On Extension of Parameterized Coinduction
パラメーター付き余帰納法の拡張について

by

Masaki Hara
原将己

A Senior Thesis
卒業論文

Submitted to
the Department of Information Science
the Faculty of Science, the University of Tokyo
on February 02, 2016
in Partial Fulfillment of the Requirements
for the Degree of Bachelor of Science

Thesis Supervisor: Ichiro Hasuo　蓮尾一郎
Associate Professor of Information Science

ABSTRACT

Most properties on software correctness are described by means of (combinations of) greatest and least fixed points. Therefore it is of interest to provide techniques for proving such specifications in a proof assistant.

In 2013, Chung-Kil Hur *et al.* proposed one of such techniques: parameterized coinduction. It assists one to interactively construct a loop invariant, which one can use to prove properties described by greatest fixed points.

We analyze the technique from two points of view. Firstly, we define a dual counterpart: parameterized induction to prove properties described by least fixed points. Although it has a similar form to parameterized coinduction, one has to provide a ranking function to guarantee that the program in question halts eventually. Secondly, we give a mathematical characterization of how powerful the technique is.

論文要旨

ソフトウェアの正しさは、最大不動点と最小不動点 (の組み合わせ) で記述される。したがって、証明支援系でこれらの仕様を証明するための手法を与えることには意味がある。

2013年に Chung-Kil Hur *et al.* は、そういった手法のひとつである、パラメーター付き余帰納法と呼ばれる手法を提案した。この手法を用いると、最大不動点で表される性質を証明するために必要なループ不変条件を、対話的に構成することができる。

この論文では、パラメーター付き余帰納法を二つの観点から分析する。はじめに、この手法の双対にあたる、最小不動点で表される性質の証明のためのパラメーター付き帰納法を定義する。これはパラメーター付き余帰納法とよく似た形だが、プログラムがいずれ停止することを保証するために、 ranking function を与える必要があるという違いがある。続いて、この手法がどれほど強力であるかを数学的に特徴づける。

Acknowledgements

I appreciate Ichiro Hasuo's helpful advice.

Contents

Chapter 1

Introduction

1.1 Induction and Coinduction

Induction and coinduction are used to define models and properties of computer systems, and therefore they are indispensable in computer science. Examples of inductive and coinductive models include computer programs and their traces of execution. Properties of these systems are also described by means of induction and coinduction: that a program halts eventually, that two parallel tasks are processed fairly, and that two programs behave in the same way.

1.2 Parameterized Coinduction

Consequently, techniques for automated theorem provers and proof assistants aimed at induction and coinduction have been developed. Among these techniques, *parameterized coinduction* is proposed recently, especially for proof assistants.

In proof assistants like Coq, we do not need to write whole proof trees; instead, we specify how to construct proof trees. For example, a proof tree for $\neg\neg(P \lor \neg P)$ is shown in Figure 1.1 and a proof in Coq for the same proposition is shown in Listing 1.1. There is a big difference: in the Coq proof script, there are no appearances of intermediate propositions; there are only tactic names and hypotheses names. This is especially convenient if the proposition in question is long.

However, the standard proof technique for coinductive properties, which uses the Knaster-Tarski theorem, does not work well in these proof assistants, in terms of giving short proofs. To use the Knaster-Tarski theorem, we have to provide beforehand a proposition called "invariant." To make matters worse, invariants are often long and complex.

$$\dfrac{\dfrac{\dfrac{}{\neg(P \lor \neg P) \vdash \neg(P \lor \neg P)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\neg(P \lor \neg P), P \vdash \neg(P \lor \neg P)} \qquad \dfrac{\dfrac{}{\neg(P \lor \neg P), P \vdash P}}{\neg(P \lor \neg P), P \vdash P \lor \neg P}}{\neg(P \lor \neg P), P \vdash \bot}}{\neg(P \lor \neg P) \vdash \neg P}}{\neg(P \lor \neg P) \vdash P \lor \neg P}}{\neg(P \lor \neg P) \vdash \bot}}{\vdash \neg\neg(P \lor \neg P)}$$

Figure 1.1: A proof tree for $\neg\neg(P \lor \neg P)$

Listing 1.1: A Coq proof for $\neg\neg(P \lor \neg P)$

```
Theorem DNLEM(P : Prop) : ¬¬(P ∨  ¬P).
Proof.
  intros H.
  apply H.
  right.
  intros HP.
  apply H.
  left.
  exact HP.
Qed.
```

Parameterized coinduction is a technique for overcoming the problem. In parameterized coinduction, we are to prove propositions called *parameterized greatest fixed points*, instead of usual greatest fixed points. This enables us to gradually increase during an interactive proof a coinductive hypothesis, which will eventually be an invariant.

1.3 Our Contribution

We have two contributions regarding parameterized coinduction.

- We introduce *parameterized induction*, which is similar to parameterized coinduction, to prove inductively defined properties.
- Although parameterized greatest fixed points are more convenient than usual greatest fixed points, there have been no mathematical backgrounds to support this intuition. We give a mathematical characterization of the power of parameterized induction and parameterized coinduction.

1.4 Related Work

For model-checking purposes, assertions to be checked are often expressed in the modal μ-calculus, and processes and programs are checked against these properties. For example, Pratt [14] and Kozen [12] each propose a model-checking algorithm for a fragment of the modal μ-calculus. Wilke [19] proposes algorithms for the model-checking and the satisfiability problem for the full modal μ-calculus, by translating propositions into alternating tree automata.

Non-classical variants of the modal μ-calculus are also considered. Baelde and Miller [2] propose a first-order multiplicative-additive fragment of linear logic augmented with equalities and fixed point operators $\mu\mathrm{MALL}^{=}$ and prove cut-elimination theorem for the logic. Hasuo *et al.* [8] propose a categorical framework, which is called a coalgebraic modal μ-calculus $\mathbf{C}_{\boldsymbol{\mu}_{\Gamma,\Lambda}}$, and give a model-checking algorithm for some of them.

The idea of adding parameters to fixed point operators is originally used for model-checking purposes by Winskel [20]. The idea of accumulated knowledge is older than parameterization and can be dated back to Kozen [12].

Stirling and Walker [17] propose local model checking, a top-down approach to model checking. Interactive proof in proof assistants is similar to local model checking in that we write a proof in a top-down manner.

Chapter 2

Preliminaries

2.1 Backgrounds on Induction and Coinduction

Although many applications of induction and coinduction are about complete boolean algebras, inductive and coinductive properties can be described in a preordered set in general.

Definition 2.1.1 (Preordered sets)**.** A *preordered set* $(C, \sqsubseteq)$ is a set C with a reflexive and transitive binary relation $(\sqsubseteq) \subseteq C \times C$.

We write $x \equiv y$ if both $x \sqsubseteq y$ and $y \sqsubseteq x$ holds.

Minimum elements are unique up to equivalence, if they exist. We write $\bot$ for one of them. We write $\top$, $x \sqcup y$, and $x \sqcap y$ for a maximum element, least upper bounds, and greatest lower bounds, respectively.

In this situation, coinductive and inductive properties are defined as greatest and least fixed points. See e.g. [1] for details.

Definition 2.1.2 (Greatest fixed points and least fixed points)**.** Let $(C, \sqsubseteq)$ be a preordered set and $f \colon C \to C$ be a function.

An element $x \in C$ is a *fixed point* of f if $f(x) \equiv x$ holds.

An element $x \in C$ is a *least fixed point* of f if x is a minimum element among fixed points of f. Similarly, an element $x \in C$ is a *greatest fixed point* of f if x is a maximum element among all the fixed points of f.

Least fixed points and greatest fixed points are unique (if they exist) up to equivalence. We write μf for a least fixed point of f, and νf for a greatest fixed point of f. We write $\mu x.\, t$ and $\nu x.\, t$ as shorthands for $\mu(\lambda x.\, t)$ and $\nu(\lambda x.\, t)$, respectively.

Especially, as proven later, $(C, \sqsubseteq)$ has all greatest and fixed points of monotone functions, if it is a complete lattice.

Definition 2.1.3 (Complete lattices)**.** A preordered set $(C, \sqsubseteq)$ is a *complete lattice*, if it has one of the following equivalent properties:

1. It has all joins: for each subset $A \subseteq C$, there is a minimum element among the upper bounds of A.

2. It has all meets: for each subset $A \subseteq C$, there is a maximum element among the upper bounds of A.

Usually only a partially ordered set with such a property is called a complete lattice. This is a natural generalization of the definition.

Before presenting the Knaster-Tarski theorem, we first define prefixed points and postfixed points.

Definition 2.1.4 (Prefixed points and postfixed points)**.** Let $(C, \sqsubseteq)$ be a preordered set and $f\colon C \to C$ be a function.

An element $x \in C$ is a *prefixed point* of f if $f(x) \sqsubseteq x$ holds.

Dually, an element $x \in C$ is a *postfixed point* of f if $x \sqsubseteq f(x)$ holds.

We show a less general form of the Knaster-Tarski theorem. See [18] for the general statement.

Theorem 2.1.5 (Knaster-Tarski)**.** *Let $(C, \sqsubseteq)$ be a complete lattice. For $f\colon C \to C$ a monotone function, there exists νf and the following equivalence holds:*

$$\nu f \equiv \bigsqcup \{\, r \in C \mid r \sqsubseteq f(r) \,\}\,.$$

Dually, for $f\colon C \to C$ a monotone function, there exists μf and the following equivalence holds:

$$\mu f \equiv \bigsqcap \{\, r \in C \mid f(r) \sqsubseteq r \,\}\,.$$

In other words, greatest fixed points are greatest postfixed points and least fixed points are least prefixed points.

Proof. Let x be $\bigsqcup \{\, r \in C \mid r \sqsubseteq f(r) \,\}$.

Let r be a postfixed point of f. We have $r \sqsubseteq x$. By monotonicity of f, we have $r \sqsubseteq f(r) \sqsubseteq f(x)$. Therefore $x \sqsubseteq f(x)$ by definition of x.

Since $f(x) \sqsubseteq f\big(f(x)\big)$ by monotonicity, $f(x)$ is a postfixed point of f. Therefore $f(x) \sqsubseteq x$ by definition of x.

Since we have $x \sqsubseteq f(x)$ and $f(x) \sqsubseteq x$, x is a fixed point. Let y be another fixed point of f. Since y is a postfixed point, we have $y \sqsubseteq x$ by definition of x. Therefore $x = \bigsqcup \{\, r \in C \mid r \sqsubseteq f(r) \,\}$ is a greatest fixed point. □

The Knaster-Tarski theorem describes μf as an infimum of a certain set. In contrast, Kleene fixed point theorem, a special version of which appears in [11], describes μf as a supremum of a certain sequence. We give a more general theorem, dropping Scott-continuity restriction. See the paper of Cousot and Cousot [4] and the paper of Echenique [6] for detail.

Definition 2.1.6 (Upper iteration sequence)**.** Let $(C, \sqsubseteq)$ be a complete lattice, $f\colon C \to C$ be a monotone function, and $x \in C$ be a postfixed point of f.

For an ordinal number α, an element of an upper iteration sequence $f^\alpha(x)$ is defined inductively as follows:

$$\begin{aligned}
f^0(x) &= x \\
f^{\alpha+1}(x) &= f\big(f^\alpha(x)\big) \\
f^\alpha(x) &= \bigsqcup_{\beta<\alpha} f^\beta(x) && \text{if } \alpha \text{ is a limit ordinal.}
\end{aligned}$$

Dually, for a prefixed point $x \in C$ of f and an ordinal number α, an element of a lower iteration sequence $f^\alpha(x)$ is defined inductively as follows:

$$\begin{aligned}
f^0(x) &= x \\
f^{\alpha+1}(x) &= f\big(f^\alpha(x)\big) \\
f^\alpha(x) &= \bigsqcap_{\beta<\alpha} f^\beta(x) && \text{if } \alpha \text{ is a limit ordinal.}
\end{aligned}$$

There is no confusion, because these two definitions of $f^\alpha(x)$ coincide if x is both a prefixed point and a postfixed point of f.

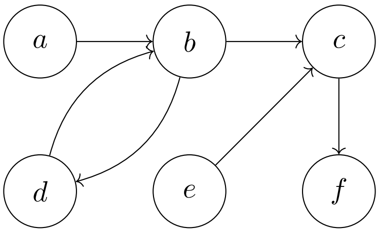

Figure 2.1: A small example of a labelled transition system, where $\Sigma = \{*\}$

Theorem 2.1.7 (Cousot-Cousot fixed point theorem)**.** *Let $(C, \sqsubseteq)$ be a complete lattice, $f\colon C \to C$ be a monotone function, and κ be a cardinal number that is greater than that of any chain in $(C, \sqsubseteq)$.*

Then we have $\mu f \equiv f^{\kappa}(\bot)$.

Dually, $\nu f \equiv f^{\kappa}(\top)$.

□

2.2 Examples of Coinduction and Induction

Now we give examples of coinductive and inductive properties.

Behaviors of programs or processes are usually interpreted in terms of labelled transition systems. See e.g. [15, 16] for details.

Definition 2.2.1 (Labelled transition systems and their modal operators)**.** A *labelled transition system (LTS)* is a set Q equipped with an indexed family of relations $\{(\to_a) \subseteq Q \times Q\}_{a \in \Sigma}$, where Σ is a fixed set called an *alphabet.*

A labelled transition system can be seen as a Kripke frame for the multimodal variant of the modal logic **K**. For each $a \in \Sigma$, two modalities $[a], \langle a \rangle \colon \mathcal{P}(Q) \to \mathcal{P}(Q)$ are defined as follows:

$$[a]Y := \{\, x \in Q \mid \forall y \in X.\, (x \to_a y) \Rightarrow y \in Y \,\},$$
$$\langle a \rangle Y := \{\, x \in Q \mid \exists y \in X,\, (x \to_a y) \land y \in Y \,\}.$$

Both $[a]$ and $\langle a \rangle$ are monotone.

We will use the small example shown in Figure 2.1 to explain how to use parameterized coinduction.

Before we give translation rule from programs or processes to labelled transition systems, we present some properties of labelled transition systems that can be expressed in the modal μ-calculus.

Example 2.2.2 (States that have an infinitely long path)**.** Let $(Q, \to)$ be a labelled transition system over the alphabet $\Sigma = \{*\}$. The set of states that have an infinitely long path starting from them can be expressed as:

$$\text{infPath} := \nu\langle * \rangle.$$

Example 2.2.3 (States that have a path to an accepting state)**.** Let $(Q, \to)$ be a labelled transition system over the alphabet $\Sigma = \{*\}$ and $F \subseteq Q$. The set of states that have a path from them to an accepting state can be expressed as:

$$\text{acc} := \mu Z.\, F \cup \langle * \rangle Z.$$

$$\begin{array}{cccc}
 & \dfrac{}{a \to_a \checkmark} & & \\
\dfrac{t \to_a \checkmark}{t + t' \to_a \checkmark} & \dfrac{t \to_a t''}{t + t' \to_a t''} & \dfrac{t' \to_a \checkmark}{t + t' \to_a \checkmark} & \dfrac{t' \to_a t''}{t + t' \to_a t''} \\
 & \dfrac{t \to_a t''}{t \cdot t' \to_a t'' \cdot t'} & \dfrac{t \to_a \checkmark}{t \cdot t' \to_a t'} &
\end{array}$$

Table 2.1: Interpretation of basic process terms

Example 2.2.4 (States that have an infinitely long path visiting accepting states infinitely often)**.** Let $(Q, \to)$ be a labelled transition system over the alphabet $\Sigma = \{*\}$ and $F \subseteq Q$. The set of states that have an infinitely long path starting from it visiting accepting states infinitely often can be expressed as:

$$\mathrm{infAcc} := \nu Z.\, \mu W.\, (F \cap \langle * \rangle Z) \cup \langle * \rangle W.$$

Now we give an example of a process algebra from [7] and its interpretation as a labelled transition system.

Definition 2.2.5 (Basic process algebra)**.** *Basic process terms* are terms generated from the grammar:

$$t ::= a \mid t + t' \mid t \cdot t'$$

where a is an element of Σ, the fixed alphabet.

Algebra of basic process terms is called *basic process algebra*.

Definition 2.2.6 (Interpretations of basic process terms)**.** Let Q be the disjoint union of the set of closed basic process terms and $\{\checkmark\}$. Interpretations of basic process terms is a labelled transition system $(Q, \{(\to_a)\}_{a \in \Sigma})$, where $(\to_a)$ is inductively defined by the rules in Table 2.1.

Bisimilarity is used to define equivalence between two processes and it is stronger than trace-equivalence. Bisimilarity is also a coinductively defined relation.

Definition 2.2.7 (Bisimulation and bisimilarity)**.** Fix a labelled transition system $(Q, \{(\to_a)\}_{a \in \Sigma})$. Define one-step simulation equation $\mathrm{sim} \colon \mathcal{P}(Q \times Q) \to \mathcal{P}(Q \times Q)$ by

$$\begin{aligned}\mathrm{sim}(R) := \{(q, q') \mid {} & (\forall a.\, \forall r.\, q \to_a r \Rightarrow \exists r'.\, q' \to_a r' \wedge (r, r') \in R) \wedge \\ & (\forall a.\, \forall r'.\, q' \to_a r' \Rightarrow \exists r.\, q \to_a r \wedge (r, r') \in R)\}.\end{aligned}$$

A *bisimulation* is a postfixed point of sim. Two states $q, q' \in Q$ are *bisimilar* if $(q, q') \in \nu\mathrm{sim}$.

2.3 Parameterized Coinduction: A Review

One way to prove a coinductive property is to use the Knaster-Tarski Theorem, i.e.

$$r \sqsubseteq f(r) \Rightarrow r \sqsubseteq \nu f.$$

For example, to prove $a \in \mathrm{infPath}$ in Figure 2.1, we give a proof like this:

$$\begin{aligned}
&a \in \mathrm{infPath} \\
&\Leftarrow \{a,b,d\} \subseteq \nu\langle * \rangle && (\textit{Discover}\text{ an invariant}) \\
&\Leftarrow \{a,b,d\} \subseteq \langle * \rangle(\{a,b,d\}) && (\text{Use the Knaster-Tarski}) \\
&\Leftrightarrow \{a,b,d\} \subseteq \{a,b,d\} && (\text{Expand the modality})
\end{aligned}$$

There is a problem, however: we have to give an invariant at an early stage of the proof. If we apply the Knaster-Tarski to the first goal, we will be confronted by the unprovable goal $\{a\} \subseteq \langle * \rangle \{a\} = \emptyset$.

To overcome this problem, Hur *et al.* [9] propose a proof technique, called *parameterized coinduction.*

Parameterized greatest fixed points play a central role in parameterized coinduction.

Definition 2.3.1 (Parameterized greatest fixed points)**.** Let $(C, \sqsubseteq)$ be a complete lattice and $f \colon C \to C$ be a monotone function. For $x \in C$, we define the *parameterized greatest fixed point* $G_f(x)$ by

$$G_f(x) := \nu y.\, f(x \sqcup y).$$

These elements $G_f(x)$ are monotone in f and x.

As the name suggests, the notion of parameterized greatest fixed point is a generalization of the ordinary notion of greatest fixed point.

Lemma 2.3.2 (Initialization of greatest fixed points)**.** *Let $(C, \sqsubseteq)$ be a complete lattice and $f \colon C \to C$ be a monotone function. Then we have $\nu f \equiv G_f(\bot)$.*

Proof. This is because we have $f \equiv \lambda y.\, f(\bot \sqcup y)$. □

When working with $G_f(x)$ we use two important lemmas: the *unfolding* lemma and the *accumulation* lemma.

Lemma 2.3.3 (The unfolding property of parameterized greatest fixed points)**.** *Let $(C, \sqsubseteq)$ be a complete lattice and $f \colon C \to C$ be a monotone function. Then for all $x \in C$, we have $G_f(x) \equiv f\big(x \sqcup G_f(x)\big)$.*

Proof. This is exactly the fact that $G_f(x)$ is a fixed point of $\lambda y.\, f(x \sqcup y)$. □

Lemma 2.3.4 (The accumulation property of parameterized greatest fixed points)**.** *Let $(C, \sqsubseteq)$ be a complete lattice and $f \colon C \to C$ be a monotone function. Then for all $x, y \in C$, we have $y \sqsubseteq G_f(x)$ if and only if $y \sqsubseteq G_f(x \sqcup y)$.*

Proof. The only-if direction is obvious. To prove the converse, we assume $y \sqsubseteq G_f(x \sqcup y)$. Then $G_f(x \sqcup y)$ is a postfixed point of $\lambda z.\, f(x \sqcup z)$, because

$$\begin{aligned}
G_f(x \sqcup y) &\equiv f\big(x \sqcup y \sqcup G_f(x \sqcup y)\big) \\
&\sqsubseteq f\big(x \sqcup G_f(x \sqcup y)\big).
\end{aligned}$$

Therefore, we have $y \sqsubseteq G_f(x \sqcup y) \sqsubseteq G_f(x)$. □

Example 2.3.5 (Example Proof with Generalized Fixed Points)**.** As a comparison to the Knaster-Tarski proof, we show below a proof of $a \in \mathrm{infPath}$ in Figure 2.1 using $G_f(x)$.

$$
\begin{array}{ll}
a \in \mathrm{infPath} & \\
\Leftrightarrow \{a\} \subseteq G_{\langle * \rangle}(\emptyset) & \text{(Initialization)} \\
\Leftrightarrow \{a\} \subseteq \langle * \rangle\big(\emptyset \cup G_{\langle * \rangle}(\emptyset)\big) & \text{(Unfolding)} \\
\Leftarrow \{b\} \subseteq G_{\langle * \rangle}(\emptyset) & \text{(Move to a next state)} \\
\Leftrightarrow \{b\} \subseteq G_{\langle * \rangle}(\{b\}) & \text{(Accumulation)} \\
\Leftrightarrow \{b\} \subseteq \langle * \rangle\big(\{b\} \cup G_{\langle * \rangle}(\{b\})\big) & \text{(Unfolding)} \\
\Leftarrow \{d\} \subseteq \{b\} \cup G_{\langle * \rangle}(\{b\}) & \text{(Move to a next state)} \\
\Leftarrow \{d\} \subseteq G_{\langle * \rangle}(\{b\}) & \\
\Leftrightarrow \{d\} \subseteq \langle * \rangle\big(\{b\} \cup G_{\langle * \rangle}(\{b\})\big) & \text{(Unfolding)} \\
\Leftarrow \{b\} \subseteq \{b\} \cup G_{\langle * \rangle}(\{b\}) & \text{(Move to a next state)} \\
\Leftarrow \{b\} \subseteq \{b\} &
\end{array}
$$

Although it is longer than the last proof, it has an advantage that it does not need discovery of an invariant. Provers only have to specify Unfolding, Accumulation, or a next state to transition to.

We get another benefit from parameterized coinduction: *compositionality.*

Lemma 2.3.6 (Compositionality)**.** *Let $(C, \sqsubseteq)$ be a complete lattice and $f : C \to C$ be a monotone function. For $r, g_1, g_2 \in C$, if $g_1 \sqsubseteq G_f(r \sqcup g_2)$ and $g_2 \sqsubseteq G_f(r \sqcup g_1)$ hold, then $g_1 \sqcup g_2 \sqsubseteq G_f(r)$.*

Proof. From the assumptions we get $g_1 \sqcup g_2 \sqsubseteq G_f(r \sqcup g_1 \sqcup g_2)$. By Lemma 2.3.4, $g_1 \sqcup g_2 \sqsubseteq G_f(r)$. □

Chapter 3

Parameterized Induction

3.1 Motivating Example: Reachability Problem in Pushdown Systems

Problems of proving coinductive properties $x \sqsubseteq \nu f$ and inductive properties $x \sqsubseteq \mu f$ are similar in that for both kinds of problems, we have to give an invariant, although we also have to give a ranking function for inductive properties. Therefore, it may be convenient to introduce an inductive counterpart of parameterized coinduction.

For example, let us consider a (usually finite-state) system with parameters in each state: pushdown systems [3].

Definition 3.1.1 (Labelled pushdown systems)**.** A *labelled pushdown system* is a tuple $(\Sigma, P, \Gamma, \Delta)$, where Σ is an *input alphabet*, P is a set of *control locations*, Γ is a *stack alphabet*, and $\Delta \subseteq \bigl(P \times (\Gamma \cup \{\epsilon\})\bigr) \times \Sigma \times (P \times \Gamma^*)$ is a set of *transition rules*. Usually finiteness is also imposed on Σ, P, Γ, and Δ.

For $p \in P$ and $w \in \Gamma^*$, a pair (p, w) is called a *configuration* of the pushdown system. The set of configurations can be regarded as a labelled transition system, where $(q, \gamma w') \to_a (q', ww')$ if $\bigl((q, \gamma), a, (q', w)\bigr) \in \Delta$ and $(q, \epsilon) \to_a (q', w)$ if $\bigl((q, \epsilon), a, (q', w)\bigr) \in \Delta$.

Example 3.1.2 (Reachability problems for pushdown systems)**.** Let $(\{*\}, P, \Gamma, \Delta)$ be a pushdown system, $c \in P \times \Gamma^*$ be an *initial configuration*, and $F \subseteq P \times \Gamma^*$ be a set of *accepting configurations*.

The *reachability problem* is a problem to decide whether there is a path from c to an element of F or not.

Since reachability can be expressed as $\{c\} \subseteq \mu Z.\, F \cup \langle * \rangle Z$, it is an inductive property.

3.2 Parameterized Least Fixed Points

Here we define parameterized least fixed points.

Definition 3.2.1 (Parameterized least fixed points)**.** Let $(C, \sqsubseteq)$ be a complete lattice and $f \colon C \to C$ be a monotone function. For $x \in C$, we define the *parameterized least fixed point* $L_f(x)$ by

$$L_f(x) := \mu y.\, f(x \sqcup y).$$

These elements $L_f(x)$ are monotone in f and x.

Note that, although it is a "dual" notion of $G_f(x)$, it is not a genuine dual in terms of dual order; the order-theoretical dual of $G_f(x)$ is $\mu y.\, f(x \sqcap y)$.

Some lemmas of $G_f(x)$ are true for $L_f(x)$ with no changes.

Lemma 3.2.2 (Initialization of least fixed points)**.** *Let $(C, \sqsubseteq)$ be a complete lattice and $f\colon C \to C$ be a monotone function. Then we have $\mu f \equiv L_f(\bot)$.*

Proof. This is because we have $f \equiv \lambda y.\, f(\bot \sqcup y)$. □

Lemma 3.2.3 (The unfolding property of parameterized least fixed points)**.** *Let $(C, \sqsubseteq)$ be a complete lattice and $f\colon C \to C$ be a monotone function. Then for all $x \in C$, we have $L_f(x) \equiv f\big(x \sqcup L_f(x)\big)$.*

Proof. This is exactly the fact that $G_f(x)$ is a fixed point of $\lambda y.\, f(x \sqcup y)$. □

3.3 The Indexed Accumulation Property

As stated later, Lemma 2.3.4 and Lemma 2.3.6 are strong enough to characterize *greatest* fixed points. Therefore, *least* fixed points does not have such properties with no change.

However, similar reasoning can be done, if we give a ranking function.

Lemma 3.3.1 (The indexed accumulation property of parameterized least fixed points)**.** *Let $(C, \sqsubseteq)$ be a complete lattice and $f\colon C \to C$ be a monotone function. Let α' be an ordinal number and $x, y\colon \alpha' \to C$ be indexed families of elements of C. Also assume x is increasing. Then we have*

$$\begin{aligned} &\big(\forall \alpha \in \alpha'.\, y(\alpha) \sqsubseteq L_f\big(x(\alpha)\big)\big) \\ \iff &\left(\forall \alpha \in \alpha'.\, y(\alpha) \sqsubseteq L_f\left(x(\alpha) \sqcup \bigsqcup_{\beta<\alpha} y(\beta)\right)\right). \end{aligned}$$

Proof. Again, the only-if direction is obvious. To prove the converse, we assume $y(\alpha) \sqsubseteq L_f\Big(x(\alpha) \sqcup \bigsqcup_{\beta<\alpha} y(\beta)\Big)$ for all $\alpha \in \alpha'$.

We will prove $y(\alpha) \sqsubseteq L_f\big(x(\alpha)\big)$ by induction on α. By the inductive hypothesis and the monotonicity of x, we have

$$\bigsqcup_{\beta<\alpha} y(\beta) \sqsubseteq L_f\big(x(\alpha)\big).$$

We get $L_f\Big(x(\alpha) \sqcup \bigsqcup_{\beta<\alpha} y(\beta)\Big) \sqsubseteq L_f\Big(x(\alpha) \sqcup L_f\big(x(\alpha)\big)\Big)$ from this. Therefore, it suffices to show $L_f\Big(x(\alpha) \sqcup L_f\big(x(\alpha)\big)\Big) \sqsubseteq L_f\big(x(\alpha)\big)$. This is true because $L_f\big(x(\alpha)\big)$ is a prefixed point of $\lambda z.\, f\Big(x(\alpha) \sqcup L_f\big(x(\alpha)\big) \sqcup z\Big)$. □

Lemma 3.3.2 (The indexed composition property of parameterized least fixed points)**.** *Let $(C, \sqsubseteq)$ be a complete lattice and $f\colon C \to C$ be a monotone function. Let α' be an ordinal number and $r, g_1, g_2\colon \alpha' \to C$ be indexed families of elements of C. Assume r is monotone.*

Assume $g_1(\alpha) \sqsubseteq L_f\big(r(\alpha) \sqcup \bigsqcup_{\beta<\alpha} g_2(\beta)\big)$ and $g_2(\alpha) \sqsubseteq L_f\big(r(\alpha) \sqcup g_1(\alpha)\big)$ for all $\alpha \in \alpha'$. Then we have $g_1(\alpha) \sqcup g_2(\alpha) \sqsubseteq L_f\big(r(\alpha)\big)$ for all $\alpha \in \alpha'$.

Proof. Define $r', g\colon 2\alpha' \to C$ as

$$\begin{aligned} g(2\alpha) &:= g_1(\alpha) \\ g(2\alpha+1) &:= g_2(\alpha) \\ r'(2\alpha) &:= r(\alpha) \\ r'(2\alpha+1) &:= r(\alpha). \end{aligned}$$

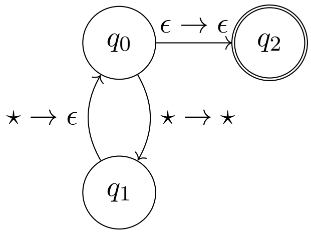

Figure 3.1: A small example of a labelled pushdown system

Then r' is monotone. From the assumptions, $g(\gamma) \sqsubseteq L_f\big(r'(\gamma) \sqcup \bigsqcup_{\delta<\gamma} g(\delta)\big)$ for all $\gamma \in 2\alpha$. By Lemma 3.3.1, $g(\gamma) \sqsubseteq L_f\big(r'(\gamma)\big)$ for all $\gamma \in 2\alpha$. Assigning $\gamma = 2\alpha$ and $\gamma = 2\alpha + 1$ leads to $g_1(\alpha) \sqsubseteq L_f\big(r(\alpha)\big)$ and $g_2(\alpha) \sqsubseteq L_f\big(r(\alpha)\big)$. □

3.4 Example Proof with Parameterized Least Fixed Points

Example 3.4.1 (An example of a pushdown system)**.** Let us consider a pushdown system $(\{*\}, \{q_0, q_1, q_2\}, \{\star\}, \Delta)$, where $\Delta = \{\big((q_0, \star), *, (q_1, \star)\big), \big((q_1, \star), *, (q_0, \epsilon)\big), \big((q_0, \epsilon), *, (q_2, \epsilon)\big)\}$ (Figure 3.1).

Let the set of accepting states $F = \{\, (q_2, w) \mid w \in \{\star\}^* \,\}$. Then for all $w \in \{\star\}^*$, from (q_0, w) one of accepting states can be reachable.

Proof.

$$\begin{aligned}
&\forall n \geq 0.\, \{(q_0, \star^n)\} \subseteq \mu Z.F \cup \langle * \rangle Z \\
&\Leftrightarrow \forall n \geq 0.\, \{(q_0, \star^n)\} \subseteq L_{\lambda Z.F \cup \langle * \rangle Z}(\emptyset) && \text{(Initialization)} \\
&\Leftrightarrow \forall n \geq 0.\, \{(q_0, \star^n)\} \subseteq L_{\lambda Z.F \cup \langle * \rangle Z}(\{\, (q_0, \star^m) \mid m < n \,\}) && \text{(Indexed Accumulation)} \\
&\Leftrightarrow \forall n \geq 0.\, \{(q_0, \star^n)\} \subseteq F \cup \langle * \rangle\big(\{\, (q_0, \star^m) \mid m < n \,\} \cup \\
&\qquad\qquad L_{\lambda Z.F \cup \langle * \rangle Z}(\{\, (q_0, \star^m) \mid m < n \,\})\big) && \text{(Unfolding)} \\
&\Leftarrow \forall n \geq 0.\, \{(q_0, \star^n)\} \subseteq \langle * \rangle\big(\{\, (q_0, \star^m) \mid m < n \,\} \cup \\
&\qquad\qquad L_{\lambda Z.F \cup \langle * \rangle Z}(\{\, (q_0, \star^m) \mid m < n \,\})\big)
\end{aligned}$$

Here we do a case analysis:

- For $n = 0$,

$$\begin{aligned}
&\{(q_0, \epsilon)\} \subseteq \langle * \rangle\big(L_{\lambda Z.F \cup \langle * \rangle Z}(\emptyset)\big) \\
&\Leftarrow \{(q_2, \epsilon)\} \subseteq L_{\lambda Z.F \cup \langle * \rangle Z}(\emptyset) && \text{(Step)} \\
&\Leftrightarrow \{(q_2, \epsilon)\} \subseteq F \cup \langle * \rangle\big(L_{\lambda Z.F \cup \langle * \rangle Z}(\emptyset)\big) && \text{(Unfolding)} \\
&\Leftarrow \{(q_2, \epsilon)\} \subseteq F
\end{aligned}$$

- For $n > 0$,

$$\begin{aligned}
&\forall n > 0.\, \{(q_0, \star^n)\} \subseteq \langle * \rangle \big(\{\, (q_0, \star^m) \mid m < n \,\} \cup \\
&\qquad\qquad L_{\lambda Z.F \cup \langle * \rangle Z}(\{\, (q_0, \star^m) \mid m < n \,\}) \big) \\
&\Leftrightarrow \forall n \geq 0.\, \{(q_0, \star^{n+1})\} \subseteq \langle * \rangle \big(\{\, (q_0, \star^m) \mid m < n+1 \,\} \cup \\
&\qquad\qquad L_{\lambda Z.F \cup \langle * \rangle Z}(\{\, (q_0, \star^m) \mid m < n+1 \,\}) \big) \\
&\Leftarrow \forall n \geq 0.\, \{(q_1, \star^{n+1})\} \subseteq \{\, (q_0, \star^m) \mid m < n+1 \,\} \cup \\
&\qquad\qquad L_{\lambda Z.F \cup \langle * \rangle Z}(\{\, (q_0, \star^m) \mid m < n+1 \,\}) && \text{(Step)} \\
&\Leftarrow \forall n \geq 0.\, \{(q_1, \star^{n+1})\} \subseteq \\
&\qquad\qquad L_{\lambda Z.F \cup \langle * \rangle Z}(\{\, (q_0, \star^m) \mid m < n+1 \,\})
\end{aligned}$$

Then,

$$\begin{aligned}
&\forall n \geq 0.\, \{(q_1, \star^{n+1})\} \subseteq \\
&\qquad\qquad L_{\lambda Z.F \cup \langle * \rangle Z}(\{\, (q_0, \star^m) \mid m < n+1 \,\}) \\
&\Leftrightarrow \forall n \geq 0.\, \{(q_1, \star^{n+1})\} \subseteq F \cup \\
&\qquad\qquad \langle * \rangle \big(\{\, (q_0, \star^m) \mid m < n+1 \,\} \cup \\
&\qquad\qquad\quad L_{\lambda Z.F \cup \langle * \rangle Z}(\{\, (q_0, \star^m) \mid m < n+1 \,\}) \big) && \text{(Unfolding)} \\
&\Leftarrow \forall n \geq 0.\, \{(q_1, \star^{n+1})\} \subseteq \langle * \rangle \big(\{\, (q_0, \star^m) \mid m < n+1 \,\} \cup \\
&\qquad\qquad L_{\lambda Z.F \cup \langle * \rangle Z}(\{\, (q_0, \star^m) \mid m < n+1 \,\}) \big) \\
&\Leftarrow \forall n \geq 0.\, \{(q_0, \star^{n})\} \subseteq \{\, (q_0, \star^m) \mid m < n+1 \,\} \cup \\
&\qquad\qquad L_{\lambda Z.F \cup \langle * \rangle Z}(\{\, (q_0, \star^m) \mid m < n+1 \,\}) && \text{(Step)} \\
&\Leftarrow \forall n \geq 0.\, \{(q_0, \star^{n})\} \subseteq \{\, (q_0, \star^m) \mid m < n+1 \,\}
\end{aligned}$$

□

Chapter 4

Functional Characterization of Parameterized (Co)Induction

4.1 Test Functions Generalizing Parameterized (Co)Induction

We are interested in how "convenient" $G_f(x)$ and $L_f(x)$ are. To investigate this, we introduce test functions F that share some properties with $G_f(x)$ and $L_f(x)$.

Definition 4.1.1 (Test functions for the parameterized fixed points)**.** Let $(C, \sqsubseteq)$ be a complete lattice and $f \colon C \to C$ be a monotone function. A *test function* for f is a monotone function $F \colon C \to C$.

A test function F for f has the *unfolding* property, if for all $x \in C$, $f\big(x \sqcup F(x)\big) \sqsubseteq F(x)$ holds.

A test function F for f has the *accumulation* property, if for all $x, y \in C$, $y \sqsubseteq F(x \sqcup y)$ implies $y \sqsubseteq F(x)$.

A test function F for f has the *composition* property, if for all $r, g_1, g_2 \in C$, if $g_1 \sqsubseteq F(r \sqcup g_2)$ and $g_2 \sqsubseteq F(r \sqcup g_1)$, then $g_1 \sqcup g_2 \sqsubseteq F(r)$.

A test function F for f has the *indexed accumulation* property, if for all ordinal numbers α' and for all $x, y \colon \alpha' \to C$ where x is monotone, the following two conditions are equivalent:

- For all $\alpha \in \alpha'$, we have $y(\alpha) \sqsubseteq F\big(x(\alpha)\big)$.
- For all $\alpha \in \alpha'$, we have $y(\alpha) \sqsubseteq F\big(x(\alpha) \sqcup \bigsqcup_{\beta<\alpha} y(\beta)\big)$.

A test function F for f has the *indexed composition* property, if the following condition is satisfied:

- Let α' be an ordinal number and $r, g_1, g_2 \colon \alpha' \to C$ be indexed families of elements of C, where r is monotone. Assume that for all $\alpha \in \alpha'$, $g_1(\alpha) \sqsubseteq F\big(r(\alpha) \sqcup \bigsqcup_{\beta<\alpha} g_2(\beta)\big)$ and $g_2(\alpha) \sqsubseteq F\big(r(\alpha) \sqcup g_1(\alpha)\big)$. Then for all $\alpha \in \alpha'$, we have $g_1(\alpha) \sqcup g_2(\alpha) \sqsubseteq F\big(r(\alpha)\big)$.

4.2 Interderivability of Accumulation and Composition

The theorem below shows that the accumulation property and the composition property are interderivable.

Theorem 4.2.1 (Accumulation is equivalent to composition)**.** *Let $(C, \sqsubseteq)$ be a complete lattice and $f \colon C \to C$ be a monotone function. For all test functions F for f, F has the accumulation property if and only if it has the composition property.*

Proof. To prove the accumulation property from the composition property, just let $r = x$ and $g_1 = g_2 = y$.

To prove the converse, assume $g_1 \sqsubseteq F(r \sqcup g_2)$ and $g_2 \sqsubseteq F(r \sqcup g_1)$. Since $g_1 \sqcup g_2 \sqsubseteq F(r \sqcup g_1 \sqcup g_2)$, we have $g_1 \sqcup g_2 \sqsubseteq F(r)$. □

Therefore we identify the accumulation property and the composition property in this chapter.

In contrast, neither indexed accumulation nor indexed composition is not essential.

Theorem 4.2.2 (Unfolding implies indexed accumulation and indexed composition)**.** *Let $(C, \sqsubseteq)$ be a complete lattice and $f : C \to C$ be a monotone function. For all test functions F for f with the unfolding property, F has the indexed accumulation and indexed composition properties.*

Proof. Essentially the same as Lemma 3.3.1 and Lemma 3.3.2. □

4.3 Characterization of Parameterized Coinduction

In this setting, $G_f(x)$ is characterized by the unfolding property and the accumulation property.

Lemma 4.3.1 (Characterization of parameterized greatest fixed point)**.** *Let $(C, \sqsubseteq)$ be a complete lattice and $f : C \to C$ be a monotone function. Assume that a test function F for f has the unfolding property and the accumulation property. Then we have $G_f \sqsubseteq F$.*

Proof. To prove $G_f(x) \sqsubseteq F(x)$, It suffices to show $G_f(x) \sqsubseteq F\big(x \sqcup G_f(x)\big)$. This is equivalent to $f\big(x \sqcup G_f(x)\big) \sqsubseteq f\Big(x \sqcup G_f(x) \sqcup F\big(x \sqcup G_f(x)\big)\Big)$. □

Note that a similar theorem to the above lemma appears in [9]. Since we employ a looser condition for the unfolding property, the result of our lemma is a bit weaker.

Using the lemma above, we can derive the following equivalence theorem.

Theorem 4.3.2 (Characterization of the unfolding property and the accumulation property)**.** *Let $(C, \sqsubseteq)$ be a complete lattice and $f : C \to C$ be a monotone function. For all $x, y \in C$, the following are equivalent:*

1. *For all test functions F for f with the unfolding property and the accumulation property, we have $y \sqsubseteq F(x)$.*
2. *$y \sqsubseteq G_f(x)$.*

Proof. 1. implies 2. because we can use G_f as F.

2. implies 1. because we have $G_f \sqsubseteq F$. □

4.4 Characterization of Parameterized Induction

There is also the least fixed point variant of the theorem above.

Lemma 4.4.1 (Characterization of parameterized least fixed point)**.** *Let $(C, \sqsubseteq)$ be a complete lattice and $f : C \to C$ be a monotone function. Assume that a test function F for f has the unfolding property. Then we have $L_f \sqsubseteq F$.*

Proof. Since F is a prefixed point of $\lambda w.\, f(x \sqcup w)$ and L_f is the least prefixed point of $\lambda w.\, f(x \sqcup w)$, we have $L_f(x) \sqsubseteq F(x)$. □

Theorem 4.4.2 (Characterization of the unfolding property)**.** *Let $(C, \sqsubseteq)$ be a complete lattice and $f\colon C \to C$ be a monotone function. For all $x, y \in C$, the following are equivalent:*

1. *For all test functions F for f with the unfolding property, we have $y \sqsubseteq F(x)$.*
2. *We have $y \sqsubseteq L_f(x)$.*

Proof. This is because $L_f(x)$ is the least prefixed point of $\lambda z.\, f(x \sqcup z)$. □

Chapter 5

Predicate Characterization and No-discovery Parameterized (Co)Induction

5.1 Motivating Example: Infinite State System

In Chapter 4, we showed that Lemma 2.3.3 and Lemma 2.3.4 are enough to characterize G_f. However, still there is a possibility that one have to discover an invariant.

Our motivating example is the labelled transition system below:

Example 5.1.1. We consider the labelled transition system in Figure 5.1.

Below is an unsuccessful attempt to prove $q_0 \in \mathrm{infPath}$.

$$\begin{aligned}
&\{q_0\} \subseteq \mathrm{infPath} && \\
&\Leftrightarrow \{q_0\} \subseteq G_{\langle * \rangle}(\{q_0\}) && \text{(Accumulation)} \\
&\Leftrightarrow \{q_0\} \subseteq \langle * \rangle\big(\{q_0\} \cup G_{\langle * \rangle}(\{q_0\})\big) && \text{(Unfolding)} \\
&\Leftarrow \{q_1\} \subseteq \{q_0\} \cup G_{\langle * \rangle}(\{q_0\}) && \text{(Step)} \\
&\Leftarrow \{q_1\} \subseteq G_{\langle * \rangle}(\{q_0\}) && \\
&\Leftrightarrow \{q_1\} \subseteq G_{\langle * \rangle}(\{q_0, q_1\}) && \text{(Accumulation)} \\
&\Leftrightarrow \{q_1\} \subseteq \langle * \rangle\big(\{q_0, q_1\} \cup G_{\langle * \rangle}(\{q_0, q_1\})\big) && \text{(Unfolding)} \\
&\Leftarrow \{q_2\} \subseteq \{q_0, q_1\} \cup G_{\langle * \rangle}(\{q_0, q_1\}) && \text{(Step)} \\
&\Leftarrow \{q_2\} \subseteq G_{\langle * \rangle}(\{q_0, q_1\}) && \\
&\vdots &&
\end{aligned}$$

In the attempt, we acquire accumulated knowledge about states that we have gone through. However, we have no opportunity to use it.

By discovering an invariant, though, $q_0 \in \mathrm{infPath}$ can be proven.

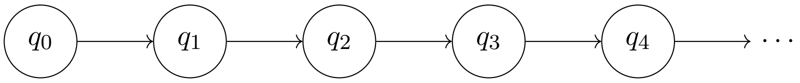

Figure 5.1: A small example of an infinite state labelled transition system

$$\begin{aligned}
&\{q_0\} \subseteq \mathrm{infPath} \\
&\Leftarrow \{\, q_i \mid i \geq 0 \,\} \subseteq \mathrm{infPath} \\
&\Leftarrow \{\, q_i \mid i \geq 0 \,\} \subseteq \langle * \rangle \{\, q_i \mid i \geq 0 \,\} && \text{(Tarski)} \\
&\Leftarrow \{\, q_i \mid i \geq 1 \,\} \subseteq \{\, q_i \mid i \geq 0 \,\} && \text{(Step)}
\end{aligned}$$

Although there is an intuition that we have to discover an invariant in the example above, Theorem 5.5 does not capture that intuition.

5.2 Test Predicates Generalizing Test Functions

To model the problem mathematically, we should define what "can be proven without discovering an invariant." There arise two problems:

- If the goal is of the form $y \sqsubseteq F(x)$, we can always use the strategy of discovering an invariant: it suffices to show $y \sqsubseteq G_f(x)$ and $G_f(x) \sqsubseteq F(x)$. We have to give appropriate generalization of $y \sqsubseteq F(x)$.
- What can be done to the goal of the form $y \sqsubseteq g(x)$ depends on the lattice C and the function $g\colon C \to C$ that we are working on. We have to define it.

We would like to resolve the first problem by generalizing test functions to *test predicates.*

Definition 5.2.1 (Test predicates for parameterized fixed points)**.** Let $(C, \sqsubseteq)$ be a complete lattice and $f\colon C \to C$ be a monotone function. A *test predicate for the parameterized fixed points* of f is a predicate $P \subseteq C \times C$.

A test predicate P has the *downward closure* property, if for all $x, y, y' \in C$, $y \sqsubseteq y'$ and $P(x, y')$ implies $P(x, y)$.

A test predicate P has the *join closure* property, if for all $Y \subseteq C$, $\bigl(\forall y \in Y.\, P(x, y)\bigr)$ implies $P(x, \bigsqcup Y)$.

Test predicates are actually generalization of test functions, as shown in the theorem below:

Theorem 5.2.2 (Test predicates generalize test functions)**.** *Let $(C, \sqsubseteq)$ be a complete lattice and $P \subseteq C \times C$ be a test predicate. The following are equivalent:*

1. *We have the downward closure property and the join closure property for P.*
2. *There exists a test function $F\colon C \to C$ such that for all $x, y \in C$, $y \sqsubseteq F(x) \iff P(x, y)$.*

Furthermore, such $F\colon C \to C$ are unique up to equivalence.

Proof. Obviously 2. implies 1.

To prove 2. from 1., let $F(x) := \bigsqcup \{\, y \mid P(x, y) \,\}$. By join closure, $P\bigl(x, F(x)\bigr)$. By downward closure, $P(x, y)$.

Assume that both F and F' satisfies the property in 2. Then from $F'(x) \sqsubseteq F'(x)$ we have $F(x) \sqsubseteq F'(x)$, and $F'(x) \sqsubseteq F(x)$ alike. □

5.3 Generalization of the Unfolding Property and the Accumulation Property

For the second problem, we want to use a left adjoint: a monotone function h such that $y \sqsubseteq g(x) \iff h(y) \sqsubseteq x$. However, the left adjoint need not exist in general. Instead, we define a looser condition:

Definition 5.3.1 (Minimal witnesses)**.** Let $(C, \sqsubseteq)$ be a complete lattice. For a monotone function $g\colon C \to C$ and $y, z \in C$, z is a *minimal witness* from y to g if and only if it is a minimal element of $\{\, z' \in C \mid y \sqsubseteq g(z') \,\}$.

Note that, if we replace "minimal" with "minimum," we get a universal arrow and a left adjoint.

Minimal witnesses have enough power to prove propositions of the form $y \sqsubseteq g(x)$.

Lemma 5.3.2 (Characterization of adjoint-like reasoning)**.** *Let $(C, \sqsubseteq)$ be a complete lattice and $g\colon C \to C$ be a monotone function preserving directed infima. Then for all $x, y \in C$, the following are equivalent:*

1. *There exists $z \in C$ such that $z \sqsubseteq x$ and z is a minimal witness from y to g.*

2. *$y \sqsubseteq g(x)$.*

Proof. To prove 2. from 1., just use the fact that z is a minimal witness.

To prove 1. from 2., let $A := \{\, z' \in C \mid z' \sqsubseteq x \land y \sqsubseteq g(z') \,\}$ and apply Zorn's lemma to $A \subseteq C$ (in a reverse order).

- A is nonempty because $x \in A$.

- Let $B \subseteq A$ be a nonempty chain of A. Then $\bigsqcap B \in A$ and this is a lower bound of B. □

Using minimal witnesses, we can generalize the unfolding condition.

Definition 5.3.3 (The unfolding property of test predicates)**.** Let $(C, \sqsubseteq)$ be a complete lattice and $f\colon C \to C$ be a monotone function. A test predicate P has the *unfolding* property, if for all minimal witnesses $z \in C$ from y to $\lambda w.\, f(x \sqcup w)$, $P(x, z)$ implies $P(x, y)$.

Theorem 5.3.4 (The unfolding property of test predicates generalizes that of test functions)**.** *Let $(C, \sqsubseteq)$ be a complete lattice and $f\colon C \to C$ be a monotone function. Assume for all $w \in C$, $\lambda w.\, f(x \sqcup w)$ preserves directed infima. Let $P \subseteq C \times C$ be a test predicate of f. Then the following are equivalent:*

1. *We have the downward closure property, the join closure property, and the unfolding property for P.*

2. *There exists a test function $F\colon C \to C$ such that F satisfies the unfolding property and for all $x, y \in C$, $y \sqsubseteq F(x) \iff P(x, y)$.*

Proof. 1. is immediate from 2. and Lemma 5.3.2.

To prove 2. from 1., as in Theorem 5.2.2, let $F(x) := \bigsqcup \{\, y \mid P(x, y) \,\}$. Since $f\bigl(x \sqcup F(x)\bigr) \sqsubseteq f\bigl(x \sqcup F(x)\bigr)$, there exists $z \in C$ such that $z \sqsubseteq F(x)$ and z is a minimal witness from $f\bigl(x \sqcup F(x)\bigr)$ to $\lambda w.\, f(x \sqcup w)$. By the unfolding property of P, we have $P\Bigl(x, f\bigl(x \sqcup F(x)\bigr)\Bigr)$ i.e. $f\bigl(x \sqcup F(x)\bigr) \sqsubseteq F(x)$.

□

The accumulation property can also be generalized to test predicates.

Definition 5.3.5 (The accumulation property of test predicates)**.** Let $(C, \sqsubseteq)$ be a complete lattice and $f\colon C \to C$ be a monotone function. A test predicate P has the *accumulation* property, if for all $x, y \in C$, $P(x \sqcup y, y)$ implies $P(x, y)$.

Theorem 5.3.6 (The accumulation property of test predicates generalizes that of test functions)**.** *Let $(C, \sqsubseteq)$ be a complete lattice and $f\colon C \to C$ be a monotone function. Assume for all $w \in C$, $\lambda w.\, f(x \sqcup w)$ preserves directed infima. Let $P \subseteq C \times C$ be a test predicate of f. Then the following are equivalent:*

1. *We have the downward closure property, the join closure property, and the accumulation property for P.*

2. *There exists a test function $F\colon C \to C$ such that F satisfies the accumulation property and for all $x, y \in C$, $y \sqsubseteq F(x) \iff P(x, y)$.*

Proof. 1. is immediate from 2.

To prove 2. from 1., as in Theorem 5.2.2, just let $F(x) := \bigsqcup \{\, y \mid P(x, y) \,\}$. □

Generalization means dropping the downward closure condition.

5.4 Characterization of No-discovery Parameterized Induction

Before characterizing no-discovery parameterized coinduction, we show that the similar result to Theorem 4.4.2 holds with a little restriction.

Theorem 5.4.1 (Characterization of unfolding, without discovery of an invariant)**.** *Let $(C, \sqsubseteq)$ be a complete lattice and $f\colon C \to C$ be a monotone function. Assume $\lambda w.\, f(x \sqcup w)$ preserves directed infima and for all $z \in C$, $\lambda w.\, z \sqcap w$ preserves directed suprema. For all $x, y \in C$, the following are equivalent:*

1. *For all test predicates P of f, if f satisfies the join closure property and the unfolding property, then $P(x, y)$ holds.*

2. *$y \sqsubseteq L_f(x)$.*

Proof. We define $g(w) := f(x \sqcup w)$.

1. implies 2. because we can use $y \sqsubseteq L_f(x)$ as $P(x, y)$.

To prove 2. from 1., we use Theorem 2.1.7. By the theorem, $y \sqsubseteq g^{\alpha}(\bot)$ for some α. Do induction on α.

If $\alpha = 0$, then $y \equiv \bot$ by the hypothesis. The join closure property can be applied to prove $P(x, \bot)$.

If $\alpha = \alpha' + 1$, then $y \sqsubseteq g\big(g^{\alpha'}(\bot)\big)$. By Lemma 5.3.2, there exists $z \in C$ such that z is a minimal witness from y to g and $z \sqsubseteq g^{\alpha'}(\bot)$. By the inductive hypothesis, $P(x, z)$. By unfolding, $P(x, y)$.

If α is a limit ordinal, then $y \sqsubseteq \bigsqcup_{\beta<\alpha} g^{\beta}(\bot)$. By the inductive hypothesis, for all $\beta < \alpha$, $P\big(x, y \sqcap g^{\beta}(\bot)\big)$. By the join closure property, we have $P\Big(x, \bigsqcup_{\beta<\alpha}\big(y \sqcap g^{\beta}(\bot)\big)\Big)$. This equivalent to $P(x, y)$. □

5.5 Characterization of No-discovery Parameterized Coinduction

Here we prove a similar theorem to for *finite* lattices.

Theorem 5.5.1 (Characterization of unfolding and accumulation, without discovery of an invariant)**.** *Let $(C, \sqsubseteq)$ be a finite lattice and $f \colon C \to C$ be a monotone function. Assume $\lambda w.\, f(x \sqcup w)$ preserves directed infima. For all $x, y \in C$, the following are equivalent:*

1. *For all test predicates P of f, if P satisfies the join closue property, the unfolding property, and the accumulation property, then $P(x, y)$ holds.*
2. *$y \sqsubseteq G_f(x)$.*

Proof. 2. is immediate from 1. We prove 1. from 2.

We construct a pair of sequences $\{x_i\}_{i \geq 0}$, $\{y_i\}_{i \geq 0}$ satisfying following conditions:

- $x_0 = x$, $y_0 = y$.
- $x_{i+1} = x_i \sqcup y_i$
- y_{i+1} is a minimal witness from y_i to $\lambda w.\, f(x_{i+1} \sqcup w)$.

Assume $x', y' \in C$ and $y' \sqsubseteq G_f(x')$. Since $y' \sqsubseteq f\big(x' \sqcup y' \sqcup G_f(x' \sqcup y')\big)$, there exists $y'' \in C$ such that $y'' \sqsubseteq G_f(x' \sqcup y')$ and y'' is a minimal witness from y' to $\lambda w.\, f(x' \sqcup y' \sqcup w)$. Therefore, by dependent choice, there is such a pair of sequences.

For all $i \geq 0$, $P(x_{i+1}, y_{i+1})$ implies $P(x, y)$. We prove this by induction on i. Assume that $P(x_{i+1}, y_{i+1})$. Since y_{i+1} is a minimal witness, we have $P(x_i \sqcup y_i, y_i)$. Therefore we have $P(x_i, y_i)$ by the accumulation property.

Since $\{x_i\}_{i \geq 0}$ is monotone and C is finite, there exists $i > 0$ such that $x_i \equiv x_{i+1}$. This means $y_i \sqsubseteq x_i$. Therefore, by the fact that y_i is a minimal witness, $y_i \sqsubseteq f(x_i)$.

Since $\bot$ is also a minimal witness from y_i to $\lambda w.\, f(x_i \sqcup y_i \sqcup w)$, $P(x_{i-1}, y_{i-1})$ is true. Therefore we have $P(x, y)$. □

5.6 Examples and Counterexamples

In Lemma 5.3.2, we assumed f preserves directed infima. An example is $\langle a \rangle$.

Theorem 5.6.1. *Let $(Q, \{\to_a\}_{a \in \Sigma})$ be a labelled transition system and $a \in \Sigma$. Assume Q has finite branches labelled a i.e. for all $q \in Q$, a set $\{\, q' \mid q \to_a q' \,\}$ is finite.*

Then $\langle a \rangle$ preserves directed infima.

Proof. Let $\mathcal{S} \subseteq \mathcal{P}(Q)$ be a downward directed set. It is straightforward to show $\langle a \rangle \bigcap_{A \in \mathcal{S}} A \subseteq \bigcap_{A \in \mathcal{S}} \langle a \rangle A$. We prove the converse.

Let $q \in Q$ and assume $q \notin \langle a \rangle \bigcap_{A \in \mathcal{S}} A$, i.e. for all $q \to_a q'$, there exists $A_{q'} \in \mathcal{S}$ such that $q' \notin A_{q'}$. Since $\{\, q' \mid q \to_a q' \,\}$ is finite and $\mathcal{S}$ is downward directed, there exists $A \in \mathcal{S}$ such that for all $q \to_a q'$, $q' \notin A$. Therefore we have $q \notin \bigcap_{A \in \mathcal{S}} \langle a \rangle \mathcal{S}$. □

The theorem below shows that minimal witnesses correctly capture the proof strategy in Example 2.3.5 and Example 5.1.1.

Theorem 5.6.2. *Let $(Q, \{\to_a\}_{a\in\Sigma})$ be a labelled transition system and $a \in \Sigma$.*

For all $q \in Q$ and $x, z \subseteq Q$, z is a minimal witness from $\{q\}$ to $\lambda w.\, \langle a\rangle(x \cup w)$ if and only if one of the following conditions are satisfied:

- *$q \notin \langle a\rangle w$, $z = \{q'\}$, and $q \to_a q'$, or*
- *$q \in \langle a\rangle w$ and $z = \emptyset$.*

□

In addition, the theorem below shows that minimal witnesses correctly capture the intuition that the proof strategy in Example 5.1.1 fails.

Example 5.6.3. We consider the labelled transition system in Figure 5.1.

We define a test predicate P for $\langle *\rangle$. Let $x, y \subseteq Q$. Let $\bar{x} := \{\, i \mid q_i \in x \,\} \cup \{-1\}$ and $\bar{y} := \{\, i \mid q_i \in y \,\} \cup \{-2\}$

- If either $\bar{x}$ or $\bar{y}$ is unbounded, then $(x, y) \in P$.
- If both $\bar{x}$ and $\bar{y}$ are bounded and $\max x > \max y$, then $(x, y) \in P$.
- If both $\bar{x}$ and $\bar{y}$ are bounded and $\max x \leq \max y$, then $(x, y) \notin P$.

Lemma 5.6.4. *Example 5.6.3 satisfies the join closure property, the unfolding property, and the accumulation property.* □

Using this counterexample, we can conclude:

Theorem 5.6.5 (Failure of no-discovery parameterized coinduction)**.** *Unlike Theorem 5.5.1, it is not the case that the following proposition is true.*

- *Let $(C, \sqsubseteq)$ be a complete lattice and $f : C \to C$ be a monotone function. Assume $\lambda w.\, f(x \sqcup w)$ preserves directed infima. For all $x, y \in C$, the following are equivalent:*

  1. *For all test predicates P of f, if P satisfies the join closue property, the unfolding property, and the accumulation property, then $P(x, y)$ holds.*
  2. *$y \sqsubseteq G_f(x)$.*

□

Chapter 6

Conclusions and Future Work

We gave the inductive counterpart of parameterized coinduction. This can be used in finite state systems with decreasing parameters. As there are parameterized induction and coinduction, generalization to mixed induction and coinduction is of interest.

Parametric corecursion [13] is similar to parameterized coinduction, but it is defined in a category, not only in a preordered set. Therefore it can be used to define coinductive sets rather than coinductive propositions. It may be convenient to generalize parameterized induction to parametric recursion.

We gave the characterization of the power of parameterized coinduction and parameterized induction, and confirmed that parameterized coinduction is stronger than usual coinductive proof in this respect. This result may be extended to infinite state systems with looser finiteness condition, like symbolic automata [5].

Syntactic approaches to characterization of such an advantage can also be considered. One way is to provide a cut-eliminable calculus. Although there is already a cut-eliminable calculus for the propositional modal μ-calculus [10], another approach is worth investigating.

References

[1] A. Arnold and D. Niwiński. *Rudiments of μ-Calculus*, volume 146 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 2001.

[2] David Baelde and Dale Miller. Least and greatest fixed points in linear logic. In Nachum Dershowitz and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning, 14th International Conference, LPAR 2007, Yerevan, Armenia, October 15-19, 2007, Proceedings*, volume 4790 of *Lecture Notes in Computer Science*, pages 92–106. Springer, 2007.

[3] Ahmed Bouajjani, Javier Esparza, and Oded Maler. Reachability analysis of pushdown automata: Application to model-checking. In Antoni W. Mazurkiewicz and Józef Winkowski, editors, *CONCUR '97: Concurrency Theory, 8th International Conference, Warsaw, Poland, July 1-4, 1997, Proceedings*, volume 1243 of *Lecture Notes in Computer Science*, pages 135–150. Springer, 1997.

[4] Patrick Cousot and Radhia Cousot. Constructive versions of Tarski's fixed point theorems. *Pacific J. Math.*, 82(1):43–57, 1979.

[5] Loris D'Antoni and Margus Veanes. Minimization of symbolic automata. In Suresh Jagannathan and Peter Sewell, editors, *The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '14, San Diego, CA, USA, January 20-21, 2014*, pages 541–554. ACM, 2014.

[6] Federico Echenique. A short and constructive proof of tarski's fixed-point theorem. *Int. J. Game Theory*, 33(2):215–218, 2005.

[7] Wan Fokkink. *Introduction to Process Algebra.* Texts in Theoretical Computer Science. An EATCS Series. Springer, 2000.

[8] Ichiro Hasuo, Shunsuke Shimizu, and Corina Cîrstea. Lattice-theoretic progress measures and coalgebraic model checking. In Rastislav Bodik and Rupak Majumdar, editors, *Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2016, St. Petersburg, FL, USA, January 20 - 22, 2016*, pages 718–732. ACM, 2016.

[9] Chung-Kil Hur, Georg Neis, Derek Dreyer, and Viktor Vafeiadis. The power of parameterization in coinductive proof. In Roberto Giacobazzi and Radhia Cousot, editors, *The 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '13, Rome, Italy - January 23 - 25, 2013*, pages 193–206. ACM, 2013.

[10] Gerhard Jäger, Mathis Kretz, and Thomas Studer. Canonical completeness of infinitary mu. *J. Log. Algebr. Program.*, 76(2):270–292, 2008.

[11] Stephen C. Kleene. *Introduction to metamathematics.* Bibliotheca Mathematica. North-Holland, Groningen;Amsterdam;, 1952.

[12] Dexter Kozen. Results on the propositional mu-calculus. *Theor. Comput. Sci.*, 27:333–354, 1983.

[13] Lawrence S. Moss. Parametric corecursion. *Theor. Comput. Sci.*, 260(1-2):139–163, 2001.

[14] Vaughan R. Pratt. A decidable mu-calculus: Preliminary report. In *22nd Annual Symposium on Foundations of Computer Science, Nashville, Tennessee, USA, 28-30 October 1981*, pages 421–427. IEEE Computer Society, 1981.

[15] Davide Sangiorgi. *Introduction to Bisimulation and Coinduction.* Cambridge University Press, 2012.

[16] Colin Stirling. Bisimulation and logic. In Davide Sangiorgi and Jan Rutten, editors, *Advanced Topics in Bisimulation and Coinduction*, number 52 in Cambridge Tracts in Theoretical Computer Science, chapter 4, pages 173–196. Cambridge University Press, 2012.

[17] Colin Stirling and David Walker. Local model checking in the modal mu-calculus. *Theor. Comput. Sci.*, 89(1):161–177, 1991.

[18] Alfred Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific J. Math.*, 5:285–309, 1955.

[19] Thomas Wilke. Alternating tree automata, parity games, and modal μ-calculus. *Bull. Belg. Math. Soc. Simon Stevin*, 8(2):359–391, 2001. Journées Montoises d'Informatique Théorique (Marne-la-Vallée, 2000).

[20] Glynn Winskel. A note on model checking the modal nu-calculus. *Theor. Comput. Sci.*, 83(1):157–167, 1991.