University of South Florida

Honors Thesis

# An Examination of Induction on Failing Derivations in Relation to Coinduction and the Law of Excluded Middle

*Author:*
Edwin Peguero
U89344461
epeguero@mail.usf.edu

*Thesis Chair:*
Dr. Jay Ligatti

# 1 Introduction

**Recursive objects** are natural models for many phenomena in computer science and mathematics. Informally, recursive objects are defined by a collection of rules through which such objects may be constructed from other objects; such a collection of rules specifies a **recursive definition**. Important examples of recursive objects include lists, programs, graphs, and numbers, which may be composed of sub-lists, sub-expressions, sub-graphs, and smaller numbers, respectively. Recursive definitions admit two interpretations: as including only *finite constructions* or as additionally including *infinite constructions*. These interpretations induce, respectively, the *smallest* set of objects satisfying recursive definition, called the **inductive set**, and the *largest* such set, called the **co-inductive set** (Sangiorgi, 2011).

Often, theoretical research aims to prove that a property holds over sets of recursive objects; examples of such properties include primality of natural numbers, divergence (i.e., non-termination) of programs in a programming language, and bisimilarity, or behaviorally equivalence, among processes. Such research relies heavily on two proof techniques: **induction** and **co-induction**. Given some property $P(x)$ defined over a set $X$, a proof by induction shows that for some inductive set $X_I \subseteq X$, $\forall x \in X : x \in X_I \Rightarrow P(x)$, whereas a proof by co-induction shows that for some co-inductive set $X_C \subseteq X$, $\forall x \in X : P(x) \Rightarrow x \in X_C$ (Pierce, 2002). In the case of a rule-based recursive definition, an inductive proof consists in demonstrating that for any arbitrary inductive object, $x \in X$, and for each rule for building that object, $P(x)$ holds assuming that $P(x_i)$ holds for all sub-object, $x_i$, an assumption known as the **inductive hypothesis**. Dually, a co-inductive proof proceeds by showing that every element satisfying $P(x)$ is the conclusion of some rule for which $P(x_i)$ holds for each premise, $x_i$ (Sangiorgi, 2011). **Induction on failing derivations**, a recently developed proof technique due to Ligatti, proves that $\forall x \in X : x \notin X_I \Rightarrow P(x)$ by showing that for each way of constructing an object, and for every way in which that construction can fail, $P(x)$ holds, given the inductive hypothesis (Ligatti, 2013).

Ultimately, such proofs must be *convincing*, a criterion that has historically been a source of controversy for both philosophical and practical reasons. Concerns of the latter type have driven some computer scientists towards the school of logic known as **intuitionism**, which stands in opposition to **classicism**. *Intuitionistic logic* rejects the **law of excluded middle** (LEM), an axiom that states that for any statement, either it or its negation holds. On the other hand, *classical logic* embraces LEM, enabling the technique of proof by contradiction, through which a statement of the form $\neg P$ may be indirectly proven by showing $\neg\neg P$, i.e., that it's negation cannot hold. By allowing indirect proofs, classical logic allows for proofs that are *non-constructive*, i.e., that do not describe the construction of the objects in question. Such proofs may be less practical than *constructive* proofs, since these involve algorithms detailing the construction of objects.

By abstaining from LEM, intuitionistic logic differentiates between existential proofs that are constructive and those that are non-constructive; proofs of the former kind often yield useful algorithms as opposed to those of the latter kind (Iemhoff, 2015).

The goal of this paper is to examine an intuitionistic justification for induction and co-induction, and to explore their connection with induction on failing derivations. To this aim, the paper is divided as follows:

- Section 2 introduces *first-order logic*, a formal system commonly used in the literature to specify recursive definitions.
- Section 3 defines a set-theoretic framework based on the notion of *fixed points* for the purposes of defining and intuitionistically justifying induction and co-induction.
- Section 4 comments on induction on failing derivations in terms of the fixed point framework.

## 2 Formalizing Recursive Definitions

The persuasiveness of a logical argument must not rely on the eloquence and intuition of the arguer; rather, they must carry *objective necessity*. That is, proofs must be verifiable from the rules of argumentation, or **inference rules**, which specify how sentences can be deduced from previously proven sentences. Such a discussion of mathematics, itself being mathematical, enters the realm of *metamathematics*, which examines the mechanical codification, or *formalization*, of mathematics through purely effective (i.e., algorithmic, constructive, finitary) means (Kleene, 1971).

In this formalized view, proofs are naturally recursive objects: each inference rule specifies how a proof for a sentence may be built from smaller proofs. Each inference rule is composed of zero or more *premises* and one *conclusion*, and these compose a **deductive system**. A sentence is **derivable** if it is the conclusion of an inference rule for which all of the premises are also derivable. The set of all finitely derivable sentences forms an inductive set called a **theory**, and each sentence in the theory has a corresponding proof, or *derivation* (Chiswell & Hodges Wilfrid, 2007).

Deductive systems separate the symbolic structure, or *syntax*, of a sentence from its underlying meaning, or *semantics*. Hence, such systems formalize the process of proof verification: the derivability of sentences can be shown independently of any intuition or understanding of the underlying subject matter. Indeed, such proofs carry objective necessity, since it can be seen that they necessarily follow from successive applications of the inference rules.

### 2.0.1 First-order Logic

Often, sentences deducible from a deductive system are expressed in **first-order logic**, a formal language characterized by a syntax that resembles a simplified, unambiguous natural language. Each part of speech of first-order logic is recursively defined, making use of the following types of symbols (Leary, 1999):

- Logical Symbols
  - Logical Connectives: $\wedge, \vee, \neg, \rightarrow, \forall, \exists$
  - Equality: the binary relation $=$
  - Variables: a denumerable set of symbols $x_0, x_1, ..., x_n, ...$
- Non-logical Symbols
  - Constants: a denumerable set of symbols $c_0, c_1, ..., c_n, ...$
  - Function Symbols: zero or more symbols $f_1^{n_1}, f_2^{n_2}, ..., f_k^{n_k}, ....$ The superscript denotes the number of arguments the function takes, i.e. its *arity*, whereas the subscript identifies the function symbol.
  - Predicate Symbols: zero or more symbols $P_1^{n_1}, P_2^{n_2}, ..., P_k^{n_k}, ....$ The symbolic naming scheme is the same as that for functions.

The distinction between logical and non-logical symbols separates those symbols that carry a logical interpretation and those associated with some domain-specific interpretation, such as in a specific mathematical domain (e.g., algebra, group theory, number theory, etc.).

The "nouns", or *terms*, in first-order logic are defined as:

- Each variable and constant is a term.
- If $t_1, t_2, ...t_k$ are terms, then $f_k^{n_k} t_1 t_2 ... t_k$ is a term. Thus, functions produce new terms from other terms.

Finally, the "sentences", or *formula* are defined by combining terms with predicate "verbs" as follows:

- If $t_1, t_2, ...t_k$ are terms, then $P_k^{n_k} t_1 t_2 ... t_k$ is a formula.
- If $t_1$ and $t_2$ are terms, then $t_1 = t_2$ is a formula.
- If $\theta_1$ and $\theta_2$ are formulas, then $\theta_1 \wedge \theta_2$ is a formula.
- If $\theta$ is a formula and $x$ is a variable, then $\exists x \theta$ is a formula.
- ...similar criteria for the other logical connectives

Figure 1 depicts a deductive system for the natural numbers; sentences in the induced *first-order theory* state simply whether a string encodes a natural number. Intuitively, '$Z$' represents $0 \in \mathbb{N}$, '$S$' encodes the function $S(n) = n + 1$, and 'Nat($n$)' encodes the predicate $n \in \mathbb{N}$. From this example, it is evident that deductive systems can be used to define recursive objects. Moreover, Figure 2 shows that deductive systems can also be used to define *properties* of recursive objects.

Figure 1: Deductive System for Natural Numbers

$$\frac{}{\text{Nat}(Z)} \qquad \frac{\text{Nat}(n)}{\text{Nat}(S(n))}$$

Figure 2: Deductive System for the $\leq$ relation

$$\frac{}{Z \leq n} \qquad \frac{n \leq m}{S(n) \leq S(m)}$$

# 3 Fixed Point Framework for Recursive Sets

The theory of *fixed points* provides a constructive, set-theoretic framework for defining recursive sets and for justifying the principles of induction and co-induction. Recursive definitions through deductive systems are particularly amenable to fixed-point representation, as they can be often be encoded with sets (Sangiorgi, 2011).

## 3.1 Background: Tarski's Fixed Point Theorem

In the language of lattice theory, a set $L$ forms a *complete lattice* iff it is a *partially ordered set* (i.e., a set equipped with a relation, $\leq$, that is reflexive, transitive, and antisymmetric) with the property that $\forall S \in \mathcal{P}(L) : (\vee S \in \mathcal{L}) \wedge (\wedge S \in \mathcal{L})$, where $\vee S$ is the **join** (least upper bound) and $\wedge S$ is the **meet** (greatest lower bound) of $S$, respectively. *Complete lattices* have both a top ($\top$) and bottom ($\bot$) element, given by $\top = \vee\mathcal{L}$ and $\bot = \wedge\mathcal{L}$.

One particularly important complete lattice, the **powerset lattice**, is induced by an arbitrary set $\mathcal{U}$ by letting $L \equiv \mathcal{P}(\mathcal{U})$, $\leq \equiv \subseteq$, $\wedge \equiv \cap$ and $\vee \equiv \cup$. For the purposes of specifying recursive definitions, we will consider the powerset lattice, $\mathcal{L}$ induced by a fixed universal set, $\mathcal{U}$, that contains the desired recursive sets; thus, the inductive and coinductive sets are both elements of $\mathcal{L}$. Furthermore, we will define an **endofunction** $F : \mathcal{L} \mapsto \mathcal{L}$ such that $\forall x \in \mathcal{U} : x \in F(T)$ if and only if $x$ is constructible from objects in $T$ in at most one application of a construction rule. Thus, $F$ can be viewed as an operation that performs two

dual functions: it both *builds* all objects constructible from at most one rule application using elements in $T$ and *filters* those malformed objects that cannot be justified, or **observed**, by the application of one rule involving objects already in $T$. Given this behavioral description of $F$, we will assume that $F$ is **monotone**, i.e., that $A \subseteq B \Rightarrow F(A) \subseteq F(B)$; this is intuitive, since any extension of an arbitrary set $A$ should allow for the construction of at least $F(A)$ in at most one rule.

In the case that $F(T) \subseteq T$, that is, when $T$ is a **prefixed point**, every object constructible from $T$ in at most one rule application is already in $T$. By the monotonicity of $F$, this property is extendes so that every object constructible from objects in $T$ in any finite number of rule applications is already in $T$; this fact emerges from the fact that $\forall n \in \mathcal{N} : F^n(T) \subseteq ... \subseteq F(F(T)) \subseteq F(T) \subseteq T$. For this reason, we say that prefixed points are **complete**. Using this notion, we can define the set consisting of exactly all finite constructions, i.e. the *inductive set*, as the **smallest complete set** defined by $\wedge\{T \in \mathcal{L} | F(T) \subseteq T\}$. The *smallest* constraint precludes the presence of malformed objects from inhabiting the inductive set, since any such object is superfluous for completeness and is therefore eliminated by the intersection.

Dually, when $T \subseteq F(T)$, that is, when $T$ is a **postfixed point**, the construction of each object in $T$ is *observable* in at most one rule application using objects in $T$. Monotonicity extends this property such that every object in $T$ is observable in an infinite number of steps, since for every ordinal number $\omega : T \subseteq F(T) \subseteq F(F(T)) \subseteq ... \subseteq F^{\omega}(T)$. Since every object in $T$ is infinitely observable, that is, all objects are well-formed, we say that postfixed points are **sound**. This notion can be used to define the set consisting exactly all possible constructions, i.e. the **coinductive set**, as the **largest sound set** defined by $\vee\{\, T \in \mathcal{L} \,|\, T \subseteq F(T) \,\}$. The *largest* constraint guarantees the presence of all well-formed object in the coinductive set, since any such object resides in a sound set captured by the union.

Clearly, we expect that the inductive set be sound, the coinductive set be complete; that is, we expect that both sets be **fixed points**. The existence of these sets and their status as fixed points is guaranteed by Tarski's Fixed Point Theorem (Sangiorgi, 2011):

***Tarski's Fixed Point Theorem:*** If $L$ is a complete lattice and $F : \mathcal{L} \mapsto \mathcal{L}$ is a monotone endofunction, then the set of fixed points forms a lattice (illustrated in Figure 3) with least fixed point ($lfp(F)$) and greatest fixed point ($gfp(F)$) such that:

- $lfp(F) = \wedge\{\, T \in \mathcal{L} \,|\, F(T) \subseteq T \,\}$
- $gfp(F) = \vee\{\, T \in \mathcal{L} \,|\, T \subseteq F(T) \,\}$

□

Thus, this fixed point framework provides a natural definition for recursive sets:

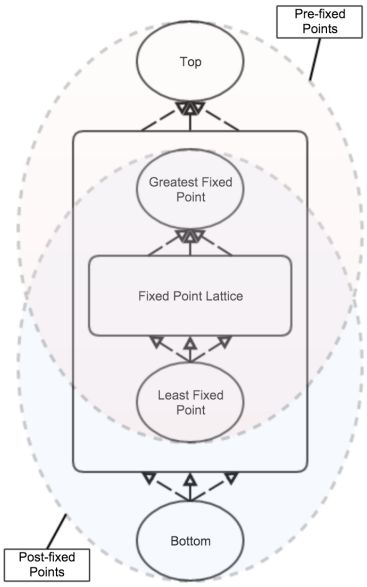


Figure 3: Fixed Point Lattice Induced by Monotone Endofunction

Inductive set on a complete lattice, $L$, with respect to $F : L \mapsto \mathcal{L}$:

$$F_{in} = \wedge\{\, T \in \mathcal{L} \,|\, F(T) \subseteq T \,\}$$

Co-inductive set on a complete lattice, $L$, with respect to $F : L \mapsto L$:

$$F_{co} = \vee\{\, T \in \mathcal{L} \,|\, T \subseteq F(T) \,\}$$

Furthermore, these definitions naturally give rise to the principles of induction and coinduction:

Principle of Induction: $F(T) \subseteq T \Rightarrow F_{in} \subseteq T$

Principle of Coinduction: $T \subseteq F(T) \Rightarrow T \subseteq F_{co}$

The formal justification for these principles follows immediately from the definition of meet and join.

*Proof.* Let $T \in \mathcal{L}$, and suppose $F(T) \subseteq T$. Then $F_{in} \subseteq T$, since $F_{in}$ is a lower bound of all prefixed points. Similarly, if we suppose $T \subseteq F(T)$, then $T \subseteq F_{co}$, since $F_{co}$ is an upper bound of all postfixed points. □

## 3.2 Proof by Rule (Co-)Induction

We now examine how recursive sets specified by deductive systems may be codified in terms of fixed points. Let $\mathcal{U}$ contain the predicate or relation defined by the deductive system. We encode inference rules as a set

Figure 4: Set Encoding of Inference Rules for Natural Numbers

$$\mathcal{U} = \{Z, S, (,)\}^*$$
$$R = \{(\emptyset, Z)\} \cup \{(\{n\}, S(n)) : \forall n \in \mathcal{U}\}$$

$R \subseteq \mathcal{P}(\mathcal{U}) \times \mathcal{U}$ of *ground rules*. Each ground rule $(S, x) \in R$ represents a substitution of a recursive object for the variables in an inference rule, where $S$ is the set of sub-objects corresponding to the premises, from which the object corresponding to the conclusion, $x$ follows. Figure 4 demonstrates the set encoding of the 'Nat' predicate defined in the deductive system of Figure 1.

We simulate the construction of objects with the endofunction $\Phi_R$, called the **rule functional**, defined over the powerset lattice induced by $\mathcal{U}$ as:

$$\Phi_R(T) = \{\, x \in \mathcal{U} \mid \exists (S, x) \in R : S \subseteq T \,\}$$

The principles of **rule induction** and **rule co-induction** can be gleaned from the principles of induction and coinduction by letting $T \equiv \{x \in \mathcal{U} \,|P(x)\} \in \mathcal{P}(\mathcal{U})$, the set induced by an arbitrary predicate $P(x)$ over $\mathcal{U}$ (Pierce, 2002):

- Principle of Rule Induction:

$$[\forall (S, x) \in R : (\forall x' \in S : P(x')) \Rightarrow P(x)] \Rightarrow [\forall x \in \mathcal{U} : x \in \Phi_{Rin} \Rightarrow P(x)]$$

- Principle of Rule Coinduction:

$$[\forall x \in \mathcal{U} : P(x) \Rightarrow (\exists (S, x) \in R : \forall x' \in S : P(x'))] \Rightarrow [\forall x \in \mathcal{U} : P(x) \Rightarrow x \in \Phi_{Rco}]$$

*Proof.* Let $T$ be the set induced by some arbitrary predicate, $P(x)$.

To prove the principle of rule induction, we must show that $\forall x \in \mathcal{U} : x \in \Phi_{Rin} \Rightarrow P(x)$, assuming that $\forall (S, x) \in R : (\forall x' \in S : P(x')) \Rightarrow P(x)$. By applying the definition of both $T$ and $\Phi_R$, we have that:

$$\forall (S, x) \in R : (\forall x' \in S : x' \in T) \Rightarrow x \in T \iff$$

$$\forall (S, x) \in R : S \subseteq T \Rightarrow x \in T \iff$$

$$\forall x \in \Phi_R(T) : x \in T \iff$$

$$\Phi_R(T) \subseteq T$$

Thus, by the principle of induction, we have that $\Phi_{Rin} \subseteq T$. It follows then, that for an arbitrary $x \in \Phi_{Rin}$, $x \in T$, or equivalently, $P(x)$.

Similarly, to prove the principle of rule coinduction, we must show $\forall x \in \mathcal{U} : P(x) \Rightarrow x \in \Phi_{Rco}$, assuming that $\forall x \in \mathcal{U} : P(x) \Rightarrow (\exists (S, x) \in R : \forall x' \in S : P(x'))$. By applying the definition of both $T$ and $\Phi_R$, we have that:

$$\forall x \in \mathcal{U} : x \in T \Rightarrow (\exists (S, x) \in R : \forall x' \in S : x' \in T) \iff$$

$$\forall x \in T : (\exists (S, x) \in R : S \subseteq T) \iff$$

$$\forall x \in T : x \in \Phi_R(T) \iff$$

$$T \subseteq \Phi_R(T)$$

Thus, by the principle of coinduction, we have that $T \subseteq \Phi_{Rco}$. It follows then, that for any arbitrary $x \in \mathcal{U}$ with $P(x)$, since $x \in T$ by definition of $T$, it follows that $x \in \Phi_{Rco}$.

Note that, at first glance, the structure of a coinductive proof seems to be operate dually to an inductive proof: the latter requires a proof by case analysis on each inference rule, whereas the former seemingly requires the satisfaction of just one case. However, this cursory view neglects the fact that each coinductive case incurs a loss of generality by assuming a specific form object form; thus, every inference rule must be analyzed independently to recover generality. One key difference lies in the treatment of axioms: since they lack premises, their respective cases can be omitted.

□

We now illustrate how familiar inductive and co-inductive proofs emerge from the machinery of the fixed-point framework. Normally, such proofs proceed without explicit reference to this underlying machinery.

**Example 1: Proof by Induction**

We prove the antisymmetry of the $\leq$ relation from Figure 2.

**Antisymmetry Lemma**: $\forall n, m \in \mathbb{N} : (n \leq m \wedge m \leq n) \Rightarrow n = m$

Proof. We can recognize the lemma as a candidate for a proof by induction by letting:

$\mathcal{U} \equiv \mathbb{N} \times \mathbb{N}$

$F_{in} \equiv lfp(\Phi_{\leq})$

$P(n, m) \equiv (m \leq n \Rightarrow n = m)$

Where, $\Phi_{\leq}$ denotes the rule functional induced by the rules of $\leq$:

$$\leq \equiv \{(\emptyset, (Z, n)) \mid \forall n \in \mathbb{N}\} \cup \{(\{(n, m)\}, (S(n), S(m))) \mid \forall n, m \in \mathbb{N}\}$$

Thus, the lemma can be restated as: $\forall (n, m) \in \mathbb{N} \times \mathbb{N} : (n, m) \in F_{in} \Rightarrow P(n, m)$. By the principle of induction, it suffices to perform a proof by rule induction on the derivation of $n \leq m$; that is, we must show: $\forall (S, (n, m)) \in \leq: (\forall (n', m') \in S : P(n', m')) \Rightarrow P(n, m)$.

Let $(S, (n, m)) \in \leq$. Then $(S, (n, m)) \in \{(\emptyset, (Z, m)) \mid \forall m \in \mathbb{N}\}$ or $\{(\{(n', m')\}, (S(n'), S(m'))) \mid \forall (n', m') \in \mathbb{N} \times \mathbb{N}\}$. Thus, an analysis of the ground rules corresponds to a case analysis of each of the inference rules in the deductive system; for each case, we must conclude $P(x)$.

Case 1: $(S, (n, m)) \in \{(\emptyset, (Z, m) \mid \forall m \in \mathbb{N}\}$

In this case, $n = Z$. By assumption, we have $m \leq Z$; therefore, there is some $S \subseteq \mathbb{N} \times \mathbb{N}$ such that $(S, (m, Z)) \in \leq$. By inspection of the rules, we observe that, since there is no $y \in \mathbb{N}$ such that $Z = S(y)$, it must be the case that $S = \emptyset$ and $m = Z$. Therefore, $n = m$.

Case 2: $(S, (n, m)) \in \{(\{(n', m')\}, (S(n'), S(m'))) \mid \forall n', m' \in \mathbb{N}\}$

In this case, $(n, m) = (S(n'), S(m'))$ and $(n', m') \in \mathbb{N} \times \mathbb{N}$. In addition, we may assume the inductive hypothesis: $m' \leq n' \Rightarrow n' = m'$. By assumption, we have $S(m') \leq S(n')$, and, as in the previous case, an inspection of the rules reveals that it must be the case that $m' \leq n'$. By the inductive hypothesis, then, we have that $m' = n'$, and so $S(n') = S(m')$. □

It is interesting to note that, rather than perform rule induction on the derivation of $n \leq m$, one could alternatively perform rule induction on the derivation of $\mathrm{Nat}(n)$ by making the following selections:

$\mathcal{U} \equiv \{S, Z, (,)\}^*$

$F_{in} \equiv \mathbb{N}$

$P(n) \equiv \forall m \in \mathbb{N} : (n \leq m \wedge m \leq n) \Rightarrow n = m$

**Example 2: Proof by Co-induction**

Consider the recursive definitions in Figures 5 and 6. The former defines the set of lists constructible from elements of a set $A$, whose elements have a natural, transitive ordering, $\leq$. The co-inductive interpretation of this definition yields the set of both finite and infinite lists: the set $A^{\omega}$. The latter defines a lexicographical ordering on lists, where $hd(\sigma)$ and $tl(\sigma)$ refer to the elements on either side of the ':' operator.

We now prove the transitivity of lexicographical ordering for infinite lists, as stated by the following lemma:

Figure 5: Recursive Definition for $A^\omega$

$$\frac{}{\text{nil} \in A^\omega} \qquad \frac{\tau \in A^\omega \qquad a \in A}{a : \tau \in A^\omega}$$

Figure 6: Recursive Definition for $\leq_L$

$$\frac{}{nil \leq_L \sigma} \qquad \frac{hd(\sigma) \leq hd(\tau) \qquad hd(\sigma) = hd(\tau) \Rightarrow tl(\sigma) \leq_L tl(\tau)}{\sigma \leq_L \tau}$$

**Transitivity Lemma**: $\forall \sigma, \tau \in A^\omega : (\exists \rho \in A^\omega : \sigma \leq_L \rho \leq_L \tau) \Rightarrow (\sigma \leq_L \tau)$

*Proof:* We can recognize the statement as a candidate for a proof by rule co-induction by letting:

$\mathcal{U} \equiv A^\omega \times A^\omega$

$P(\sigma, \tau) \equiv \exists \rho \in A^\omega : \sigma \leq_L \rho \leq_L \tau$

$F_{co} \equiv gfp(\Phi_{\leq_L})$

Where $\Phi_{\leq_L}$ is the rule functional induced by the ground rules:

$$\leq_L \equiv \{ \, ( \, \{(tl(\sigma), tl(\tau)) \mid hd(\sigma) = hd(\tau)\}, (\sigma, \tau)) \mid \forall (\sigma, \tau) \in A^\omega \times A^\omega : hd(\sigma) \leq hd(\tau)\} \cup \{(\emptyset, (nil, \sigma)) \mid \forall \sigma \in A^\omega\}$$

Thus, the lemma can be restated as: $\forall (\sigma, \tau) \in \mathcal{U} : P(\sigma, \tau) \Rightarrow (\sigma, \tau) \in \Phi_{\leq_L co}$. By the principle of co-induction, it suffices to perform a proof by rule co-induction on the derivation of $\sigma \leq_L \tau$; that is, we must show: $\forall (\sigma, \tau) \in \mathcal{U} : P(\sigma, \tau) \Rightarrow \exists (S, (\sigma, \tau)) \in \leq_L : \forall (\sigma', \tau') \in S : P(\sigma', \tau')$.

Let $(\sigma, \tau) \in A^\omega \times A^\omega$ be such that $\exists \rho \in A^\omega : \sigma \leq_L \rho \leq_L \tau$, i.e., the **coinductive hypothesis**. We must find a ground rule $(S, (\sigma, \tau)) \in \leq_L$ such that $\forall (\sigma', \tau') \in S : P(\sigma', \tau')$. As explained earlier, this corresponds to a case analysis of the ground rules, where each case requires the construction of the premises, $S$, and a proof of $P(x)$ for each $x \in S$, in addition to any side conditions:

Case 1: $(S, (\sigma, \tau)) \in \{(\emptyset, (nil, \sigma)) \mid \forall \sigma \in A^\omega\}$

In this case we have $\sigma = nil$, and so the proof follows immediately since $(\emptyset, (nil, \tau)) \in \leq_L$. As this case carried no proof obligations, it could have been omitted.

Case 2: $(S, (\sigma, \tau)) \in \{ \, ( \, \{(tl(\sigma), tl(\tau)) \mid hd(\sigma) = hd(\tau)\}, (\sigma, \tau)) \mid \forall (\sigma, \tau) \in A^\omega \times A^\omega : hd(\sigma) \leq hd(\tau)\}$

In this case, we have three proof obligations:

**Construction of Premises** $S$: $(\{(tl(\sigma), tl(\tau)) \mid hd(\sigma) = hd(\tau)\}, (\sigma, \tau)) \in \leq_L$

**Proof of Side Condition**: $hd(\sigma) \leq hd(\tau)$

**Show Property on Premises**: $\forall(\sigma', \tau') \in \{(tl(\sigma), tl(\tau)) \,|hd(\sigma) = hd(\tau)\} : P(\sigma', \tau')$

The side condition follows immediately from the coinductive hypothesis, which gives us $hd(\sigma) \leq hd(\rho) \leq hd(\tau)$, and the transitivity of $\leq$.

Now, we consider two subcases: $hd(\sigma) = hd(\tau)$ and $hd(\sigma) \neq hd(\tau)$. The latter case is immediately proven by the fact that there are no premises.

Thus, suppose $hd(\sigma) = hd(\tau)$. Then we have $hd(\sigma) = hd(\rho)$ and $hd(\rho) = hd(\tau)$. By inspection of the rules $\leq_L$ in the coinductive hypothesis, and the fact that neither $\sigma$ nor $\tau$ can be $nil$, since $hd(nil)$ is undefined, it follows that $hd(\sigma) = hd(\rho) \Rightarrow tl(\sigma) \leq_L tl(\rho)$ and $hd(\rho) = hd(\tau) \Rightarrow tl(\rho) \leq_L tl(\tau)$. Thus, we have $tl(\sigma) \leq_L tl(\rho) \leq_L tl(\tau)$, satisfying the final proof obligation: $\exists\rho \in A^\omega : tl(\sigma) \leq_L \rho \leq_L tl(\tau)$.

□

# 4 Proof by Induction on Failing Derivations

A proof by induction on failing derivations proves $\forall x \in X : P(x) \Rightarrow x \in F_{in}$ by proving the contrapositive of $\forall x \in X : x \in (F_{in})^C \Rightarrow \neg P(x)$ by induction, where '$^C$' denotes the set complement operation.

- Proof by Induction on Failing Derivations:

$$[\forall x \in X : x \in (F_{in})^C \Rightarrow \neg P(x)] \Rightarrow [\forall x \in X : P(x) \Rightarrow x \in F_{in}]$$

The technique therefore relies on both a *proof by contrapositive* and on a proof by induction. In general, the former proof technique depends on a meta-theoretical application of the LEM, which arises from the fact that a proof by contrapositive demonstrates $P \Rightarrow \neg\neg Q$, but $\neg\neg Q \Rightarrow Q$ (i.e., the principle of *double negation*) is sufficient to prove LEM. We can remove this dependence by proving $\forall x \in X : P(x) \vee \neg P(x)$ for the particular predicate, $P$.

On the other hand, the latter proof technique requires that $(F_{in})^C$ be an inductive set. Intuitively, $(F_{in})^C$ is inductive when the derivation of each $C \notin F_{in}$ fails finitely. In such cases, we expect that there should be a correspondingly finite refutation. In general, however, a derivation may fail infinitely; in such cases, we do not expect for $(F_{in})^C$ to be inductive. We capture this intuition in the context of a monotonic rule functional, $\Phi_R$, by defining a complement rule functional, $\Phi_{R'}$, induced by the ground rules of the complement deductive system, $R'$, and claim that $lfp(\Phi_R)^C = gfp(\Phi_{R'})$. We conjecture that $lfp(\Phi_R)^C$ is inductive, and therefore that proof by induction on failing derivations is a sound proof technique, iff $lfp(\Phi_{R'}) = lfp(\Phi_{R'})$.

In addition, we observe that, in the case where $F_{in} = F_{co}$, proofs by induction on failing derivations may replace proofs by rule co-induction, since in this case, the proof goal of both techniques coincide.

## 5 Conclusion and Future Work

We have seen how induction and co-induction emerge from recursive definitions through the lens of fixed point theory. Moreover, we have conjectured on the soundness of induction on failing derivations. As of yet, the status of the conjecture remains unknown. A positive answer to the conjecture may further motivate the development of alternative proof techniques.

## References


Chiswell, I., & Hodges Wilfrid. (2007). *Mathematical Logic*. Oxford, NY: Oxford University Press.

Iemhoff, R. (2015). Intuitionism in the philosophy of mathematics. In E. N. Zalta (Ed.), *The stanford encyclopedia of philosophy* (Spring 2015 ed.). `http://plato.stanford.edu/archives/spr2015/entries/intuitionism/`.

Kleene, S. C. (1971). *S.C. Kleene Introduction to Metamathematics*. Wolters-Noordhoff Publishing.

Leary, C. (1999). *A friendly introduction to mathematical logic*. Prentice Hall.

Ligatti, J. (2013). *Induction on failing derivations* (Tech. Rep.). Technical report, University of South Florida.

Pierce, B. C. (2002). *Types and Programming Languages*. The MIT Press.

Sangiorgi, D. (2011). *Introduction to Bisimulation and Coinduction*. Cambridge University Press.