⟨*proof*⟩

lemma *antisymm*:
$[\![a \preceq b; b \preceq a]\!] \Longrightarrow a = b$
⟨*proof*⟩

lemma *weakenR*:
$a = b \Longrightarrow a \preceq b$
⟨*proof*⟩

lemma *weakenL*:
$a = b \Longrightarrow b \preceq a$
⟨*proof*⟩

lemma *sup-idem*:
$a \sqcup a = a$ **for** a :: $'a$::*complete-lattice seq*
⟨*proof*⟩

lemma *inf-idem*:
$a \sqcap a = a$ **for** a :: $'a$::*complete-lattice seq*
⟨*proof*⟩

lemma *sup-introL*:
$[\![a \preceq c; b \preceq c]\!] \Longrightarrow a \sqcup b \preceq c$
⟨*proof*⟩

lemma *sup-introR*:
$a \preceq b \Longrightarrow a \preceq b \sqcup c$
⟨*proof*⟩

lemma *inf-introL*:
$b \preceq c \Longrightarrow a \sqcap b \preceq c$
⟨*proof*⟩

lemma *inf-introR*:
$[\![a \preceq b; a \preceq c]\!] \Longrightarrow a \preceq b \sqcap c$
⟨*proof*⟩

lemma *sup-elim*:
$a \sqcup b \preceq b \Longrightarrow a \preceq b$ **for** a :: $'a$::*complete-lattice seq*
⟨*proof*⟩

lemma *inf-elim*:
$a \preceq a \sqcap b \Longrightarrow a \preceq b$ **for** a :: $'a$::*complete-lattice seq*
$\langle proof \rangle$

3 Shifts, Fig 3, Section 3

lemma *next-mono*:
$a \preceq b \Longrightarrow \bigcirc a \preceq \bigcirc b$
$\langle proof \rangle$

lemma *next-of-bot*:
$\bigcirc\bot \preceq \bot$
$\langle proof \rangle$

lemma *next-of-top*:
$\top \preceq \bigcirc\top$
$\langle proof \rangle$

lemma *next-over-sup*:
$\bigcirc(a \sqcup b) = \bigcirc a \sqcup \bigcirc b$
$\langle proof \rangle$

lemma *next-over-inf*:
$\bigcirc(a \sqcap b) = \bigcirc a \sqcap \bigcirc b$
$\langle proof \rangle$

lemma *diamond-inflate*[*simp*, *intro*!]:
$a \preceq \Diamond a$
$\langle proof \rangle$

lemma *box-deflate*[*simp*, *intro*!]:
$\Box a \preceq a$
$\langle proof \rangle$

lemma *dd-le-d*:
$\Diamond\Diamond a \preceq \Diamond a$
$\langle proof \rangle$

lemma *diamond-idem*[*simp*]:
$\Diamond\Diamond a = \Diamond a$
$\langle proof \rangle$

lemma *b-le-bb*:
$\Box a \preceq \Box\Box a$
$\langle proof \rangle$

lemma *box-idem*[*simp*]:
$\Box\Box a = \Box a$
$\langle proof \rangle$

lemma *diamond-mono*:
$a \preceq b \Longrightarrow \Diamond a \preceq \Diamond b$
$\langle proof \rangle$

lemma *box-mono*:
$a \preceq b \Longrightarrow \Box a \preceq \Box b$
$\langle proof \rangle$

lemma *next-diamond-comm*:
$\bigcirc\Diamond a = \Diamond\bigcirc a$
$\langle proof \rangle$

lemma *next-box-comm*:
$\bigcirc\Box a = \Box\bigcirc a$
$\langle proof \rangle$

lemma *next-induct'*:
$\Diamond a \preceq a \Longrightarrow \bigcirc a \preceq a$
$\langle proof \rangle$

lemma *next-induct*:
$\bigcirc a \preceq a \Longrightarrow \Diamond a \preceq a$
$\langle proof \rangle$

lemma *next-coinduct'*:
$a \preceq \Box a \Longrightarrow a \preceq \bigcirc a$
$\langle proof \rangle$

lemma *next-coinduct*:
$a \preceq \bigcirc a \Longrightarrow a \preceq \Box a$
$\langle proof \rangle$

lemma *box-introL*:
$a \preceq b \Longrightarrow \Box a \preceq b$
$\langle proof \rangle$

lemma *diamond-introR*:
$a \preceq b \Longrightarrow a \preceq \Diamond b$
$\langle proof \rangle$

lemma *box-introR*:
$[\![a \preceq \bigcirc a; a \preceq b]\!] \Longrightarrow a \preceq \Box b$
$\langle proof \rangle$

lemma *diamond-introL*:
$[\![a \preceq b; \bigcirc b \preceq b]\!] \Longrightarrow \Diamond a \preceq b$
$\langle proof \rangle$

lemma *diamond-elim*:
$\Diamond a \preceq b \Longrightarrow a \preceq b$
⟨*proof*⟩

lemma *box-elim*:
$a \preceq \Box b \Longrightarrow a \preceq b$
⟨*proof*⟩

lemma *diamond-desc*:
$\bigcirc\Diamond a \preceq \Diamond a$
⟨*proof*⟩

lemma *box-asc*:
$\Box a \preceq \bigcirc\Box a$
⟨*proof*⟩

lemma *diamond-exp*:
$\Diamond a = a \sqcup \Diamond\bigcirc a$
⟨*proof*⟩

lemma *box-exp*:
$\Box a = a \sqcap \Box\bigcirc a$
⟨*proof*⟩

4 Function Applications and Iteration

lemma *F-next-comm*:
$F \ \$ \ \bigcirc a = \bigcirc \ (F \ \$ \ a)$
⟨*proof*⟩

Relating the characterisation of "monotone" for sequences with the built-in predicate "mono"

lemma *monotone-mono*:
$\forall\, a\ b.\ a \preceq b \longrightarrow F \ \$ \ a \preceq F \ \$ \ b \Longrightarrow mono\ F$
⟨*proof*⟩

Fixing a context *mono-F* in which we assume as a background fact that F is monotone

locale *mono-F* =
fixes $F :: 'a \Rightarrow 'a :: complete\text{-}lattice$
assumes *mono*: *mono F*
begin

lemma *F-mono*:
$a \preceq b \Longrightarrow F \ \$ \ a \preceq F \ \$ \ b$
⟨*proof*⟩

lemma *F-orbit-mono*:
$a \preceq b \Longrightarrow F\star\ a \preceq F\star\ b$

$\langle proof \rangle$

end

lemma *F-orbit-comm*:
$F \ \$ \ (F\star \ a) = F\star \ (F \ \$ \ a)$
$\langle proof \rangle$

lemma *iter*:
$\bigcirc \ F\star \ a = F \ \$ \ (F\star \ (\bigcirc a))$
$\langle proof \rangle$

context *mono-F*
begin

lemma *mono-pow-n*:
$x \leq F \ x \Longrightarrow x \leq (F \ \hat{}\hat{} \ n) \ x$
$\langle proof \rangle$

lemma *mono-pow-n'*:
$F \ x \leq x \Longrightarrow (F \ \hat{}\hat{} \ n) \ x \leq x$
$\langle proof \rangle$

lemma *F-ind*:
$F \ \$ \ a \preceq a \Longrightarrow F\star \ a \preceq a$
$\langle proof \rangle$

lemma *F-coind*:
$a \preceq F \ \$ \ a \Longrightarrow a \preceq F\star \ a$
$\langle proof \rangle$

5 Additional Axioms for Function Applications and Orbits, Fig 3

lemma *semi-cont*:
$\Diamond \ (F \ \$ \ a) \preceq F \ \$ \ \Diamond a$
$\langle proof \rangle$

lemma *semi-cocont*:
$F \ \$ \ \Box a \preceq \Box \ (F \ \$ \ a)$
$\langle proof \rangle$

lemma *asc-iter*:
$a \preceq \bigcirc a \Longrightarrow F \ \$ \ F\star \ a \preceq \bigcirc \ F\star \ a$
$\langle proof \rangle$

lemma *desc-iter*:
$\bigcirc a \preceq a \Longrightarrow \bigcirc \ F\star \ a \preceq F \ \$ \ F\star \ a$

$\langle proof \rangle$

lemma *orbit-asc*:
$[\![\ a \preceq F\ \$\ a;\ a \preceq \bigcirc a\ ]\!] \Longrightarrow F\star\ a \preceq \bigcirc\ F\star\ a$
$\langle proof \rangle$

lemma *orbit-desc*:
$[\![\ \bigcirc a \preceq\ a;\ F\ \$\ a \preceq\ a\ ]\!] \Longrightarrow \bigcirc\ F\star\ a \preceq F\star\ a$
$\langle proof \rangle$

lemma *F-orbit-introL*:
$[\![\ a \preceq\ b;\ F\ \$\ b \preceq\ b\ ]\!] \Longrightarrow F\star\ a \preceq\ b$
$\langle proof \rangle$

lemma *F-orbit-introR*:
$[\![\ a \preceq F\ \$\ a;\ a \preceq\ b\ ]\!] \Longrightarrow a \preceq F\star\ b$
$\langle proof \rangle$

end

6 Paper Section 4

Contexts for different continuity assumptions

locale *omega-cont* = *mono-F* +
assumes *omega-cont*: $\bigwedge a.\ a \preceq \bigcirc a \Longrightarrow F\ \$\ \Diamond a \preceq \Diamond\ (F\ \$\ a)$

locale *omega-cocont* = *mono-F* +
assumes *omega-cocont*: $\bigwedge a.\ \bigcirc a \preceq\ a \Longrightarrow \Box\ (F\ \$\ a) \preceq F\ \$\ \Box a$

locale *c-cont* = *mono-F* +
assumes *c-cont*: $\bigwedge a.\ F\ \$\ \Diamond a \preceq \Diamond\ (F\ \$\ a)$
begin

sublocale *omega-cont*
$\langle proof \rangle$

end

locale *c-cocont* = *mono-F* +
assumes *c-cocont*: $\bigwedge a.\ \Box\ (F\ \$\ a) \preceq F\ \$\ \Box a$
begin

sublocale *omega-cocont*
$\langle proof \rangle$

end

7 Tarski Kantorovich

lemma *lim-inf-leq-lim-sup*:
$\Diamond\Box a \preceq \Box\Diamond a$
⟨*proof*⟩

lemma *monotonic-converges*:
$a \preceq \bigcirc\ a \Longrightarrow \Box\Diamond\ a \preceq \Diamond\Box\ a$
⟨*proof*⟩

lemma *monotonic-converges-manual*:
$a \preceq \bigcirc\ a \Longrightarrow \Box\Diamond\ a \preceq \Diamond\Box\ a$
⟨*proof*⟩

context *mono-F*
begin

lemma *pre-tarski-kantorovich-auto*:
$a \preceq \bigcirc\ a \Longrightarrow \Diamond\ (F\ \$\ F\star\ a) \preceq \Diamond F\star\ a$
⟨*proof*⟩

lemma *pre-tarski-kantorovich*:
$a \preceq \bigcirc\ a \Longrightarrow \Diamond\ (F\ \$\ F\star\ a) \preceq \Diamond F\star\ a$
⟨*proof*⟩

end

context *c-cont*
begin

lemma *tkp-pre-fp*:
$a \preceq \bigcirc a \Longrightarrow F\ \$\ \Diamond\ F \star a \preceq \Diamond\ F \star a$
⟨*proof*⟩

lemma *d-quasi-post-fp*:
$a \preceq F\ \$\ a \Longrightarrow \Diamond\ F \star a \preceq \Diamond\ (F\ \$\ F \star a)$
⟨*proof*⟩

lemma *tkp-post-fp*:
$a \preceq F\ \$\ a \Longrightarrow \Diamond\ F \star a \preceq F\ \$\ \Diamond\ F \star a$
⟨*proof*⟩

lemma *tarski-kantorovich-fp*:
$[\![\ a \preceq \bigcirc a;\ a \preceq F\ \$\ a\ ]\!] \Longrightarrow F\ \$\ \Diamond\ F \star a = \Diamond\ F \star a$

⟨*proof*⟩

lemma *tkp-above*:
$a \preceq F \ \$ \ a \Longrightarrow a \preceq \diamond \ F \star a$
⟨*proof*⟩

lemma *tkp-least*:
$[\![\ a \preceq b; \ F \ \$ \ b \preceq b; \ \bigcirc b \preceq b \ ]\!] \Longrightarrow \diamond \ F \star a \preceq b$
⟨*proof*⟩

end

8 Olszewski

lemma (**in** *c-cont*) *bd-quasi-pre-fp*:
$a \preceq \bigcirc a \Longrightarrow \square \ \diamond \ (F \ \$ \ F \star a) \preceq \square \ \diamond \ F \star a$
⟨*proof*⟩

context *mono-F*
begin

lemma *bd-quasi-post-fp*:
$\bigcirc a \preceq a \Longrightarrow \square\diamond F\star \ a \preceq \square\diamond(F \ \$ \ F\star \ a)$
⟨*proof*⟩

lemma *bd-quasi-post-fp-manual*:
$\bigcirc a \preceq a \Longrightarrow \square\diamond F\star \ a \preceq \square\diamond(F \ \$ \ F\star \ a)$
⟨*proof*⟩

end

lemma (**in** *omega-cocont*) *olszewski-post-fp*:
$\bigcirc a \preceq a \Longrightarrow \square\diamond F\star \ a \preceq F \ \$ \ \square\diamond \ F\star \ a$
⟨*proof*⟩

lemma (**in** *c-cont*) *olszewski-pre-fp*:
$a \preceq \bigcirc a \Longrightarrow F \ \$ \ \square\diamond \ F\star \ a \preceq \square\diamond F\star \ a$
⟨*proof*⟩

locale *stronger-olszewski* = *c-cont* + *omega-cocont*
begin

lemma *stronger-olszewski-fp*:
$[\![\ \bigcirc a \preceq a; \ a \preceq \bigcirc a \ ]\!] \Longrightarrow F \ \$ \ \square\diamond \ F\star \ a = \square\diamond F\star \ a$
⟨*proof*⟩

end

9 Latticed k-Induction

definition G :: $'a\ seq \Rightarrow 'a\ seq \Rightarrow 'a$::$complete\text{-}lattice\ seq$ **where**
$G\ b \equiv \lambda c.\ b \sqcap c$

lemma *G-0*:
$(G\ b\ \widehat{\ }\widehat{\ }\ 0)\ c = c$
⟨*proof*⟩

lemma *G-Suc*:
$(G\ b\ \widehat{\ }\widehat{\ }\ Suc\ k)\ b = (G\ b\ \widehat{\ }\widehat{\ }\ k)\ b \sqcap b$
⟨*proof*⟩

lemma *G-deflate*:
$(G\ b\ \widehat{\ }\widehat{\ }\ k)\ b \preceq b$
⟨*proof*⟩

context *mono-F*
begin

lemma *G-Suc-deflate*:
$(G\ b\ \widehat{\ }\widehat{\ }\ Suc\ k)\ b \preceq (G\ b\ \widehat{\ }\widehat{\ }\ k)\ b$
⟨*proof*⟩

lemma *k-ind-park*:
$F\ \$\ (G\ b\ \widehat{\ }\widehat{\ }\ k)\ b \preceq b \Longrightarrow F\ \$\ (G\ b\ \widehat{\ }\widehat{\ }\ k)\ b \preceq (G\ b\ \widehat{\ }\widehat{\ }\ k)\ b$
⟨*proof*⟩

lemma *G-asc-pres*:
$\bigcirc b \preceq b \Longrightarrow \bigcirc\ (G\ b\ \widehat{\ }\widehat{\ }\ k)\ b \preceq (G\ b\ \widehat{\ }\widehat{\ }\ k)\ b$
⟨*proof*⟩

lemma *k-ind*:
$[\![\ F\ \$\ (G\ b\ \widehat{\ }\widehat{\ }\ k)\ b \preceq b;\ \bigcirc b \preceq b\ ]\!] \Longrightarrow \Diamond\ F\star\ \bot \preceq b$
⟨*proof*⟩

end

end