Constructive substitutes for Kőnig's lemma

Dominique Larchey-Wendling
Université de Lorraine, CNRS, LORIA, Vandœuvre-lès-Nancy, France

Abstract

We propose weaker but constructively provable variants of the contrapositive of Kőnig's lemma. We derive those from a generalization of the FAN theorem for inductive bars to inductive covers, for which we give a concise proof. We compare the positive, negative and sequential characterizations of covers and bars in classical and constructive contexts, giving precise accounts of the role played by the axioms of excluded middle and dependent choice. As an application, we discuss some examples where the use of Kőnig's lemma can be replaced by one of our weaker variants to obtain fully constructive accounts of results or proofs that could otherwise appear as inherently classical.

2012 ACM Subject Classification Theory of computation → Type theory

Keywords and phrases König's lemma, FAN theorem, constructive mathematics, inductive covers, inductive bars, almost full relations, inductive type theory, Coq

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

Funding *Dominique Larchey-Wendling*: partially supported by NARCO (ANR-21-CE48-0011)

1 Introduction

Kőnig's (infinity) lemma, named after Dénes Kőnig, was originally published as a theorem of graph theory [14]. Nowadays, it is usually conflated with the following statement:

Any infinite tree which is finitely branching has an infinite branch[1]

The restriction to at most binary trees is of particular importance because it can be stated within lightweight foundations like e.g. RCA_0 [22], and is usually called weak Kőnig's lemma (WKL). Notice that Kőnig's lemma is also used in its contrapositive form:

Any finitely branching tree with only finite branches must be finite.

Classical mathematicians would not mind switching between the two formulations but herein, we refrain from using excluded middle at will, and we adopt a constructivist point of view. In this context, that contrapositive form is sometimes referred to as "Brouwer's FAN theorem" [5, p. 13]. Although there is no universal agreement on what constitutes constructive mathematics, we use the inductive type theory that is the basis of Coq, free of additional axioms, as our constructive foundations.

Kőnig's lemma plays critical roles in various fields of mathematics like logic, computability, tiling theory, etc. and has been investigated by reverse mathematics, e.g. as WKL_0 in [22], and constructive reverse mathematics [3, 2]. Although some of our investigations might be relevant to the program of reverse mathematicians, we do not follow that approach. We favour a more pragmatic perspective: since the lemma does not belong to the realm of purely constructive mathematics, can we propose *weaker* alternatives that could be used, not as drop-in, but rather as low cost replacements for Kőnig's lemma? Of course, we require that those alternatives are constructively provable.

Kőnig's lemma can (in particular) be used to establish the termination of algorithms, typically the decision procedure for implicational relevance logic [8, 16]. It is instrumental to

[1] the original statement rather talks about paths in a graph.

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:22
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

show the existence of Harvey Friedman's [10] TREE(n) monster (extremely fast growing) function, in combination with Kruskal's tree theorem, see e.g. [11]. These are two example applications of our tools aimed at giving constructive accounts of results and proofs that could otherwise look inherently classical.

As simple as it sounds, Kőnig's lemma involves the notion of infinite tree. Hence, the trees cannot simply be understood as the inductively defined structure to be found in computer science (these are always finite). Also, the notion of infinite is not as straightforward in the constructive world. In reverse mathematics, where the usage of versatile data structures may be constrained, a tree is often conflated with its set of finite branches (so finite sequences of nodes where the next node is a son of the current node). As such, trees are non-empty, prefix closed, sets of finite sequences, possibly with a computable membership predicate. And infinite branches are sequences for which every finite prefix belongs to the tree, i.e. the upper limit of a growing sequence of finite branches of the tree.

In that context, one can prove Kőnig's lemma using excluded middle and a weak form of the axiom of choice (e.g. dependent choice). If a canonical choice can be made over the sons, typically when there is a total order that can sort the sons at every node of the tree, the infinite construction process in the proof can be determinized (by choosing the least son) and the reliance on the axiom of choice is avoidable in that case. However, excluded middle is more critical, in particular to show that when the union of finitely many subtrees is infinite, it must be because one of them is infinite. "Being infinite" is not a decidable property so the selection performed by excluded middle cannot be turned into a computable value.

Kleene [15] famously gave a counterexample to a computational interpretation of weak Kőnig's lemma: he builds a computable infinite binary tree, so a decidable set of finite sequences of Booleans,[2] for which there exists no computable infinite branch, i.e. no infinite sequence of Booleans of which every finite prefix belongs to the tree. This gives a very strong argument against the constructive acceptability of Kőnig's lemma, at least when one "interprets Bishop's mathematics in a recursive way" [6].[3]

Not only Kőnig's lemma could be rejected from a constructivist point of view, but some of its consequences suffer similar defects. Consider the compactness result for Wang tilings:

A finite set of tiles can tile the plane if and only if it can tile any finite square.

Similarly to Kleene's result, Hanf [12] and Myers [21] famously gave examples of finite sets of tiles that can tile the whole plane, but only in a nonrecursive way. This invalidates a computational understanding of the compactness result. Hence no constructive account of the proof of the compactness result can be given, otherwise it would entail the existence of a recursive tiling.[4]

So there is no real hope at a drop-in constructive replacement for Kőnig's lemma because some of its consequences might live outside the realm of constructive or computable mathematics. Nevertheless, we argue that it might be used in contexts where weaker alternatives would also fit. And it is our aim here to explore some of those alternatives.

For instance, there is an interpretation of its contrapositive form, i.e. "any finitely branching tree with only finite branches must be finite," where the notion of infinity is replaced by finitary notions. Notice that the referred statement still relies on arbitrary (finite

[2] choices between the left or the right son.
[3] as said earlier, the notion of what is constructively acceptable is not universally agreed on.
[4] Notice that the tileability of a finite square is a decidable property.

or infinite) trees: when saying "only finite branches," one must consider the possibility that it contains infinite branches otherwise this hypothesis is vacuous:

- one classical way to understand "only finite branches" is by saying that no infinite sequence can have all its finite prefixes in the tree. Hence even though the statement does not refer to infinity, it is hidden in this unfolding;
- another way is to understand "only finite branches" is to give a characterization of the finiteness of branches using the inductive `acc`(essibility) predicate, see Section 3.2:

$$\frac{\forall y,\, F\,x\,y \to \mathtt{acc}\,F\,y}{\mathtt{acc}\,F\,x}$$

  where $F\,x\,y$ means that x is the father of y in the tree (or y is a son of x). If nodes are conflated with finite branches/sequences, then $F\,x\,y$ simply means that y has the shape $x \mathbin{+\!\!+} [\_]$, i.e. x followed by a single choice of branching/son.

In that later case, finiteness can be defined by ($\mathtt{acc}\,R\,\mathit{root}$), and thus understood as the unavoidable termination of the nondeterministic process of expending branches by adding sons after sons, starting from the *root*. In that inductive understanding of "only finite branches," the contrapositive of Kőnig's lemma can be established by well founded induction, see e.g. [1, p. 15]. We will derive it as a corollary in Section 5.3.

Intuitionists have compared (weak) Kőnig's lemma with Brouwer's Fan theorem [23, 24], itself a consequence of the Bar theorem, originally designed to grasp the full continuum in an intuitionistic approach to real analysis [5, 23].

We do not assume Brouwer's real thesis on bars [23]: one way to understand the thesis is to say that every bar is an inductive bar. Rather, following Coquand [6] (see discussion in Section 3.5), we choose to work directly with inductive bars (on finite sequences), avoiding Brouwer's axiom completely. Actually, we start working with the more general notion of inductive cover [4] on (transition) relations.

As for our contributions, in Section 3 we show that notion of inductive cover generalizes both inductive bars and (inductive) accessibility, w.r.t. its definition as well as w.r.t. the results that it entails. We then give a detailed comparison between the constructive and classical strength of three characterizations of covers: positive, negative and sequential. In particular, for the classical part of the comparison, we separate the role played by excluded middle (XM) and dependent choice (DC) and show the key role played by the intermediate negative characterization. This negative characterization will also play an important role in a constructive context, as a substitute to the sequential characterization, when used in combination with the FAN theorem.

In Section 4, we give a type theoretic interpretation of the FAN theorem for inductive covers, with a concise proof. The central argument, the stability of upward closed inductive covers under binary union, differs from that of the proof of Fridlender's FAN theorem for inductive bars [9] which relies on the stability of monotone inductive bars under binary intersection. However, we derive the FAN for bars as an instance of the FAN for covers, to make the generalization explicit.

In Section 5, we exploit the FAN for inductive covers, followed by an application of the negative characterization of covers, to give several weaker versions of (the contrapositive of) Kőnig's lemma, showing how relations can be represented by rose trees (hence finitary). This includes an extra covering assumption, or an extra bar assumption, or else an extra almost fullness assumption.

In Section 6, we give two examples where Kőnig's lemma can successfully be replaced with one of these weaker variants to give constructive accounts of results of which the former proofs where using the classical form of the lemma.

Additionally, we contribute a mechanization of all the results of the paper in a Coq script that can of course be type checked for correctness, but was especially designed to be read by humans, not only by computers. The script is mostly self contained, largely commented, with concise proofs: the longest is 25 loc but most of them are shorter than 10 loc. It is accessible under a free software license at

`https://github.com/DmxLarchey/Constructive-Konig`

2 Coq preliminaries

We denote by $\mathbb{P}$ the type of propositions and simply by `Type` the Coq hierarchy of types, as usual with this framework. We write $\bot : \mathbb{P}$ for the empty proposition and use the standard notations for logical connectives. Recall that the logic of Coq is intuitionistic hence the negation is defined by $\neg P \coloneqq P \to \bot$. Following the BHK interpretation, $X \to Y$ more generally denotes the type of maps from X to Y, and write $\forall x : X,\ P\,x$ for the dependent product, irrelevant of whether $P : X \to \mathbb{P}$ or $P : X \to \mathtt{Type}$. Whenever it can be guessed, the type annotation in $x : X$ is simply avoided. The dependent sum has several flavors in Coq: for $P : X \to \mathbb{P}$ we have the proposition $\exists x,\ P\,x : \mathbb{P}$ and the type $\{x \mid P\,x\} : \mathtt{Type}$ which behave somewhat similarly but are however fundamentally different because proposition cannot systematically be eliminated to build terms of sort `Type`.[5]

The type of Peano natural numbers $\mathbb{N}$ is inductively defined in Coq as $\mathbb{N} : n \coloneqq 0 \mid \mathtt{S}\,n$ and arithmetic in Coq, which we assume, is build on this type. We will mainly use it as indices for infinite sequences and we favor writing $1 + n$ over $\mathtt{S}\,n$ (they are identical by definition).

We will manipulate finite sequences as lists, polymorphic over the carrier type X,[6] in the inductive type $\mathtt{list}\,X : \ l \coloneqq [] \mid x :: l$ where $x : X$. The constructors are $[] : \mathtt{list}\,X$ for the empty list and $\cdot :: \cdot : X \to \mathtt{list}\,X \to \mathtt{list}\,X$. These notations $[]/::$ correspond to the names `nil`/`cons` in vanilla Coq. Additionally, list concatenation (resp. membership) is named `app` (resp. `In`), denoted infix by $\cdot +\!\!+ \cdot : \mathtt{list}\,X \to \mathtt{list}\,X \to \mathtt{list}\,X$ (resp. $\cdot \in \cdot : X \to \mathtt{list}\,X \to \mathbb{P}$), and defined by a guarded fixpoint. Moreover, we use the reverse $\mathtt{rev} : \mathtt{list}\,X \to \mathtt{list}\,X$ and the length $\lfloor\cdot\rfloor : \mathtt{list}\,X \to \mathbb{N}$ functions as well as the permutation relation $\cdot \sim_p \cdot : \mathtt{list}\,X \to \mathtt{list}\,X \to \mathbb{P}$.[7]

We define finiteness as a property $\mathtt{finite}\,P : \mathbb{P}$ of unary relations (view as sets):[8]

$$\mathtt{finite}\ \{X\}\ (P : X \to \mathbb{P}) \coloneqq \exists l,\ \forall x,\ P\,x \leftrightarrow x \in l$$

i.e. there exists a list spanning the relation P. This characterization of finiteness as listability is equivalent *Kuratowski finiteness* but much easier to manipulate formally. The list l is not unique in general, unless P is empty. The `finite` property is $\mathbb{P}$-bounded herein, so the list l *can only* be recovered when building a value of sort $\mathbb{P}$, and *not* when of sort `Type`.

We manipulate relations as functions outputting propositions, hence we denote by $\mathtt{rel}_2\,X\,Y \coloneqq X \to Y \to \mathbb{P}$ the type of heterogeneous binary relations between X and Y. In

[5] For $P : X \to \mathtt{Type}$, Coq also defines the variant $\{x : X \mathbin{\&} P\,x\}$ but we will not need this one.
[6] Operators on lists are parametric in X and this first argument is nearly always left implicit.
[7] as inductively defined in the `Permutation` module of the Coq standard library.
[8] Like lists based results, `finite` is parametric in X and the braces around it specify an *implicit* argument.

the homogeneous case, we simply write $\texttt{rel}_2\,X \coloneqq X \to X \to \mathbb{P}$, and $\texttt{rel}_1\,X \coloneqq X \to \mathbb{P}$ in the unary case. We use the letters $P, Q : \texttt{rel}_1\,\_$ to denote unary relations and $R, T : \texttt{rel}_2\,\_\,\_$ to denote binary relations. We write $P \subseteq Q$ or $R \subseteq T$ for the inclusion between relations. Except for commonly found notations like $\in$, $\sim_p$ or $\subseteq$, we generally write related pairs with a letter for the relation name, in prefix order, e.g. like in $T\,x\,y$. Hence, we refrain from using infix order or using symbols for naming relations. If we want to refer to the relation corresponding to an infix notation like e.g. membership, we may write e.g. $\cdot \in \cdot$. We mostly avoid infix notations because of the constraints in rendering notations in Coq scripts that would make them diverge too much from the paper rendering, especially when e.g. composing operators that are rendered as subscripts or superscripts.

For complex inductive predicates, we rather present the constructors using rules with a horizontal line separating the premises from the conclusion. As an example, we below display those of $\texttt{Forall}\,P : \texttt{rel}_1\,(\texttt{list}\,X)$ (denoted $\wedge_1 P$) and $\texttt{Forall2}\,R : \texttt{rel}_2\,(\texttt{list}\,X)\,(\texttt{list}\,Y)$ (denoted $\wedge_2 R$) which are finitary conjunctions defined in the `List` module of the standard library, for $P : \texttt{rel}_1\,X$ and $R : \texttt{rel}_2\,X\,Y$:

$$\frac{}{\wedge_1 P\ []} \qquad \frac{P\,x \qquad \wedge_1 P\ l}{\wedge_1 P\ (x :: l)} \qquad \frac{}{\wedge_2 R\ []\ []} \qquad \frac{R\,x\,y \qquad \wedge_2 R\ l\ m}{\wedge_2 R\ (x :: l)\ (y :: m)}$$

The free symbols $x, y : X$ and $l, m : \texttt{list}\,X$ can be instantiated by any value in their respective types. In the corresponding Coq constructors, they are universally quantified over.

3 Inductive covers

We recall the notion of inductive cover [4]. Our motivation for using covers is not topological but rather, such inductive covers conveniently subsume both accessibility and bar inductive predicates; see Sections 3.2 and 3.3. We discuss three characterizations of covers, the positive, the negative and the sequential, from the strongest to the weakest (constructively), but also explain in some details how to get their classical equivalence, separating the roles played by the axioms of excluded middle and dependent choice. We discuss these characterizations in the context Brouwer's intuitionistic understanding of infinite sequences.

But before we switch to covers, we import the standard notion of being upward closed to be encountered in order or lattice theory, however not requiring partial orders but any binary relation instead.

▶ **Definition 1** (Upward closed). *Given a type X and a binary relation $T : \texttt{rel}_2\,X$, we say that a unary relation $P : \texttt{rel}_1\,X$ is* T-upward closed *if P is stable under direct T-images. We define:* $\texttt{upclosed}\,T\,P \coloneqq \forall x\,y, T\,x\,y \to P\,x \to P\,y$.

For instance, the finitary conjunction $\wedge_1 P$ is upward closed for permutations, formally stated as $\texttt{upclosed}\,(\cdot \sim_p \cdot)\,\wedge_1 P$. Upward closed unary relations will be preserved by covers, and some results about covers (incl. the FAN theorem) assume upward closed relations.

3.1 Inductive covers definition, basic results

As in [4], we work with the particular class of singleton inductively generated formal topologies, as opposed to the more general (e.g. indexed) presentation of [7]. They are defined by the notion of inductive cover of a set (i.e. unary relation) along a (transition) binary relation.

▶ **Definition 2** (Inductive cover [4]). *Given a type X, a binary relation $T : \texttt{rel}_2\,X$ and a unary relation $P : \texttt{rel}_1\,X$, we define the* inductive T-cover of P, *denoted* $\texttt{cover}\;T\,P : \texttt{rel}_1\,X$ *by the two following inductive rules:*

$$\frac{P\,x}{\texttt{cover}\;T\,P\,x}\;[\texttt{cover\_stop}] \qquad \frac{\forall y,\, T\,x\,y \to \texttt{cover}\;T\,P\,y}{\texttt{cover}\;T\,P\,x}\;[\texttt{cover\_next}]$$

Notice that $P\,x$ (resp. $T\,x\,y$ and $\texttt{cover}\;T\,P\,x$) is denoted by $x \in P$ (resp. $y \in T(x)$ and $x \lhd P$) in [4] but we favor prefix notations to infix ones. Remark that the transition relation T is hidden in the infix notation $x \lhd P$ used for the cover whereas we keep it in $\texttt{cover}\,T\,P\,x$. Also in [4], the constructor `cover_stop` (resp. `cover_next`) is called reflexivity (resp. infinity).

The non-dependent induction principle (or eliminator, depending on your preferred terminology) generated for the $\texttt{cover}\;T\,P$ predicate has the following type:

$$\texttt{cover\_ind}\;T\,P:\;\forall Q,\, P \subseteq Q \to (\forall x,\, T\,x \subseteq Q \to Q\,x) \to (\forall x,\, \texttt{cover}\;T\,P\,x \to Q\,x).$$

Coq auto-generates a slight variant of it[9] but they are equivalent as non-dependent eliminators. We choose to present the above one because of its direct link with the positive, negative and sequential characterizations of the cover that we compare and analyze in upcoming Section 3.4. In our Coq code, we give a straightforward implementation of `cover_ind` as a guarded `Fixpoint`, very similar to the one auto-generated by Coq.

Using the `cover_ind` induction principle in combination with the constructors, we show how a morphism can be used to transfer covers between different types and relations.

▶ **Proposition 3** (`cover_morphism`). *Let X, Y be two types, $R : \texttt{rel}_2\,X$ and $T : \texttt{rel}_2\,Y$ be binary relations, and $P : \texttt{rel}_1\,X$ and $Q : \texttt{rel}_1\,Y$ be unary relations. We further assume a map $f : Y \to X$ which is supposed to be a morphism w.r.t. P/Q and R/T, i.e. satisfying*

$$\forall y,\, P(f\,y) \to Q\,y \qquad \textit{and} \qquad \forall y_1\,y_2, T\,y_1\,y_2 \to R\,(f\,y_1)\,(f\,y_2).$$

Then we have $\forall x\,y,\, x = f\,y \to \texttt{cover}\;R\,P\,x \to \texttt{cover}\;T\,Q\,y$.

Proof. We prove $\texttt{cover}\;R\,P\,x \to \forall y,\, x = f\,y \to \texttt{cover}\;T\,Q\,y$ by induction on $\texttt{cover}\;R\,P\,x$ using `cover_ind`. ◀

For the rest of the section, we assume a fixed type X to be used as carrier for binary relations $R, T : \texttt{rel}_2\,X$ and unary relations $P, Q : \texttt{rel}_1\,X$. The monotonicity of `cover` can be obtained as a particular case, using the identity morphism $f \coloneqq \lambda\,x,\,x$. More precisely, $\texttt{cover}\;(\cdot)\,(\cdot)$ is antitonic in its first argument and monotonic in its second argument.

$$\texttt{cover\_mono}\;R\;T\,P\,Q:\;T \subseteq R \to P \subseteq Q \to \texttt{cover}\;R\,P \subseteq \texttt{cover}\;T\,Q.$$

Additionally to be increasing (by `cover_stop`) and monotonic (by `cover_mono`), $\texttt{cover}\;T$ is also an idempotent operator making it a closure operator:

$$\texttt{cover\_idempotent}\;T\,P:\;\texttt{cover}\;T\,(\texttt{cover}\;T\,P) \subseteq \texttt{cover}\;T\,P.$$

Proof. Assuming an arbitrary x, the proof of $\texttt{cover}\,T\,(\texttt{cover}\,T\,P)\,x \to \texttt{cover}\,T\,P\,x$ proceeds by induction on $\texttt{cover}\;T\,(\texttt{cover}\;T\,P)\,x$. ◀

[9] namely $\forall Q,\, P \subseteq Q \to (\forall x,\, T\,x \subseteq \texttt{cover}\;T\,P \to T\,x \subseteq Q \to Q\,x) \to (\forall x,\, \texttt{cover}\;T\,P\,x \to Q\,x)$.

Then we get that the $\texttt{cover}\,T$ operator preserves T-upward closed unary relations:

$$\texttt{cover\_upclosed}\;T\;P:\;\texttt{upclosed}\,T\,P \to \texttt{upclosed}\,T\,(\texttt{cover}\,T\,P).$$

Proof. We assume $\texttt{upclosed}\,T\,P$ and an arbitrary x and show $\texttt{cover}\,T\,P\,x \to \forall y,\,T\,x\,y \to \texttt{cover}\,T\,P\,y$ by induction on $\texttt{cover}\,T\,P\,x$. ◀

3.2 Inductive cover and accessibility

In this section, we fix a type X to serve as carrier for relations below. We recall that the cover predicate is a generalization of the accessibility predicate, also called R-founded in [4].

▶ **Definition 4** (acc(essibility), R-founded). *Given a binary relation* $R : \texttt{rel}_2\ X$*, the* acc(essibility) predicate[10] *for* R *and the* R-founded predicate[11] *are defined inductively, each with one single rule:*

$$\frac{\forall y,\,R\,x\,y \to \texttt{acc}\ R\ y}{\texttt{acc}\ R\ x}\,[\texttt{acc\_intro}] \qquad \frac{\neg R\,x\,x \qquad \forall y,\,R\,x\,y \to \texttt{founded}\,R\,y}{\texttt{founded}\,R\,x}$$

A simple observation shows that the shape of the constructor acc_intro is the same as the second constructor cover_next of the cover predicate. Furthermore, the first constructor cover_stop can be neutralized by setting P as the empty relation $\emptyset \coloneqq \lambda\_,\,\bot$. Hence we immediately derive the equivalence:

▶ **Proposition 5.** *The* acc*(essibility) predicate is an instance of the* cover *predicate.*

$$\texttt{acc\_iff\_cover\_empty}\ R\ x:\ \texttt{acc}\ R\ x \leftrightarrow \texttt{cover}\ R\ \emptyset\ x.$$

Moreover, accessible elements are necessarily irreflexive. Indeed, we show $\forall x,\,\texttt{acc}\ R\,x \to R\,x\,x \to \bot$ by induction on $\texttt{acc}\ R\,x$. Hence it follows that the left premise $\neg R\,x\,x$ of the constructor of R-founded is superfluous:

▶ **Proposition 6.** *R-founded and accessibility define equivalent notions:*

$$\texttt{founded\_iff\_acc}\ R\ x:\ \texttt{founded}\,R\,x \leftrightarrow \texttt{acc}\ R\,x.$$

As a corollary we get $\texttt{founded}\,R\,x \leftrightarrow \texttt{cover}\ R\,\emptyset\,x$, a result already established in [4, Theorem 3.2] but, seemingly, the authors did not observe that the left premise ($\neg R\,x\,x$ i.e. irreflexivity) of the introduction rule for R-founded was superfluous.

3.3 Inductive cover and inductive bars

Let X be a carrier type for lists. We consider unary relations on the type $\texttt{list}\,X$ that we use to represent finite sequences. We show that inductive covers, in addition to generalizing accessibility predicates (see Section 3.2), also generalize inductive bar predicates [9, 6].

▶ **Definition 7** (Inductive bar). *Let* $P : \texttt{rel}_1\ (\texttt{list}\,X)$ *be a unary relation on lists. We define the inductive* $\texttt{bar}\,P : \texttt{rel}_1\ (\texttt{list}\,X)$ *unary relation with the two following inductive rules:*

$$\frac{P\,l}{\texttt{bar}\,P\,l}\,[\texttt{bar\_stop}] \qquad \frac{\forall x,\,\texttt{bar}\,P\,(x::l)}{\texttt{bar}\,P\,l}\,[\texttt{bar\_next}]$$

[10] The variant Acc as defined in the Coq standard library module Prelude, simply uses the reversed relation R^{-1} instead of R for acc. So we have $\texttt{Acc}\ R \simeq \texttt{acc}\ R^{-1}$ and $\texttt{Acc}\ R^{-1} \simeq \texttt{acc}\ R$.

[11] R-foundness is defined in [4, Definition 3.1].

Compared to [9, Definition 6], there are two slight differences. First our lists expand from the left, whereas often in the literature [9, 6, 23], finite sequences expand from the right. Hence rule `bar_next` would be written

$$\frac{\forall x,\ \texttt{bar}\,P\,(l +\!\!+ [x])}{\texttt{bar}\,P\,l}$$

with such a reversed convention. However, this difference can be viewed as just of matter of ordering the display of the arguments of the list constructor ::. Another more important difference compared to [9, Definition 6] or else [23], is the absence of the inductive rule

$$\frac{\texttt{bar}\,P\ l}{\texttt{bar}\,P\,(x :: l)}\ [\texttt{bar\_monotone}]$$

in Definition 7. We discard rule [`bar_monotone`] because it is admissible for monotone unary relations on finite sequences.

▶ **Definition 8** (Monotone unary relation). *A unary relation* $P : \texttt{rel}_1\ (\texttt{list}\,X)$ *is* monotone *if it satisfies* $\texttt{monotone}\,P \coloneqq \forall x\,l,\ P\,l \to P(x :: l)$*.*

The (discarded) [`bar_monotone`] rule/constructor would ensure that $\texttt{bar}\,P$ is a monotone predicate even when P is not monotone. However, as an instance of `cover_upclosed`, if P is monotone then so is $\texttt{bar}\,P$; see `bar_monotone` below. We observe that monotone unary relations are those which are upward closed under list extension:

▶ **Definition 9.** *The* $\texttt{extends} : \texttt{rel}_2\ (\texttt{list}X)$ *binary relation on lists is defined by the single inductive rule:* $\dfrac{}{\texttt{extends}\ l\ (x :: l)}$

Alternatively, we could have defined `extends` using $\texttt{extends}\ l\ m \leftrightarrow \exists x,\ m = x :: l$. With this notion, we get the equivalence

$$\texttt{upclosed\_extends\_iff\_monotone}\ P :\ \texttt{upclosed}\,\texttt{extends}\,P \leftrightarrow \texttt{monotone}\,P$$

as an immediate consequence but the specialization goes further:

▶ **Proposition 10** (`bar_iff_cover_extends`). *Given a unary relation* $P : \texttt{rel}_1\ (\texttt{list}\,X)$ *and a list* $l : \texttt{list}\,X$*, we have the equivalence* $\texttt{bar}\,P\ l \leftrightarrow \texttt{cover}\,\texttt{extends}\,P\ l$*.*

Thanks to Proposition 10 and `upclosed_extends_iff_monotone`, the two below results are specializations of respectively `cover_upclosed` and `cover_mono`.

$$\begin{array}{ll} \texttt{bar\_monotone}\ P : & \texttt{monotone}\,P \to \texttt{monotone}\,(\texttt{bar}\,P); \\ \texttt{bar\_mono}\ P\ Q : & P \subseteq Q \to \texttt{bar}\,P \subseteq \texttt{bar}\,Q. \end{array}$$

More generally, the analysis that we are going to present for inductive covers in the next section can be specialized to either accessibility predicates and inductive bar predicates.

3.4 Positive, negative and sequential characterizations

We now discuss other characterizations of covers, which are not constructively equivalent to the inductive one, but however are classically equivalent, hence the abusive use of the word "characterization." We present a detailed analysis of those characterizations and under which classical axioms their equivalence depends on.

The results of this section that assume classical axioms are not used elsewhere in this paper: these axioms are (propositional) excluded middle (XM), giving us De Morgan laws for logical connectives and quantifiers, and dependent choice (DC):

$$\texttt{xm} : \forall A : \mathbb{P},\ A \lor \neg A;$$
$$\texttt{dc} : \forall X\,(R : \texttt{rel}_2\ X),\ (\forall x \exists y,\ R\,x\,y) \to \forall x \exists \rho : \mathbb{N} \to X,\ \rho_0 = x \land \forall n,\ R\ \rho_n\ \rho_{1+n}.$$

The names of the results that depend on these added axioms are suffixed with `_XM` or `_DC` or both for an unambiguous exposition.

We start with the following definitions of the positive characterization `cover_pos`, the negative characterization `cover_neg` and the sequential characterization `cover_seq`.

▶ **Definition 11** (Nonequivalent characterizations of cover).

$$\texttt{cover\_pos}\ T\ P\ x \coloneqq \lambda\, Q : \texttt{rel}_1\ X,\ P \subseteq Q \to (\forall y,\ T\,y \subseteq Q \to Q\,y) \to Q\,x$$
$$\texttt{cover\_neg}\ T\ P\ x \coloneqq \lambda\, Q : \texttt{rel}_1\ X,\ Q\,x \to (\forall y,\ Q\,y \to \exists z,\ Q\,z \land T\,y\,z) \to \exists y, P\,y \land Q\,y$$
$$\texttt{cover\_seq}\ T\ P\ x \coloneqq \lambda\, \rho : \mathbb{N} \to X,\ \rho_0 = x \to (\forall n,\ T\,\rho_n\ \rho_{1+n}) \to \exists n,\ P\,\rho_n.$$

Although not equivalent, the constructive strength of these characterizations can be compared: they are displayed from the strongest (`cover_pos`) to the weakest (`cover_seq`). Beware that both Q and ρ are universally quantified over in the characterizations below.

The positive characterization `cover_pos` is really just a reordering of the implications in the induction principle `cover_ind`, so we get the following equivalence purely constructively:

$$\texttt{cover\_iff\_cover\_pos}\ T\ P\ x :\ \texttt{cover}\ T\,P\,x \leftrightarrow \forall Q,\ \texttt{cover\_pos}\ T\ P\ x\ Q.$$

The positive characterization is constructively stronger that the negative one:

$$\texttt{cover\_pos\_\_cover\_neg}\ T\ P\ x :\ (\forall Q,\ \texttt{cover\_pos}\ T\ P\ x\ Q) \to (\forall Q,\ \texttt{cover\_neg}\ T\ P\ x\ Q).$$

Proof. We use $\forall Q$, `cover_pos` $T\ P\ x\ Q$ as the formulation of an induction principle. ◀

The negative characterization is constructively stronger than the sequential one. The below proof argument anticipates the intuition behind the definition of the negative characterization.

$$\texttt{cover\_neg\_\_cover\_seq}\ T\ P\ x :\ (\forall Q,\ \texttt{cover\_neg}\ T\ P\ x\ Q) \to (\forall \rho,\ \texttt{cover\_seq}\ T\ P\ x\ \rho).$$

Proof. Assuming a T-sequence $\rho : \mathbb{N} \to X$, we instantiate Q with the direct image $\rho(\mathbb{N}) \coloneqq \lambda\, y,\ \exists n, \rho_n = y$. We show `cover_neg` $T\ P\ x\ \rho(\mathbb{N}) \to$ `cover_seq` $T\ P\ x\ \rho$ and conclude. ◀

We now explain the intuition behind those definitions by turning to a *classical interpretation* where all those characterizations are equivalent, discussing the precise roles played by XM and DC. The negative characterization `cover_neg` is central to our analysis and can be understood in two ways, either as deriving from `cover_pos` or generalizing `cover_seq`:

- The first understanding of `cover_neg` is as contrapositive form of `cover_pos`:

  $$\texttt{cover\_pos\_iff\_neg\_XM}\ T\ P\ x\ Q : \texttt{cover\_pos}\ T\ P\ x\ Q \leftrightarrow \texttt{cover\_neg}\ T\ P\ x\ (\neg Q).$$

  The proof involves excluded middle but *first-order De Morgan transformations* are enough to get the equivalence.[12] The converse implication of `cover_pos__cover_neg` above is unlikely to be constructively provable (see Section 3.5), but it is a direct corollary to `cover_pos_iff_cover_neg_XM`,[13] however assuming XM as an added axiom;

[12] In the Coq script, we insist on obtaining that equivalence via De Morgan rewriting and congruence only.
[13] see `cover_neg__cover_pos_XM`.

- The second way to understand the negative characterization `cover_neg` is to view it as a generalization of the sequential characterization `cover_seq`. Notice that the statement $\forall \rho$, `cover_seq` $T\,P\;x\;\rho$ is the usual intuitive definition of being a T-cover for P:

  Any infinite T-sequence starting at x meets P.[14]

  However this interpretation depends on *what are the inhabitants of the type* $\mathbb{N} \to X$ of which ρ is a member; see Section 3.5. In the proof of `cover_neg__cover_seq`, we used the direct image $\rho(\mathbb{N})$ as a particular instance of Q in `cover_neg`. Q represents a set of values containing x and over which T is a total binary relation, which generalizes T-sequences by removing the requirement of determinism. The quantification over T-sequences $\rho : \mathbb{N} \to X$ is replaced by quantification over Q which is an *T-unstoppable non-deterministic process*: indeed any point in Q has at least T-image in Q. This property of unstoppability $\forall y,\, Q\,y \to \exists z,\, Q\,z \wedge T\,y\,z$ is shared also by Brouwer's notion of spread.

As a consequence of the above discussion, constructively already, the positive characterization is equivalent to the inductive one, and stronger than the negative one, which is itself stronger than the sequential one. Hence we derive:

$$\texttt{cover\_negative}\; T\,P\;x : \;\; \texttt{cover}\; T\,P\;x \to \forall Q,\, Q\,x \to (\forall y,\, Q\,y \to \exists z,\, Q\,z \wedge T\,y\,z) \to \exists y, P\,y \wedge Q\,y$$
$$\texttt{cover\_sequences}\; T\,P\;x : \texttt{cover}\; T\,P\;x \to \forall \rho,\, \rho_0 = x \to (\forall n,\, T\,\rho_n\;\rho_{1+n}) \to \exists n,\, P\,\rho_n$$

If one is interested in the converse implications, then, on the one hand, XM would be used to prove that $\forall Q$, `cover_neg` $T\,P\;x\;Q$ implies `cover` $T\,P\;x$. On the other hand, to recover $\forall Q$, `cover_neg` $T\,P\;x\;Q$ from $\forall \rho$, `cover_seq` $T\,P\;x\;\rho$, one uses `DC` $\{x \mid Q\,x\}$ which is dependent choice specialized on the Σ-type $\{x \mid Q\,x\}$ where $Q : \texttt{rel}_1\; X$. Indeed, the statement of `DC` X, i.e. dependent choice specialized on type X is:

$$\texttt{DC}\; X \coloneqq \forall\, R : \texttt{rel}_2\; X,\, (\forall x \exists y,\, R\,x\,y) \to \forall x \exists \rho : \mathbb{N} \to X,\, \rho_0 = x \wedge \forall n,\, R\;\rho_n\;\rho_{1+n}.$$

When $Q : \texttt{rel}_1\; X$, we reformulate the instance `DC` $\{x \mid Q\,x\}$ as[15]

$$\forall R,\, (\forall x,\, Q\,x \to \exists y,\, Q\,y \wedge R\,x\,y) \to \forall x,\, Q\,x \to \exists \rho,\, \rho_0 = x \wedge \forall n,\, Q\,\rho_n \wedge R\;\rho_n\;\rho_{1+n}$$

which is exactly what is needed to extract a sequence $\rho : \mathbb{N} \to X$ out of the T-unstoppable process Q starting at x.

$$\texttt{cover\_seq\_\_cover\_neg\_DC}\; T\,P\;x : (\forall \rho,\, \texttt{cover\_seq}\; T\,P\;x\;\rho) \to (\forall Q,\, \texttt{cover\_neg}\; T\,P\;x\;Q).$$

▶ **Theorem 12** (in the spirit of Brouwer's bar theorem)**.** *Assuming* `xm` *and* `dc`*, the inductive and the sequential characterizations of covering are equivalent:*

$$\texttt{cover}\; T\,P\;x \;\;\leftrightarrow\;\; \forall \rho, \rho_0 = x \to (\forall n,\, T\,\rho_n\;\rho_{1+n}) \to \exists n,\, P\,\rho_n.$$

Hence under XM+DC, any cover is an inductive cover while Brouwer's "bar theorem" states that "any bar is inductive bar," or, quoting [6], for sequences of natural numbers:

$$\forall P : \texttt{rel}_1\, (\texttt{list}\, \mathbb{N}),\, \texttt{bar}\; P\; [\,] \;\leftrightarrow\; \forall \alpha : \mathbb{N} \to \mathbb{N},\, \exists n,\, P\,[\alpha_{n-1}; \ldots; \alpha_0]$$

The bar theorem statement is an instance of Theorem 12 where $T \coloneqq$ `extends`. Indeed, an `extends`-sequence of lists in $\mathbb{N} \to \texttt{list}\, X$ corresponds to the n-prefixes of a sequence $\mathbb{N} \to X$; see `Brouwer_bar_XM_DC` in the Coq code.

[14] Such formulation are more commonly found for the "intuitive" (read sequential) definition of "being a bar for P" [6]. See `bar_sequences` in Section 3.5 for the corresponding specialization.

[15] see `DC_sig__DC_Σ` in the Coq code.

3.5 Discussion

We have explained how the inductive predicates `bar` and `acc` are just specializations of the notion of inductive `cover` so the remarks below also apply to those restricted notions. For instance we get the following specializations for $R : \texttt{rel}_2\, X$ and $P : \texttt{rel}_1\, (\texttt{list}\, X)$:

$$\begin{array}{ll}\texttt{acc\_negative}\ x: & \texttt{acc}\ R\ x \to \forall Q,\ Q\,x \to (\forall y,\ Q\,y \to \exists z,\ Q\,z \wedge R\,y\,z) \to \bot;\\ \texttt{acc\_sequences}\ x: & \texttt{acc}\ R\ x \to \forall \rho,\ \rho_0 = x \to (\forall n,\ R\ \rho_n\ \rho_{1+n}) \to \bot;\\ \texttt{bar\_negative}: & \texttt{bar}\ P\ [] \to \forall Q,\ Q\,[] \to \big(\forall l,\ Q\,l \to \exists x,\ Q\,(x :: l)\big) \to \exists l, P\,l \wedge Q\,l;\\ \texttt{bar\_sequences}: & \texttt{bar}\ P\ [] \to \forall \alpha : \mathbb{N} \to X, \exists n,\ P\,[\alpha_{n-1}; \ldots; \alpha_0].\end{array}$$

The negative characterization is intermediate between the inductive/positive characterization (strongest) and the sequential characterization (weakest). We isolate the role played by XM (in fact rewriting using De Morgan laws) and DC. While it avoids DC, the negative characterization, using a notion of unstoppable non-deterministic process instead of the notion of sequence, still likely requires XM to be equivalent with the positive characterization. Indeed, were the negative characterization be constructively equivalent to positive/inductive characterization, such a result would instantly give us Theorem 12 (and Brouwer's bar theorem) using DC alone, hence avoiding XM.

The discussion on what is nature of (infinite) sequences is central to the sequential characterization of bars, and of course, as the infinite itself, is very much debated in constructive mathematics. Clearly, adjoining XM and DC populates the type $\mathbb{N} \to X$ with enough *lawless sequences*. Brouwer however rejected XM and DC and instead justifies his bar theorem using "Brouwer's thesis" [23] which is not as strong as an axiom as XM+DC. In [6], Coquand criticizes the use of the type $\mathbb{N} \to X$ to cover "all" sequences in the sequential characterization of bars:

> "This example is paradigmatic: by replacing systematically the intuitive notion of bar by the notion of inductive bar, we can now prove Brouwer's fan theorem. More generally, we can think of `bar` P $[]$ as the *correct format expression of a universal quantification over all sequences*, not necessarily given by a law." (emphasis added)

To be more specific, absent of extra axioms, the type $\mathbb{N} \to X$ of *lawlike sequence* (on which the sequential characterization is based) cannot account for sequences that do not evolve according to a predetermined law, see e.g. Veldman [23]:

> "the intuitionistic mathematician [...] admits the possibility of sequences $\alpha_0, \alpha_1, \alpha_2, \ldots$ that are created step-by-step and thus, in some sense, are given by a black box. He is very much aware that *he is unable to make any kind of survey of the totality of all infinite sequences* of natural numbers." (emphasis added)

In a way, we follow and extend to covers the program proposed by Coquand [6] to systematically replace the intuitive (understand sequential) notion of cover by the inductive version, avoiding axioms altogether. But we can still use the sequential or negative versions, in a limited way, at the end of a constructive deduction, e.g. following the FAN theorem.

4 The FAN theorem for inductive covers

In this section, we present another interpretation of the FAN theorem in type theory, generalizing the FAN theorem for inductive bars [9] to inductive covers [7, 4] instead. We give a concise proof for this result, which differs significantly from that of [9, Theorem 6]. Hence, as an specialization, we get an alternate proof of that former result as well.

In this section, let us fix a type X and a binary relation $T : \texttt{rel}_2\, X$. We extend the binary relation T to lists (viewed as finite sets), as $T^\dagger : \texttt{rel}_2\,(\texttt{list}\, X)$ using the direct image and this way, we can view FANs over T as $T^\dagger$-sequences over finite sets.

4.1 Lifting a relation to finite sets

We define the finitary image relation on $\texttt{list}\, X$ viewed as finite sets, i.e. permutations and contractions are admissible for lists used in that context.

▶ **Definition 13** (Finitary image). *We define the* finitary image *binary relation on lists, denoted* $T^\dagger : \texttt{rel}_2\,(\texttt{list}\, X)$*, by* $T^\dagger \coloneqq \lambda\, l\, m,\, \forall y,\, y \in m \to \exists x,\, x \in l \wedge T\, x\, y$*, i.e.* $T^\dagger\, l\, m$ *holds when* m *is included in the direct image of* l*.*

The finitary image relation $T^\dagger$ is increasing w.r.t. its first argument and decreasing w.r.t. the second, i.e. $l_1 \subseteq l_2 \to m_2 \subseteq m_1 \to T^\dagger\ l_1\ m_1 \to T^\dagger\ l_2\ m_2$ holds.

One critical observation for the proof of the FAN theorem below is how $T^\dagger$ behaves when splitting its first/source argument in two halves. Then there is a corresponding splitting of the second/image argument, but since $T^\dagger$ ignores the order on the elements of lists, this splitting only holds up to a permutation of the image list:

$$\texttt{fimage\_split\_inv}\ l_1\ l_2\ m :\ T^\dagger\ (l_1 \mathbin{+\!\!+} l_2)\ m \to \exists\, m_1\, m_2,\ \wedge \begin{cases} m \sim_p m_1 \mathbin{+\!\!+} m_2 \\ T^\dagger\ l_1\ m_1 \\ T^\dagger\ l_2\ m_2. \end{cases}$$

Proof. We proceed by induction on m. ◀

Additionally, we show that $T^\dagger\ (\cdot)\ k$ is upward closed for permutations for any k, which can be written as $\texttt{upclosed}\,(\cdot \sim_p \cdot)\,(T^\dagger\ \cdot\ k)$. And to conclude this section, if P is upward closed for T then the finitary conjunction $\wedge_1 P$ of P (over lists) is upward closed for $T^\dagger$, i.e. $\texttt{upclosed}\, T\, P \to \texttt{upclosed}\, T^\dagger\, \wedge_1 P$.

4.2 Proof of the FAN theorem for inductive covers

We give a proof of the statement of the FAN theorem for inductive covers, using the finitary image relation $T^\dagger$ to represent FANs over the relation T.

▶ **Theorem 14** (FAN for inductive covers). *Assume* $P : \texttt{rel}_1\ X$ *is unary relation upward closed for* T*. If* x *is in the* T*-cover of* P *then the singleton list* $[x]$ *is in the* $T^\dagger$*-cover of* $\wedge_1 P$*, i.e.*

$$\texttt{FAN\_cover} :\ \texttt{upclosed}\, T\, P \to \forall x,\, \texttt{cover}\, T\, P\ x \to \texttt{cover}\, T^\dagger\ \wedge_1 P\ [x].$$

Using a sequential understanding of covers, the statement could be read as: if any T-sequence starting at x meets P then any $T^\dagger$-sequence starting at $[x]$ meets $\wedge_1 P$, hence "any finitary FAN rooted at x meets a monotone P uniformly," which is a commonly found informal statement of the FAN theorem.

While this sequential understanding cannot be established in our constructive framework (for reasons discussed in Section 3.5), we below give a quite compact inductive proof of the positive/inductive understanding of the statement of the FAN theorem for inductive covers.

Proof. Let us assume P with $\texttt{upclosed}\, T\, P$. We first show that $\texttt{cover}\, T^\dagger\ \wedge_1 P$ is upward closed for permutations, stated as $\texttt{upclosed}\,(\cdot \sim_p \cdot)\,(\texttt{cover}\, T^\dagger\ \wedge_1 P)$. For this, we prove $\texttt{cover}\, T^\dagger\ \wedge_1 P\ l \to \forall m,\, l \sim_p m \to \texttt{cover}\, T^\dagger\ \wedge_1 P\ m$ by induction on $\texttt{cover}\, T^\dagger\ \wedge_1 P\ l$.

Now, we establish the *key result* that $\texttt{cover}\,T^\dagger\ \wedge_1 P$ is stable under (binary) union, herein represented by the append operation on lists:

$$\texttt{cover\_fimage\_union}\ l\ m:\ \texttt{cover}\,T^\dagger\ \wedge_1 P\ l \to \texttt{cover}\,T^\dagger\ \wedge_1 P\ m \to \texttt{cover}\,T^\dagger\ \wedge_1 P\ (l \mathbin{++} m).$$

The proof proceeds by *nested induction*, first on $\texttt{cover}\,T^\dagger\ \wedge_1 P\ l$ and then on $\texttt{cover}\,T^\dagger\ \wedge_1 P\ m$, with a critical use of `fimage_split_inv` to invert two statements of shape $T^\dagger\ (\cdot \mathbin{++} \cdot)\ (\cdot)$ where the first argument of $T^\dagger$ is a union of lists. As a corollary of `cover_fimage_union`, we get the specialization where $l \coloneqq [x]$ is a singleton as

$$\forall\, x\, m,\ \texttt{cover}\ T^\dagger\ \wedge_1 P\ [x] \to \texttt{cover}\ T^\dagger\ \wedge_1 P\ m \to \texttt{cover}\ T^\dagger\ \wedge_1 P\ (x :: m)$$

and then, as a direct consequence

$$\texttt{cover\_fimage\_Forall}\ l:\ \big(\forall x,\ x \in l \to \texttt{cover}\ T^\dagger\ \wedge_1 P\ [x]\big) \to \texttt{cover}\ T^\dagger\ \wedge_1 P\ l$$

for which we proceed by induction on l.

We can conclude with the proof of the FAN theorem for inductive covers. We establish $\texttt{cover}\ T^\dagger\ \wedge_1 P\ [x]$, reasoning by induction on $\texttt{cover}\ T\ P\ x$:

- the base case where $P\,x$ holds is trivially solved by giving a proof of $\wedge_1 P\ [x]$ and then deriving $\texttt{cover}\ T^\dagger\ \wedge_1 P\ [x]$ with an instance of first constructor `cover_stop`;
- in the recursive case where $\forall y,\ T\,x\,y \to \texttt{cover}\ T^\dagger\ \wedge_1 P\ [y]$ is the induction hypothesis, we show $\forall l,\ T^\dagger\ [x]\ l \to \forall y,\ y \in l \to \texttt{cover}\ T^\dagger\ \wedge_1 P\ [y]$ and then combine `cover_fimage_Forall` and an instance of the second constructor `cover_next`.

This concludes our proof of the FAN theorem for inductive covers. ◀

We can immediately derive $\wedge_1(\texttt{cover}\ T\ P)\ l \to \texttt{cover}\ T^\dagger\ \wedge_1 P\ l$ by induction on l and then the following characterization of covering for the finitary image:

$$\texttt{cover\_fimage\_iff}:\ \texttt{upclosed}\,T\ P \to \forall l,\ \texttt{cover}\,T^\dagger\ \wedge_1 P\ l \leftrightarrow (\forall x,\ x \in l \to \texttt{cover}\,T\ P\ x).$$

i.e. the list l is $T^\dagger$-covered for $\wedge_1 P$ if and only if all the member of l are T-covered for P.

4.3 The FAN theorem for inductive bars

We recall the interpretation of the FAN theorem in type theory [9] and derive an alternate proof of that result as an instance of Theorem 14, which illustrates our claim of generalization. We fix a carrier type X for lists and consider relations over $\texttt{list}\,X$ and $\texttt{list}\,(\texttt{list}\,X)$. For $lc : \texttt{list}\,(\texttt{list}\,X)$, let us first define the

$$\texttt{FAN}\ lc \coloneqq \lambda\, l,\ \wedge_2(\cdot \in \cdot)\ l\ lc$$

i.e. if written as $l = [x_1; \ldots; x_n]$ and $lc = [c_1; \ldots; c_p]$, $\texttt{FAN}\ lc\ l$ means $n = p$ and $x_1 \in c_1, x_2 \in c_2, \ldots, x_n \in c_n$. Stated in plain english, l is a list of one-to-one choices for the choice list lc; see the inductive definition of $\wedge_2 R$ in Section 2. Using generic tools designed for the `finite` abstraction, we can show that $\texttt{FAN}\ lc$ is a finite, i.e.

$$\texttt{FAN\_finite}\ lc:\ \texttt{finite}\,(\texttt{FAN}\ lc).$$

However in [9, page 102], the author gives a *specific* construction of a list which collects the lists of choices l s.t. $\texttt{FAN}\ lc\ l$, that we denote $\texttt{list\_fan}\ lc$ herein, satisfying:

$$\texttt{list\_fan\_spec}\ lc:\ \forall l,\ \texttt{FAN}\ lc\ l \leftrightarrow l \in \texttt{list\_fan}\ lc.$$

Thus the dependent pair ($\texttt{list\_fan}\ lc, \texttt{list\_fan\_spec}\ lc$) is an (explicitly given) inhabitant of the type $\texttt{FAN\_finite}\ lc$. The value of $\texttt{list\_fan}\ lc$ can be viewed as a generalization of the exponential function to lists, computing the *list of choice sequences for lc*.

The FAN theorem as stated and proved in [9] relies on the particular implementation of the exponential `list_fan` given there, but the result itself only depends on the fact that `list_fan` satisfies `list_fan_spec`. Theorem 6 of [9] also assumes the added rule [`bar_monotone`] in the inductive definition of the `bar` predicate but it is admissible for monotone relations.

▶ **Theorem 15** (reminder of Theorem 6 of [9]). *Let* $P : \texttt{rel}_1\,(\texttt{list}\,X)$ *be unary relation. The following statement holds:* $\texttt{monotone}\,P \to \texttt{bar}\ P\ [] \to \texttt{bar}\ \big(\lambda\,lc,\ \wedge_1 P\,(\texttt{list\_fan}\ lc)\big)\ []$.

Proof. We first reformulate the result as

$$\texttt{FAN\_bar}\ P:\ \texttt{monotone}\,P \to \texttt{bar}\ P\ [] \to \texttt{bar}\ (\lambda\,lc,\ \texttt{FAN}\ lc \subseteq P)\ []$$

which is an equivalent statement thanks to the monotonicity `bar_mono` of the `bar` predicate. Indeed, using `list_fan_spec`, we get the equivalence $\wedge_1 P\,(\texttt{list\_fan}\ lc) \leftrightarrow \texttt{FAN}\ lc \subseteq P$ for any lc. But the statement $\texttt{FAN\_bar}\,P$ is independent of the implementation of `list_fan`.

Using the results of Section 3.3, we replace the hypotheses $\texttt{monotone}\,P$ and $\texttt{bar}\ P\ []$ by $\texttt{upclosed}\,\texttt{extends}\,P$ and $\texttt{cover}\ \texttt{extends}\ P\ []$, and the goal $\texttt{bar}\ (\lambda\,lc,\ \texttt{FAN}\ lc \subseteq P)\ []$ becomes $\texttt{cover}\ \texttt{extends}\ (\lambda\,lc,\ \texttt{FAN}\ lc \subseteq P)\ []$. Hence, by Theorem 14 we get $\texttt{cover}\,\texttt{extends}^\dagger\ \wedge_1 P\ [[]]$. Then we transfer the inductive cover using `list_fan` as a morphism, see Proposition 3:

$$\texttt{cover}\,\texttt{extends}^\dagger\ \wedge_1 P\ [[]] \to \texttt{cover}\ \texttt{extends}\ (\lambda\,lc,\ \texttt{FAN}\ lc \subseteq P)\ []$$

after having checked that both $\texttt{extends}\ l\ m \to \texttt{extends}^\dagger\ (\texttt{list\_fan}\,l)\ (\texttt{list\_fan}\,m)$, $\wedge_1 P\ (\texttt{list\_fan}\ lc) \to \texttt{FAN}\ lc \subseteq P$, and $[[]] = \texttt{list\_fan}\ []$ hold.[16] ◀

The above result, and its proof, even though it uses one particular implementation of `list_fan` both in the proved statement and inside the arguments, can be adapted to work for any implementation of `list_fan` as soon as it satisfies `list_fan_spec`. The reason is that we pass through `FAN_bar` which is independent of the actual implementation of `list_fan`. This is how the proof is actually implemented in the Coq code.

Besides the previous remark and the detour via inductive covers, the proof we give differs from that of [9] in an important way. Indeed, the core argument of the later proof is the closure of monotone inductive `bar`s under binary intersection [9, Proposition 3]:

$$\texttt{monotone}\,P \to \texttt{monotone}\,Q \to \texttt{bar}\ P\ l \to \texttt{bar}\ Q\ l \to \texttt{bar}\ (P \cap Q)\ l$$

which is there established by nested inductions on $\texttt{bar}\ P\ l$, and then on $\texttt{bar}\ Q\ l$. On the contrary, the core argument in the proof of Theorem 14 lies in `cover_fimage_union`, i.e. the closure of $\texttt{cover}\ T^\dagger\ \wedge_1 P$ under binary union (the append operator on lists). In a way, it generalizes to upward closed inductive covers the stability under binary unions of finiteness.

5 Weaker variants of Kőnig's lemma

Recall the contrapositive form of Kőnig's lemma: any finitely branching tree without infinite branches is finite. We introduce (inductive) rose trees, i.e. finite trees but with arbitrary (but finite) branching at each node.

[16] Notice that to obtain $[[]] = \texttt{list\_fan}\ []$, we use an implementation of `list_fan` which satisfies this property, in addition to the specification `list_fan_spec`.

We give a type theoretic variant which has stronger assumptions (e.g. the covering assumption below), and which replaces the notion of possibly infinite tree that is implicit in formulation "any ... tree without infinite branches" with that of a relation:

> Assume a finitely branching relation $T : \texttt{rel}_2\, X$ and $P : \texttt{rel}_1\, X$ which is T-upward closed. If x belongs to the T-cover of P then the finite paths along T starting at x and avoiding P are the branches of a rose tree rooted at x.

Notice that we use equivalence between paths and branches to express that (part of) a relation is "the same" as a rose tree. Because we only view the relation via its paths, the acyclicity assumption, as used when (infinite) trees are viewed as graphs, can be dropped. But before we formalize this statement, we must define paths, rose trees and their branches.

5.1 Path, rose trees and their branches

Let us fix a type X as carrier for relations and indices of rose trees below.

▶ **Definition 16** (Inductive path). *For a relation $T : \texttt{rel}_2\, X$, the paths in T are described by a ternary relation $\texttt{path}\ T : X \to \texttt{list}\, X \to X \to \mathbb{P}$ defined by two inductive rules:*

$$\frac{}{\texttt{path}\ T\ x\ []\ x} \qquad \frac{T\ x\ y \qquad \texttt{path}\ T\ y\ p\ z}{\texttt{path}\ T\ x\ (y :: p)\ z}$$

Intuitively, $\texttt{path}\ T\, x\, p\, y$ means that p is the sequence of values encountered on a path from x to y, following the relation T, including the endpoint y but excluding starting point x. The existence of a T-path from x to y is equivalent to the reflexive and transitive closure of T (we do not use this characterization however), and hence we have:

$$\texttt{upclosed\_path}\ T\, P : \texttt{upclosed}\, T\, P \to \texttt{upclosed}\, (\lambda\, x\, y,\ \exists p,\ \texttt{path}\ T\ x\ p\ y)\, P.$$

▶ **Definition 17** (Inductive rose tree). *The type of X-indexed* rose trees *denoted* $\texttt{tree}\, X : \texttt{Type}$ *is inductively defined by a single rule:*

$$\frac{x : X \qquad l : \texttt{list}\,(\texttt{tree}\, X)}{\langle x|l\rangle : \texttt{tree}\, X}\ [\texttt{node}]$$

where we denote $\langle x|l\rangle$ as a shortcut for $(\texttt{node}\ x\ l)$*. The* root *of $t = \langle x|l\rangle$ is indexed by x and we write $\texttt{root}\, t = x$, and l is the* list of the sons *of t. We define the* height *of a rose tree, denoted $\texttt{tree\_ht} : \texttt{tree}\, X \to \mathbb{N}$, using the fixpoint equation $\texttt{tree\_ht}\, \langle x|[t_1; \ldots ; t_n]\rangle = 1 + \texttt{list\_max}\, [\texttt{tree\_ht}\, t_1; \ldots ; \texttt{tree\_ht}\, t_n]$.*

The branches *of a rose tree (the paths starting at the root) are characterized using a ternary relation $\texttt{branch} : \texttt{tree}\, X \to \texttt{list}\, X \to X \to \mathbb{P}$ inductively defined by two rules:*

$$\frac{}{\texttt{branch}\ \langle x|l\rangle\ []\ x} \qquad \frac{\langle y|m\rangle \in l \qquad \texttt{branch}\ \langle y|m\rangle\ p\ z}{\texttt{branch}\ \langle x|l\rangle\ (y :: p)\ z}$$

Hence a branch is either empty, stopping at the root, or the choice of a son (i.e. sub-tree) and of a branch in that son. The predicate $\texttt{branch}\ t\ p\ y$ relates a tree t, a list of visited indices p up to the index y of the root of a sub-tree of t.

5.2 Representing binary relations using rose trees

We give a formal definition for the statement "a binary relation is a finite tree." This is required indeed because the type of binary relations and the type of rose trees are very different. We use paths in relations and branches in rose trees as a mean to define the notion of *representation* by a rose tree, for the part of a relation $T : \mathtt{rel}_2\, X$ rooted at x of which the paths from x satisfy the property $P : \mathtt{list}\, X \to X \to \mathbb{P}$.

▶ **Definition 18** (Representation). *Assume a binary relation $T : \mathtt{rel}_2\, X$, a property for paths $P : \mathtt{list}\, X \to X \to \mathbb{P}$ and a point $x : X$. We say that* P in T at x is strongly represented by $t : \mathtt{tree}\, X$ *and write* $\mathtt{strongly\_represents}\; T\, P\, x\, t$ *if:*

$$\mathtt{strongly\_represents}\; T\, P\, x\, t \coloneqq \mathtt{root}\, t = x \wedge \forall\, p\, y,\, \mathtt{branch}\; t\; p\; y \leftrightarrow (\mathtt{path}\; T\; x\; p\; y \wedge P\; p\; y).$$

We say that P in T at $x : X$ is represented by $t : \mathtt{tree}\, X$ *and write* $\mathtt{represents}\; T\, P\, x\, t$ *if:*

$$\mathtt{represents}\; T\, P\, x\, t \coloneqq \mathtt{root}\, t = x \wedge \forall\, p\, y,\, P\, p\, y \to (\mathtt{branch}\; t\; p\; y \leftrightarrow \mathtt{path}\; T\; x\; p\; y).$$

The property P for paths is applied only to those originating at x but can depend on the destination as well as on the sequence of visited nodes on the path to the destination.

We observe that $\mathtt{strongly\_represents}\; T\, P\, x\, t \to \mathtt{represents}\; T\, P\, x\, t$. While the strong notion would be a first/natural choice to formalize the idea that the relation T starting at x and restricted by P "is a tree," this choice can however be questioned in the light of decidability issues. Indeed, when X is equipped with a (propositionally) decidable equality,[17] e.g. when $X = \mathbb{N}$, then both $\mathtt{branch}\; t\; p\; y$ and $\mathtt{path}\; T\; x\; p\; y$ become decidable predicates. In that case, $\mathtt{strongly\_represents}\; T\, P\, x\, t$ implies that P is decidable as well, an assumption we want to avoid for building representations. In the case of $\mathtt{represents}$, P does not need to be decidable but the representing tree may contain branches which do not satisfy P.[18]

We assume a fixed $T : \mathtt{rel}_2 X$ which is furthermore *finitely branching*, i.e. $\forall x,\, \mathtt{finite}\, (T\, x)$. We show that paths of bounded length can be strongly represented.

▶ **Theorem 19.** *When $T : \mathtt{rel}_2\, X$ is finitely branching, for any $n : \mathbb{N}$ and any $x : X$, the property $(\lambda\, p\, y,\, \lfloor p \rfloor \le n)$ in T at x has a strong representation.*

Proof. We build the tree t s.t. $\mathtt{strongly\_represents}\; T\; (\lambda\, p\, y,\, \lfloor p \rfloor \le n)\; x\; t$ by induction on n, after generalizing on x. ◀

Now we characterize the properties of paths that have representations as those which hold only for small paths.

▶ **Theorem 20.** *When $T : \mathtt{rel}_2 X$ is finitely branching, for any property $P : \mathtt{list}\, X \to X \to \mathbb{P}$ and any point $x : X$, the two following properties are equivalent:*

- $\exists t,\, \mathtt{represents}\; T\, P\, x\, t$;
- $\exists n, \forall\, p\, y,\, \mathtt{path}\; T\; x\; p\; y \to P\, p\, y \to \lfloor p \rfloor < n$.

Proof. In the forward direction, the bound n can be chosen to be the height $\mathtt{tree\_ht}\, t$ of the representation of P in T at x. In the reverse direction, given a bound n for the length of paths satisfying P, we first obtain a tree t s.t. $\mathtt{strongly\_represents}\, T\, (\lambda\, p\, y,\, \lfloor p \rfloor \le n)\, x\, t$. We then check that this tree t represents P in T at x. ◀

---

[17] i.e. $\forall\, x\, y : X,\, x = y \vee x \neq y$.

[18] Pruning them out to achieve strong representation should be possible if one further assumes that P is decidable and monotone, but we will not elaborate further herein.

Theorem 20 states that, in the finitely branching case, a relation is represented by a (finite) rose tree if and only if there is a global bound on the length of its paths. It can be compared to the characterization of binary trees[19] which are finite as those for which there is a uniform bound on the length of their branches, see e.g. [13].

5.3 Kőnig's lemma for inductive covers, accessibility and inductive bars

We prove statements of weakened variants of Kőnig's lemma assuming the existence of a cover for the root of the "tree," or its accessibility, or that P is inductively barred.

▶ **Theorem 21** (Kőnig's lemma for inductive covers)**.** *Let us assume a finitely branching binary relation $T : \texttt{rel}_2\, X$, i.e. $\forall x, \texttt{finite}\,(T\,x)$, a T-upward closed unary relation $P : \texttt{rel}_1\, X$, a root $x : X$ which is T-covered by P. Then the paths which refute P at their tail are represented in T at x, i.e. $\exists t, \texttt{represents}\ T\ (\lambda\, p\, y, \neg P\, y)\ x\ t$.*

Proof. Using the length of paths, we define the circle (centered at x) of radius n as

$$\texttt{circle}\,n \coloneqq \lambda\, y, \exists p, \texttt{path}\ T\ x\ p\ y \wedge n = \lfloor p \rfloor.$$

We show that circles are finite by induction on n and we get $\forall n, \texttt{finite}\,(\texttt{circle}\,n)$. This of course relies on T having finite direct images.

Let us define $Q : \texttt{rel}_1\,(\texttt{list}\,X)$ collecting the lists which are the support of some circle:

$$Q\,l \coloneqq \exists n, \forall x, \texttt{circle}\,n\,x \leftrightarrow x \in l.$$

We prove that Q meets $\wedge_1 P$. Indeed, as we assume $\texttt{cover}\,T\,P\,x$, using the FAN Theorem 14 for covers we get $\texttt{cover}\,T^\dagger \wedge_1 P\ [x]$. Then we use `cover_negative` with Q. We only need to show that Q holds at $[x]$ and is $T^\dagger$-unstoppable i.e. $\forall l, Q\,l \to \exists m, Q\,m \wedge T^\dagger\,l\,m$:

- $Q\,[x]$ holds because $[x]$ is a support for the circle of radius 0;
- Q is $T^\dagger$-unstoppable because the circle of radius $1+n$ is a $T^\dagger$-image of that of radius n.

As Q meets $\wedge_1 P$, then P includes some circle, i.e. there is n such that $\texttt{circle}\,n \subseteq P$. As a consequence, since T-paths from x of length greater that n cross $\texttt{circle}\,n$ hence meet P at that crossing point, their tail must belong to P as well, because P is T-upward closed. Hence $\forall\, p\, y, \texttt{path}\ T\ x\ p\ y \to n \le \lfloor p \rfloor \to P\, p\, y$ holds and we conclude using Theorem 20. ◀

This proof uses the FAN Theorem 14 for inductive covers, and then combines it with the `cover_negative` characterization. The finiteness of circles $\forall n, \texttt{finite}\,(\texttt{circle}\,n)$, which lives $\mathbb{P}$ (and not in `Type`), is not strong enough to be able to define `circle` as a map $\mathbb{N} \to \texttt{list}\,X$, which would be needed if the `cover_sequences` characterization were to be used instead of `cover_negative`.[20]

From Theorem 21, we can recover the finitary form of Kőnig's lemma similar to the one of [1]. A direct proof by induction on (the proof of) $\texttt{acc}\,T\,x$ would probably be shorter but we here illustrate the generality of Kőnig's lemma for inductive covers.

▶ **Corollary 22** (Kőnig's lemma for accessibility [1])**.** *Let $T : \texttt{rel}_2\,X$ be a binary relation s.t. $\forall x, \texttt{finite}\,(T\,x)$ and let $x : X$ be a T-accessible point of X, i.e. $\texttt{acc}\,T\,x$. Then there is a rose tree $t : \texttt{tree}\,X$ with root x such that the T-paths from x are exactly the branches of t.*

---

[19] as sets of finite sequences of Booleans representing their finite branches.

[20] Strong finiteness $\forall n, \{l \mid \forall x, \texttt{circle}\,n\,x \leftrightarrow x \in l\}$ instead of weak finiteness $\forall n \exists l\, \forall x, \texttt{circle}\,n\,x \leftrightarrow x \in l$ would overcome this issue but thanks to `cover_negative`, this stronger assumption is not required.

Proof. This is simply an instance of Theorem 21 where $P \coloneqq \emptyset$ is the empty unary relation. Notice that the empty property is decidable, hence we get a strong representation here. ◀

We present a variant of Kőnig's lemma for inductive bars. It is not exactly an instance of Theorem 21 because the properties of paths are not limited to those of the endpoint.

▶ **Theorem 23** (Kőnig's lemma for inductive bars). *Let us assume a finitely branching binary relation $T : \texttt{rel}_2\,X$, a monotone unary relation $P : \texttt{rel}_1\,(\texttt{list}\,X)$, and a point $x : X$. If* $\texttt{bar}\ P\ []$ *then* $\exists t$, $\texttt{represents}\ T\ \big(\lambda\, p\, y,\ \neg P\,(\texttt{rev}\, p)\big)\ x\ t$.

Proof. The proof is comparable (not identical) to the proof of Theorem 21 and uses `FAN_bar` and `bar_negative` instead as replacements for `FAN_cover` and `cover_negative`. ◀

5.4 Kőnig's lemma for sequences of finite choices

Bar predicates can be specialized using the notion of *good sequence*, i.e. one containing a redundant pair w.r.t. a binary (redundancy) relation. This relation can be the identity, but there are other interesting cases, e.g. multiset inclusion [17]. In this case, bar predicates characterize *inductive almost full relations* [25, 17].

We assume a binary relation $R : \texttt{rel}_2\,X$ to represent a notion of redundancy, and define two unary relation $\texttt{good}\,R$ and $\texttt{irred}\,R$ of type $\texttt{rel}_1\,(\texttt{list}\,X)$, $\texttt{good}\,R$ characterizing lists which contain a good pair, and $\texttt{irred}\,R$ characterizing lists which are irredundant, i.e. avoiding good pairs:[21]

$$\begin{aligned}\texttt{good}\,R\,p\ &\coloneqq \exists\, l\,x\,m\,y\,r,\ p = l \mathbin{+\!\!+} x :: m \mathbin{+\!\!+} y :: r \wedge R\,y\,x;\\ \texttt{irred}\,R\ p &\coloneqq \forall\, l\,x\,m\,y\,r,\ p = l \mathbin{+\!\!+} x :: m \mathbin{+\!\!+} y :: r \to R\,x\,y \to \bot.\end{aligned}$$

It is obvious that $\texttt{good}\,R$ is a monotone. Moreover, we show the correspondence between bad (i.e. not good) lists and irredundant ones:[22]

$$\texttt{not\_good\_iff\_irred}\ R\ p:\ \neg\big(\texttt{good}\,R\ (\texttt{rev}\,p)\big) \leftrightarrow \texttt{irred}\,R\ p.$$

▶ **Definition 24** (Almost full relation [25]). *For binary relations $R : \texttt{rel}_2\,X$, we define the predicate $\texttt{af}\ R : \mathbb{P}$ using the two inductive rules, where $R{\uparrow}u \coloneqq \lambda\, x\, y,\ R\,x\,y \vee R\,u\,x$:*

$$\frac{\forall x\, y,\ R\,x\,y}{\texttt{af}\ R} \qquad \frac{\forall u,\ \texttt{af}\ R{\uparrow}u}{\texttt{af}\ R}\ .$$

We recall that $\texttt{af}\ R$ is another way of stating (i.e. is equivalent to) $\texttt{bar}\ (\texttt{good}\,R)\ []$ (see e.g. [17, p. 11] or [18]) but below we just need the implication in this direction:

$$\texttt{af\_\_bar\_good\_nil}\ R:\ \texttt{af}\ R \to \texttt{bar}\ (\texttt{good}\,R)\ [].$$

Proof. First we establish $\texttt{bar}\ (\texttt{good}\,R{\uparrow}u)\ p \to \texttt{bar}\ (\texttt{good}\,R)\ (p \mathbin{+\!\!+} [u])$ by induction on $\texttt{bar}\ (\texttt{good}\,R{\uparrow}u)\ p$. As an instance where $p \coloneqq []$, we get $\texttt{bar}\ (\texttt{good}\,R{\uparrow}u)\ [] \to \texttt{bar}\ (\texttt{good}\,R)\ [u]$. Then we can show the implication $\texttt{af}\ R \to \texttt{bar}\ (\texttt{good}\,R)\ []$ by induction on $\texttt{af}\ R$. ◀

We can deduce usual the sequential characterization of almost full relations, but, as with covers and bars, this characterization is constructively weaker.

$$\texttt{af\_sequences}\ R:\ \texttt{af}\ R \to \forall \alpha,\ \exists\, i\, j,\ i < j \wedge R\,\alpha_i\,\alpha_j.$$

[21] See [18] for an equivalent inductive characterization of $\texttt{good}\,R$.
[22] `good` and `irred` view lists in opposite ways.

Proof. We obtain n such that $\texttt{good}\,R\,[\alpha_{n-1};\ldots;\alpha_0]$ using `af__bar_good_nil` above followed by `bar_sequences` from Section 3.5. We conclude by analyzing the identity $l +\!\!+ x :: m +\!\!+ y :: r = [\alpha_{n-1};\ldots;\alpha_0]$ where $R\,y\,x$ holds. ◀

We finish our tour of weakened forms of Kőnig's lemma with a slightly different form where the outcome is not a representing rose tree but just its height, hence a bound on the length of its branches. In light of Theorem 20, these are equivalent conditions for finitely branching relations. While we insisted so far on getting tree representations is the spirit of Kőnig's lemma, in its applications on e.g. termination, a bound on the height of this tree is often sufficient to conclude.

Given a sequence of relations $P : \mathbb{N} \to \texttt{rel}_1\,X$, we define a predicate $\texttt{choice\_list}\,P : \texttt{rel}_1\,(\texttt{list}\,X)$ such that $\texttt{choice\_list}\,P\,[x_0;\ldots;x_{n-1}] \leftrightarrow P_0\,x_0 \wedge \cdots \wedge P_{n-1}\,x_{n-1}$, i.e. $\texttt{choice\_list}\,P\,l$ exactly when the members of l are successive choices in P_0, P_1, etc.[23]

▶ **Theorem 25.** *Given an almost full relation, i.e. $R : \texttt{rel}_2\,X$ s.t. $\texttt{af}\,R$, and a sequence of finite unary relations, i.e. $P : \mathbb{N} \to \texttt{rel}_1\,X$ s.t. $\forall n, \texttt{finite}\,P_n$. Then the length of irredundant choice lists for P is (uniformly) bounded, i.e.:*

$$\exists n, \forall l, \texttt{choice\_list}\,P\,l \to \texttt{irred}\,R\,l \to \lfloor l \rfloor < n.$$

Proof. From `af__bar_good_nil`, we know that $\texttt{bar}\,(\texttt{good}\,R)\,[]$ holds and we apply the `FAN_bar` form of Theorem 15 and derive $\texttt{bar}\,(\lambda\,lc, \texttt{FAN}\,lc \subseteq \texttt{good}\,R)\,[]$.

We define $\texttt{support}\,n\,l \coloneqq \forall x, P_n\,x \leftrightarrow x \in l$, meaning that l is a supporting list for the (finite) unary relation P_n. We use the `bar_negative` characterization of inductive bars applied to $\texttt{bar}\,(\lambda\,lc, \texttt{FAN}\,lc \subseteq \texttt{good}\,R)\,[]$ with $Q \coloneqq \lambda\,lc, \texttt{choice\_list}\ \texttt{support}\,(\texttt{rev}\,lc)$. We get lc such that $\texttt{FAN}\,lc \subseteq \texttt{good}\,R$ and $\texttt{choice\_list}\ \texttt{support}\,(\texttt{rev}\,lc)$.

Then we check that $n \coloneqq \lfloor lc \rfloor$ satisfies the property $\forall l, \texttt{choice\_list}\,P\,l \to \lfloor l \rfloor = n \to \texttt{good}\,R\,(\texttt{rev}\,l)$. The same value then bounds the length of irredundant choice lists for P. ◀

Again, we use a combination of a FAN theorem followed by the negative characterization of inductive bars. Indeed, the assumption of finiteness $\forall n, \texttt{finite}\,P_n : \mathbb{P}$ is not strong enough to be able to build a *sequence* $\mathbb{N} \to \texttt{list}\,X$ that enumerates the respective supports for P_0, P_1,... The negative characterization allows us to reason without needing an escape from the $\mathbb{P}$ sort of Coq.

6 Two examples of replacements of Kőnig's lemma

In this section, we discuss two applications of our constructive variants of Kőnig's lemma that allow to transfer some "classical" proofs into the realm of constructive mathematics.

6.1 The decidability from implicational relevance logic

In [17], we use a variant of Kőnig's lemma for almost full relations, corresponding here to Theorem 25, to show the termination of an exhaustive proof search procedure for implicational relevance logic (IR), based on a sequent system designed by Curry [8]. The termination of this system was established by Kripke [16], building on Curry's work, rediscovering Dickson's lemma, and concluding with Kőnig's lemma.

[23] Hence, if for instance $P_n = \{\alpha_n\}$ is a singleton for any n, where $\alpha : \mathbb{N} \to X$, then there is exactly one inhabitant of $\texttt{choice\_list}\,P\,l$ for a given length of l and this list is $[\alpha_0;\ldots;\alpha_{\lfloor l \rfloor - 1}]$.

The idea of the proof is the following. Curry's sequent proof system is proved sound and complete for IR. It has three essential properties:

- each sequent rule has finitely many premises, in fact less than two;
- for any conclusion, there are only finitely many rule instances having that conclusion;
- there is a notion of redundancy for sequents such that, if a sequent S_2 is redundant over S_1, then any proof of S_2 can be contracted into a proof of S_1 of less height. This property is called *Curry's lemma*.

Kripke proved that the notion of redundancy, derived from the natural inclusion ordering on multiset, forms a well quasi order (WQO), and thus any sequence of sequents contains a redundant pair. Notice that the WQO terminology and Dickson's lemma, the key ingredient in the result, were only popularized later on.

Then, using Curry's lemma, Kripke argued that any proof search branch must contain a redundant pair, and by Kőnig's lemma, the proof search tree for irredundant proofs is finite.

Replacing the classical approach to WQOs by inductive almost full relations, in [17] we prove that the notion of redundancy is AF, the constructive form of Dickson's lemma been derived from Coquand's constructive form of Ramsey's theorem [25]. Then we use a variant of Theorem 25 called `Constructive_Koenigs_lemma` to show that the irredundant part of the proof search tree is finite. Notice that this variant is `Type`-bounded, as opposed to the $\mathbb{P}$-bounded variant presented here. Since the redundancy relation is (strongly) decidable, we could also proceed with the $\mathbb{P}$-bounded variant (i.e. Theorem 25) serving as a justification of termination for unbounded linear search.

6.2 Building Harvey Friedman's $\mathrm{TREE}(n)$ monster

In [20], we build on a Coq constructive proof of Kruskal's tree theorem [19] to implement $\mathrm{TREE}(n)$ function (that we specify below), invented and studied by Harvey Friedman [10] in his groundbreaking work on reverse mathematics.

The *(homeomorphic) embedding on rose trees* is a WQO as soon as the comparison between decorations of the nodes is itself a WQO: this is the statement of Kruskal's theorem in a classical setting. In [19], we implement a constructively provable form by replacing WQOs with (inductive) `af` relations (see Definition 24). Notice that this constructive form of Kruskal's theorem has a quite involved proof that we do not discuss here.

Using Kruskal's theorem, the homeomorphic embedding between roses trees decorated with elements of the finite set $\{1, \ldots, n\}$ is `af` and we use this relation as our redundancy relation. This means, using the sequential characterization `af_sequences` of Section 5.4, that any sequence T_1, T_2,... of roses trees contains a redundant pair. Now Friedman bounds the number of possible choice for T_i by requiring that its size (number of nodes) is less than i: we say that T_i *is sized*. Hence, considering the set of all such sized sequences $(T_i)_{0<i}$, they form a finitely branching tree and all infinite branches contain a redundant pair. Following the argumentation of e.g. [11], by Kőnig's lemma, the irredundant part of that tree is finite and thus sized sequences have maximal length, which is by definition $\mathrm{TREE}(n)$.

We circumvent this classical argumentation by applying Theorem 25, hence, according to its proof, first applying the FAN theorem for inductive bars and then the negative characterization of inductive bars. We obtain, constructively, the existence of a uniform bound on the length of irredundant sequences of sized trees $(T_i)_{0<i}$. The exact value of the bound, i.e. $\mathrm{TREE}(n)$, can then be computed by unbounded linear search [20].[24]

[24] Using a `Type`-bounded variant of Theorem 25, one can use bounded linear search instead of unbounded

7 Conclusion

Besides the Coq script that supports the results presented herein, we can summarize our contributions as following. We show that the notion of inductive cover generalizes both accessibility and bar inductive predicates, hence we can discuss concepts and results at the level of covers and they instantiate on these restricted notions as well. We follow Coquand's program [6] and replace characterizations based on sequences with inductive ones, that constructively do not fall short on lawless sequences.

We compare the strength of the positive, negative and sequential characterizations of covers, or (as an instance) of "being a bar," both in constructive and classical contexts. We analyze the precise roles played by the axioms of excluded middle and dependent choice.

The negative characterization is a remarkable intermediate notion: a) it is a De Morgan dual of the positive characterization; b) it expels determinism from the sequential characterization, and shares properties with Brouwer's notion of spread; c) it is relevant in practice, for instance when dealing with `Prop`-bounded Coq definitions.

We give a concise constructive proof of a FAN theorem for inductive covers that generalizes the type theoretic interpretation of the FAN theorem for inductive bars [9]. We notice that the respective core argument of these two proofs differ significantly.

The negative or sequential characterizations of covers (or bars) are weaker than the positive/inductive characterization. They fail when trying to constructively establish important closure properties, such as the FAN theorem. However, they can still be used constructively, after the inductive FAN theorem, to obtain uniform bounds on the length of branches of trees. This is the core argumentation behind several weaker variants of Kőnig's lemma that we derive and present, herein insisting on representations by inductive rose trees.

To conclude, we discuss two applications of those constructive variants of Kőnig's lemma that allow the transport of classical results in the constructive realm.

As a quite reasonable perspective to this work, we could implement a `Type`-bounded version of the results of the paper. Possibly, as in [18], within a unified code base, generic for both the (herein presented) `Prop`-bounded and the `Type`-bounded versions.

Almost full relations give a satisfactory constructive account for the notion of well quasi order, i.e., finitary closure properties such as Dickson's lemma, Higman's lemma and Kruskal's tree theorem can be constructively established with this notion. However, as far as we are aware, the stronger notion of better quasi order (BQO) has not yet been given a suitable inductive account, and it would be quite a challenge to lean towards an inductive definition of BQOs, hopefully satisfying additional infinitary closure properties.

References

1. Franz Baader and Tobias Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.
2. Josef Berger. Brouwer's fan theorem, 2021. `doi:10.48550/arXiv.2001.00064`.
3. Josef Berger and Hajime Ishihara. Brouwer's fan theorem and unique existence in constructive analysis. *Mathematical Logic Quarterly*, 51(4):360–364, 2005. `doi:10.1002/malq.200410038`.
4. Claudio Sacerdoti Coen and Silvio Valentini. General Recursion and Formal Topology. In Ekaterina Komendantskaya, Ana Bove, and Milad Niqui, editors, *PAR-10. Partiality and*

linear search. However, there is little to gain in obtaining an efficient algorithm for computing TREE(n) since writing TREE(3) in decimal would already exhaust all atoms of the known universe.

Recursion in Interactive Theorem Provers, volume 5 of *EPiC Series in Computing*, pages 72–83. EasyChair, 2012. doi:10.29007/hl75.

5 Robert L. Constable and Crystal Cheung. Brouwer's Fan Theorem: An Overview. 2016. URL: https://api.semanticscholar.org/CorpusID:203584866.

6 Thierry Coquand. About Brouwer's Fan Theorem. *Revue internationale de philosophie*, 230:483–489, September 2004. URL: http://www.jstor.org/stable/23955601.

7 Thierry Coquand, Giovanni Sambin, Jan Smith, and Silvio Valentini. Inductively generated formal topologies. *Annals of Pure and Applied Logic*, 124(1):71–106, 2003. doi:10.1016/S0168-0072(03)00052-6.

8 Haskell B. Curry. *A Theory of Formal Deductibility.* Notre Dame mathematical lectures. University of Notre Dame, 1957.

9 Daniel Fridlender. An Interpretation of the Fan Theorem in Type Theory. In Thorsten Altenkirch, Bernhard Reus, and Wolfgang Naraschewski, editors, *Types for Proofs and Programs*, pages 93–105, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg. doi:10.1007/3-540-48167-2_7.

10 Harvey M. Friedman. Internal finite tree embeddings. In Wilfried Sieg, Richard Sommer, and Carolyn Talcott, editors, *Reflections on the Foundations of Mathematics: Essays in Honor of Solomon Feferman*, Lecture Notes in Logic, page 60–91. Cambridge University Press, 2002.

11 Jean H. Gallier. What's so special about Kruskal's theorem and the ordinal Γ_0? A survey of some results in proof theory. *Annals of Pure and Applied Logic*, 53(3):199–260, 1991. doi:10.1016/0168-0072(91)90022-E.

12 William Hanf. Nonrecursive Tilings of the Plane. I. *The Journal of Symbolic Logic*, 39(2):283–285, 1974. URL: http://www.jstor.org/stable/2272640.

13 Hajime Ishihara. Weak Kőnig's Lemma Implies Brouwer's Fan Theorem: A Direct Proof. *Notre Dame Journal of Formal Logic*, 47(2):249–252, 2006. doi:10.1305/ndjfl/1153858649.

14 Dénes Kőnig. Über eine Schlussweise aus dem Endlichen ins Unendliche. *Acta Sci. Math. (Szeged)*, 3(2–3):121–130, 1927.

15 Stephen C. Kleene and Richard E. Vesley. *The Foundations of Intuitionistic Mathematics: Especially in Relation to Recursive Functions.* North-Holland Publishing Co., Amsterdam, 1965.

16 Saul Kripke. The Problem of Entailment (abstract). *Journal of Symbolic Logic*, 24:324, 1959.

17 Dominique Larchey-Wendling. Constructive Decision via Redundancy-Free Proof-Search. *Journal of Automated Reasoning*, 64:1197–1219, 2020. doi:10.1007/s10817-020-09555-y.

18 Dominique Larchey-Wendling. Quasi Morphisms for Almost Full Relations. In *30th International Conference on Types for Proofs and Programs, TYPES*, 2024. URL: https://easychair.org/publications/preprint/M3Km.

19 Dominique Larchey-Wendling. The Coq-Kruskal project, April 2024. URL: https://github.com/DmxLarchey/Coq-Kruskal.

20 Dominique Larchey-Wendling. The Friedman-TREE project, November 2024. URL: https://github.com/DmxLarchey/Friedman-TREE.

21 Dale Myers. Nonrecursive Tilings of the Plane. II. *The Journal of Symbolic Logic*, 39(2):286–294, 1974. URL: http://www.jstor.org/stable/2272641.

22 Stephen G. Simpson. *Subsystems of Second Order Arithmetic.* Perspectives in Logic. Cambridge University Press, 2 edition, 2009.

23 Wim Veldman. Brouwer's Real Thesis on Bars. *Philosophia Scientiæ CS 6*, pages 21–42, 2006.

24 Wim Veldman. Brouwer's Fan Theorem as an axiom and as a contrast to Kleene's alternative. *Arch. Math. Logic*, 53:621–693, 2014. doi:10.1007/s00153-014-0384-9.

25 Dimitrios Vytiniotis, Thierry Coquand, and David Wahlstedt. Stop When You Are Almost-Full. In Lennart Beringer and Amy Felty, editors, *Interactive Theorem Proving*, pages 250–265, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. doi:10.1007/978-3-642-32347-8_17.