Investigating Streamless Sets

Erik Parmann

University of Bergen
Norway
`Erik.Parmann@ii.uib.no`

Abstract

In this paper we look at *streamless sets*, recently investigated by Coquand and Spiwack [4]. A set is *streamless* if every stream over that set contain a duplicate. It is an open question in constructive mathematics whether the Cartesian product of two streamless sets is streamless.

We look at some settings in which the Cartesian product of two streamless sets is indeed streamless; in particular, we show that this holds in Martin-Löf intentional type theory when at least one of the sets have decidable equality. We go on to show that the addition of functional extensionality give streamless sets decidable equality, and then investigate these results in a few other constructive systems.

1998 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases Type theory, Constructive Logic, Finite Sets

Digital Object Identifier 10.4230/LIPIcs.TYPES.2014.187

1 Introduction

One of the interesting aspects of working in constructive mathematics is that notions often become more nuanced than they do in classical mathematics. This holds for the notion of finiteness, for instance; there are a multitude of possible definitions of a set being finite which would be equivalent classically, but are different constructively.

In this paper, we will look at a particular definition of finite sets in a constructive context, given in terms of streamless sets. This is essentially a constructive version of the classical statement that a set is finite if there are no injections from $\mathbb{N}$ into it. It is formulated positively: a set A is *streamless* when

$$\forall f : \mathbb{N} \to A, \exists i, j : \mathbb{N}, i < j \wedge f(i) = f(j).$$

It is not known who first looked at finiteness in a constructive setting, but it was recently investigated by Coquand and Spiwack [4], who look at four different definitions of a set being finite. These four are, in decreasing order of strength:

- Enumerated: there is a list containing all the elements in the set;
- Bounded: there is an $n : \mathbb{N}$ such that every list with more than n elements has duplicates;
- Noetherian: no matter how one adds elements from the set to a list, one eventually gets duplication in the list; and
- Streamless: every stream over the set contains duplicates.

They show that these notions form a strict hierarchy, except in the streamless and noetherian cases (where strictness is left open): they show that any noetherian set is streamless, and conjecture that the converse does not hold.

It is relatively easy to see that not all bounded sets are enumerated: with enumerated sets we actually have all of its elements, while with bounded sets we only know a bound on the size of the set. In fact, emptiness is in general undecidable for bounded sets, but decidable

20th International Conference on Types for Proofs and Programs (TYPES 2014).
Editors: Hugo Herbelin, Pierre Letouzey, and Matthieu Sozeau; pp. 187–201
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

for enumerated sets. Coquand and Spiwack [4] cite an example offered by F. Richman of a way to generate subsets of natural numbers with the property that one cannot *a priori* know the size of the subset, but if one gets any element in the set then one knows the size of the set. These sets are noetherian but not bounded.

Marc Bezem and other authors have a model of Martin-Löf Type Theory [6] in which there is a streamless set which is not provably noetherian, thus showing that the noetherian property is strictly stronger than streamlessness (personal communication, September 2014). This model is rather complicated and has yet to be published. The authors construct a set parameterized by a undecidable predicate on $\mathbb{N}$. Equality on this set is decidable, which is important for the proof that it is streamless. They assume Markov's principle, and use that as the "engine" which finds duplicates in the streams. They are also able to show that this set cannot be noetherian. In this way they show – since Markov's principle is consistent with Type Theory – that it is not possible to prove that streamlessness implies noetheriannes.

In addition to giving the hierarchy, Coquand and Spiwack [4] also prove several closure properties of the different notions of finiteness. They show that all four are closed under sum; that is, for any of the notions of finiteness, the sum of two finite sets is itself finite by the same notion. The situation is more complicated for Cartesian products. The two strongest notions, enumerated and bounded, are shown to be closed under products, and noetherian sets are closed under products, as long as one of the sets has decidable equality. The use of decidable equality in one of the sets in the proof in [4] was first pointed out in [2]. Whether streamless sets are closed under Cartesian products was left as an open problem.

Our main result will be the following: in Martin-Löf intentional type theory (ITT) [5] streamlessness is closed under Cartesian products, granted that one of the sets has decidable equality or is bounded.

An important feature of ITT is strong Σ-elimination. Consequently, from a proof of $\forall x \exists y.\phi(x, y)$ we are able to get, for any x, an actual y which can be used in the construction of new functions/streams. This plays an important role in the proof of our main result. In other systems, like HA which we will look at in Section 6, we need to assert a axiom of choice to get the same.

In Coq we have the choice of formalizing statements either in Set, which enjoy strong Σ-elimination, or Prop which does not. The proof we provide here will, on the face of it, only hold when streamlessness is formalized in Set; but we will see that, as long as *both* sets have decidable equality, the two formalizations actually correspond.

Decidable equality plays an important role in our proof, and we conjecture that streamlessness is not closed under products when both sets have undecidable equality. We show that, in ITT with functional extensionality, streamless sets have decidable equality, meaning that a potential counter-model must reject functional extensionality.

The main motivation behind this work is curiosity as to the strength of streamless sets, but there is also potential for practical applications. One such example is outlined in Coquand and Spiwack [4], namely automaton reachability testing. They give the regular depth-first graph algorithm for finding reachable states, and then proceed to show that if one assumes that the set of states in the automaton is finite in the sense of streamless, then this algorithm terminates. It is not uncommon to take the Cartesian product of two automata to create one which has as its language the intersection of the two original languages. Given that streamlessness is closed under product, one can show that the reachability algorithm also terminates on this new automaton.

In Section 2, we introduce streamless sets and some machinery which lets us find any number of duplicate elements. In Section 3, we prove the main theorem: that streamless sets with decidable equality are closed under Cartesian products in ITT. In Section 4, we see that adding functional extensionality gives streamless sets decidable equality. Section 5 relates our findings to Coq and its Set vs Prop distinction; it also briefly touches upon Homotopy Type Theory with Univalence. In Section 6, we relate our finding to Heyting arithmetic in the systems (E-) HA^ω. Section 7 provides a brief overview of related works; Section 8 highlights some remaining questions; and we conclude in Section 9.

1.1 Notation

We work in Martin-Löf intensional type theory (ITT) [5], where both propositions and sets are modeled as types.

We assume a inductive type $\mathbb{N}$ for the natural numbers, and we have the usual type constructors: If A is a type and B is a type family over A then both $\Pi_{x:A}B(x)$ and $\Sigma_{x:A}B(x)$ are types, the dependent function type and the dependent pair type with the usual computation rules. We use $\pi_1 : (\Sigma_{x:A}B(x)) \to A$ and $\pi_2 : \Pi_{p:\Sigma_{x:A}B(x)}B(\pi_1(p))$ as the two projections of dependent pairs. In the special cases where $B(x)$ does not depend on x, we abbreviate $\Pi_{x:A}B(x)$ as $A \to B$ and $\Sigma_{x:A}B(x)$ as $A \times B$, the latter being the Cartesian product of A and B. If A and B are types, then $A + B$ is their disjoint union with the constructors $\mathrm{inl} : A \to A + B$ and $\mathrm{inr} : B \to A + B$.

We will use the notation $\mathrm{Dec} =_A$ to stand for the type $\Pi_{x:A}\Pi_{y:A}(I_A(x, y) + \neg I_A(x, y))$, where $I_A(x, y)$ is the inductive identity type. We will use $=_A$ as an infix version of I_A, or just $=$ if the type A is clear from context. With A having decidable equality we mean that we have an inhabitant of $\mathrm{Dec} =_A$.

A stream over a set A is any function of type $\mathbb{N} \to A$. Given a stream $g : \mathbb{N} \to A$ we also have "cut" streams $g|_n : \mathbb{N} \to A$ for every $n : \mathbb{N}$ defined by

$$g|_n(x) := g(x + n).$$

When we say that we have duplicates in a stream $g : \mathbb{N} \to A$, we mean that we have two indices $i < j$ such that $g(i) =_A g(j)$.

Given a stream g over $A \times B$, we can project out two streams $g_1 : \mathbb{N} \to A$ and $g_2 : \mathbb{N} \to B$ being $g_i = \pi_i \circ g$. As usual, two elements in $A \times B$ are equal if both their first and second projection are equal. We also say that two elements in $A \times B$ are A-equal (resp. B-equal) if their first (resp. second) projections are equal.

2 Introduction to streamless sets

A set A is *streamless* if all streams over it contains duplicates; that is, for all streams $g : \mathbb{N} \to A$, we have indices $i < j$ with $g(i) =_A g(j)$. Formally, it means that we have a inhabitant of the type

$$\mathrm{Streamless}(A) := \Pi_{f:\mathbb{N}\to A}\Sigma_{p:\mathbb{N}\times\mathbb{N}}(\pi_1(p) < \pi_2(p) \times f(\pi_1(p)) =_A f(\pi_2(p))).$$

In what follows we will mostly be interested in the pair $p : \mathbb{N} \times \mathbb{N}$, and not the proof that it has the desired features. To avoid having to project out the number and clutter up the construction more than needed, we will assume that if we have a streamless set A, we have a witness

$$M_A : (\mathbb{N} \to A) \to \mathbb{N} \times \mathbb{N},$$

which, given a stream g over A, gives out two indices $i < j$ such that $g(i) = g(j)$.

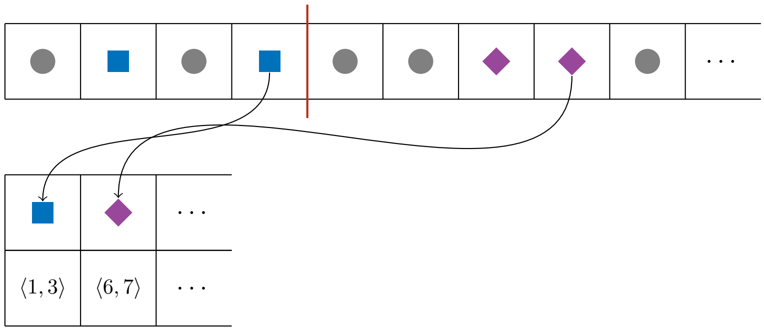

Figure 1 g^2, the stream of duplicates in g.

First, we show that if we have a stream over a streamless set B, we can find not only duplicates, but for any n we can find elements occurring at least n times. This is clear classically; we just have to look at the first $|B| \times n$ elements in the stream. Constructively it is less clear, as we do not know the actual size of the set – only that it is streamless. As seen in the introduction, one cannot, in general, deduce the size of a set from the fact that it is streamless. The first part of this construction, for $n = 2$, is also used to prove that streamless is closed under sum in [4].

Given a stream g over streamless B, we make a new stream g^2 over $B \times \mathbb{N} \times \mathbb{N}$, such that for every $\langle b, i, j\rangle$ we have $i < j$ and $g(i) = g(j) = b$, and for all $g^2(n) = \langle b, i_1, j_1\rangle$ and $g^2(n+1) = \langle c, i_2, j_2\rangle$ we have $j_1 < i_2$. We get this by letting g^2 begin with $\langle g(j), i, j\rangle$ where $\langle i, j\rangle = M_B(g)$, and then continue likewise on the stream $g|_{j+1}$.

Formally, g^2 is defined as follows, where _ indicates a value which we do not use (and thus prefer not to name).

▶ **Definition 1** ($g^2 : \mathbb{N} \to B \times \mathbb{N} \times \mathbb{N}$).

$$
\begin{aligned}
g^2(0) &= \langle g(j), i, j\rangle && \text{where } \langle i, j\rangle = M_B(g),\\
g^2(n+1) &= \langle g(j+p), i+p, j+p\rangle && \text{where } \langle \_, \_, p\rangle = g^2(n)\\
& && \text{and } \langle i, j\rangle = M_B(g|_{p+1})
\end{aligned}
$$

Figure 1 contains a visual representation of g^2, the top being g and the bottom g^2. The two blue boxes make up the first duplicate pair found by $M_B(g)$. The vertical red line indicates that this is where we "cut" the stream, and by using M_B again on this new stream we get a new duplicate pair, the purple diamonds. This process continues, defining a new stream of representatives of duplicates in g.

The first projection of g^2 is itself a B-stream, and we can then use the same process on this stream. This provides duplicate duplicates, giving us elements which occur four times in g.

We can iterate this process and, for every $n : \mathbb{N}$ and stream $g : \mathbb{N} \to B$, we get a stream $g^n : \mathbb{N} \to B \times (\mathit{List}\ \mathbb{N})$ such that every element in the new stream gives a $\langle b, l\rangle$ such that b occurs at least n times in g, at the n different indices given in l.

To formally define g^n it is easiest to first define a slightly stronger function $f^n : \mathbb{N} \to B \times (\mathit{List}\ \mathbb{N}) \times \mathbb{N}$. The last natural number is used when defining $f^{m+1}(n+1)$, it tells it

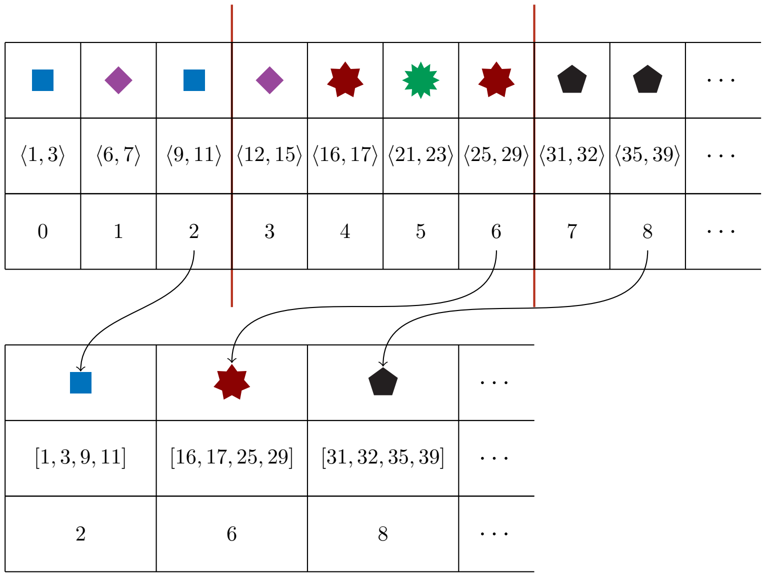

Figure 2 Calculating f^3 from f^2.

where in $(f^m)_1$ the n^{th} duplicate was found, enabling us to cut $(f^m)_1$ at the right place.

$$
\begin{aligned}
f^2(n) &= \langle b, [i_1, i_2], i_2\rangle && \text{where } \langle b, i_1, i_2\rangle = g^2(n) \\
f^{m+1}(0) &= \langle (f^m)_1(i), (f^m)_2(i) ++ (f^m)_2(j), j\rangle && \text{where } \langle i, j\rangle = M_B((f^m)_1), \\
f^{m+1}(n+1) &= \langle (f^m)_1(i), (f^m)_2(i) ++ (f^m)_2(j), j\rangle && \text{where } \langle \_, \_, p\rangle = f^{m+1}(n) \\
& && \text{and } \langle i, j\rangle = M_B((f^m)_1|_p)
\end{aligned}
$$

This process is illustrated in Figure 2, where we show how to calculate f^3 from f^2. Having f^n we define g^n by simply dropping the third number:

▶ **Definition 2** ($g^n : \mathbb{N} \to B \times \mathit{List}\ \mathbb{N}$).

$$
g^n(x) = \langle e, l\rangle \qquad \text{where } \langle e, l, \_\rangle = f^n(x)
$$

The attentive reader notices that this does not actually produce linearly many indices, but exponentially many. g^3 is actually a stream of items occurring 4 times, and g^4 is a stream of objects occurring 8 times. We will not make use of this property, and we will, for the sake of simplicity, assume that g^n contains elements that occur n times.

Observe that we use strong ∃ elimination for this construction. Not only do we know that there are indices, but we know what they are; we are also free to use them in the construction of a new stream, to which we can apply M_B once more. As mentioned above, [4] uses a stream which is the first projection of g^2 in the proof that streamless is closed under sum. We do not know of a proof that streamlessness is closed under sum which does not assume strong ∃ elimination.

3 Products of streamless sets

This section applies the machinery developed in the previous section to the product of streamless sets.

We will first see that the Cartesian product of a bounded set with a streamless set is streamless. It is worth noting that this is independent of whether any of the sets has decidable equality or not.

▶ **Lemma 3.** *In ITT we have: If at least one of A and B is bounded and the other is streamless then $A \times B$ is streamless.*

Proof. We assume that A is bounded by n. (If it were B then the construction below would be "mirrored".) Given the stream $g : \mathbb{N} \to A \times B$ we look at $g_2 : \mathbb{N} \to B$, its second projection. By looking at $(g_2)^{n+1}(0)$ we get a pair $\langle b, [i_0, \ldots, i_n] \rangle$ such that b occurs at all the indices $i_0, \ldots, i_n$ in g_2. Note that $g_1(i_0), \ldots, g_1(i_n)$ are $n+1$ elements of A, so since A is bounded by n, there must be at least two indices $i_k < i_l$ such that $g_1(i_k) = g_1(i_l)$. As $g_2(i_k) = b = g_2(i_l)$, we get $g(i_k) = g(i_l)$. ◀

We now show that Markov's principle and decidable equality of one of the sets imply that streamlessness is closed under product. This result is a warm up for the later, more general result shown in Theorem 6. The proofs have interesting similarities, especially in how we can use the streamlessness of a set to "emulate" Markov's principle.

First a reminder of Markov's principle.

▶ **Definition 4** (Markov's principle)**.** For decidable predicates P on $\mathbb{N}$ we have $\neg\neg\Sigma_{x:\mathbb{N}} P(x) \to \Sigma_{x:\mathbb{N}} P(x)$.

Markov's principle has a quite computational flavour, which unsurprisingly makes it easier to prove a set streamless. All we need to do to find the duplicate indices is to show that it cannot be the case that they do not exist.

▶ **Lemma 5.** *In ITT with MP we have: If at least one of A and B has decidable equality and A and B are both streamless then $A \times B$ is streamless.*

Proof. We assume a stream $g : \mathbb{N} \to A \times B$. We also assume, without loss of generality, that A is the set with decidable equality. (If it were B then the construction below would be "mirrored".)

We define the following predicate on $\mathbb{N}$:

$$P(n) := \text{For } \langle \_, [i_0, \ldots, i_{n-1}] \rangle = (g_2)^n(0) \text{ we have duplicates in } [g_1(i_0), \ldots, g_1(i_{n-1})].$$

Remember that g_2 gets the B-stream, and $(g_2)^n$ finds n indices with equal elements. Note that if A has decidable equality, $P(n)$ is decidable.

We now proceed to show that (1) $\neg\neg\exists n P(n)$ and that (2) from $\exists n P(n)$ we can get $\langle i, j \rangle$, with $i < j$ and $g(i) = g(j)$.

Proof of (1). We assume $\neg\exists n P(n)$ and proceed to produce a contradiction. $\neg\exists n P(n)$ implies $\forall n \neg P(n)$, which says that for any n we have that for $\langle b, [i_0, \ldots, i_{n-1}] \rangle = (g_2)^n(0)$ the list $[g_1(i_0), \ldots, g_1(i_{n-1})]$ has no duplicates. Notice that the list $[g_1(i_0), \ldots, g_1(i_{n-1})]$ has n elements, all from A.

We now make a duplicate-free stream $f : \mathbb{N} \to A$; that is, for every $n : \mathbb{N}$, we have for all $j < n$ that $f(n) \neq f(j)$, contradicting that A is streamless. Defining $f(n)$ we first find $n+1$ indices with the same b element, $\langle \_, [i_0, \ldots, i_n] \rangle = (g_2)^{n+1}(0)$. We now let

$$f(n) = ([g_1(i_0), \ldots, g_1(i_n)] \setminus [f(0), \ldots f(n-1)])(0).$$

That is, $f(n)$ is the first element in the list resulting from removing any element from $[g_1(i_0), \ldots, g_1(i_n)]$ that the stream already contains. As $[g_1(i_0), \ldots, g_1(i_n)])$ contains $n+1$

different elements from A, we know that the resulting list is non-empty. Since this stream only outputs elements which have not been output up until that point, it will never introduce a duplicate pair. Thus we have contradicted that A is streamless, enabling us to conclude $\neg\neg\exists n P(n)$. ◀

Proof of (2). From $\exists n P(n)$ we have that there is an n such that for $\langle b, [i_0, \ldots, i_{n-1}]\rangle = (g_2)^n(0)$ the list $[g_1(i_0), \ldots, g_1(i_{n-1})]$ has duplicates. Let those indices be $i_k < i_l$. Since every element in $g(i_0), \ldots, g(i_{n-1})$ has b as its second coordinate, we get that $g(i_k) = g(i_l)$. ◀

Having proved $\neg\neg\exists n P(n)$, we apply Markov's principle and get $\exists n P(n)$. By (2) above, this gives us the indices $\langle i, j\rangle$ with $i < j$ and $g(i) = g(j)$. ◀

We will now proceed to get rid of Markov's principle. Several parts of the previous proof will be recognisable, but we use the streamlessness of the two underlying sets to do the work that Markov's principle did in the previous proof.

▶ **Theorem 6.** *In ITT we have: If at least one of A and B has decidable equality and A and B are both streamless, then $A \times B$ is streamless.*

Proof. We assume that A is the set with decidable equality, and we want to construct

$$M_{A\times B} : (\mathbb{N} \to A \times B) \to \mathbb{N} \times \mathbb{N},$$

which, given any $A \times B$-stream g, finds a pair of indices $i < j$ such that $g(i) = g(j)$.

Given an $A \times B$-stream g, we inductively define an A-stream f by letting $f(n)$ first look at $f(m)$ for all $m < n$, and see if two equal elements are outputted. This can be done since A has decidable equality. If there is a duplicate element, $f(n)$ outputs it. If there are no duplicates outputted so far, we let $f(n)$ look at all the A-elements corresponding to the n B-elements given by $(g_2)^n(0)$. Remember, $(g_2)^n(0) = \langle b, l\rangle$ where $l = [i_0, \ldots, i_{n-1}]$ is a list of n indices. Looking up these indices in g_1 gives us a list $la : List\ A$ of n A-elements.

By using the decidability of A, we can check whether there are duplicate elements in la. In the case of no duplicates, we know that there must be at least one of the n elements which does not already occur in f so far (as we have only produced $n-1$ elements so far). We can check which one this is, as we have already defined f up to $n-1$. We then let $f(n)$ be one of those elements which has not occurred in f so far. More precisely,

$$f(n) = ([g_1(i_0), \ldots, g_1(i_{n-1})] \setminus [f(0), \ldots f(n-1)])(0).$$

If, on the other hand, there is some duplicate element in the list, we let $f(m)$ be that element for all $m \geq n$. Notice that if this is the case, this is the first time a duplicate is introduced in f. This completes the construction of $f : \mathbb{N} \to A$, and we will now use f to find duplicates in $g : \mathbb{N} \to A \times B$.

From the construction of f, we have the following property:

▶ **Lemma 7.** *For the smallest i such that $f(i) = f(i+1)$ we have duplicates in the list $[g(l_0) \ldots, g(l_{i-1})]$, where $\langle b, [l_0 \ldots, l_{i-1}]\rangle = (g_2)^i(0)$.*

As A is streamless and f is a A-stream, we can use M_A to find indices $k < d$ of duplicates in f. Since A had decidable equality, we can do a bounded search downward from k to find the first index i such that $f(i) = f(i+1)$. By Lemma 7, we have duplicates in $[g(l_0) \ldots, g(l_{i-1})]$ where $\langle b, [l_0 \ldots, l_{i-1}]\rangle = (g_2)^i(0)$. Thus, we have two indices $l_k < l_m$ in l such that $g_1(l_k) = g_1(l_m)$. By construction all the indices in $[l_0 \ldots, l_{i-1}]$ are B-equal, so $g_2(l_k) = g_2(l_m)$, giving $g(l_k) = g(l_m)$. ◀

Finally observe that if it were B and not A that had decidable equality, the construction above would be "mirrored"; f would have to be a B-stream, and we would use $(g_1)^n(0)$ instead of $(g_2)^n(0)$.

As an example of the construction, let us look at a particular calculation of $f(4)$, where no duplicates have been found so far. That means that so far f looks like

$$f = a_0, a_1, a_2$$

with none of them being equal any other.

$(g_2)^4(0)$ is $\langle b, [n_0, n_1, n_2, n_3\rangle]$, giving four indices in g with the same b-element. This means that g looks somewhat like

$$g = \dots, \langle b, a'_0\rangle \dots, \langle b, a'_1\rangle, \dots, \langle b, a'_2\rangle, \dots, \langle b, a'_3\rangle, \dots$$

By the decidability of A, we can check whether there is a duplicate among $[a'_0, a'_1, a'_2, a'_3]$. If not, then we know that there is some element in $[a'_0, a'_1, a'_2, a'_3] \setminus [a_0, a_1, a_2]$, and we let $f(4)$ be the first such element. If there are duplicates, e.g $a'_1 = a'_3$, we let $f(n) = a'_1$ for all $n \geq 4$.

Comparing this proof with the proof using Markov's principle we see that we can use the streamlessness of one of the underlying sets to search for the n which gives us A-duplicates. The trick is to control exactly when duplicates are introduced in the f-stream, and then use the streamlessness of A to recover this point.

We combine Lemma 3 and Theorem 6 to get the following corollary.

▶ **Corollary 8.** *In ITT we have: If at least one of A and B has decidable equality or is bounded, and A and B are both streamless, then $A \times B$ is streamless.*

4 Streamlessness and decidable equality

It should be clear by now that decidable equality of the underlying set is quite important for the ability to produce streamless sets; we will see another indication of this in this section. We will show that in ITT, functional extensionality give streamless sets decidable equality. In addition to showing the close relation between finiteness and decidable equality, it is relevant to the search for a potential counter-model to the claim that streamlessness is closed under Cartesian products even without decidable equality.

As a warm-up, we look at the situation where the set is not only streamless, but bounded. Remember that this means that we have an $n : \mathbb{N}$ such that, for every A-list of more than n elements, we can find a duplicate pair. Formally, this means that we have an inhabitant of the type

$$\text{Bounded}(A) := \Sigma_{n:\mathbb{N}}(\Pi_{l:list A}(\text{len}(l) > n \to \Sigma_{i,j:\mathbb{N}}(i < j \times l[i] = l[j]))).$$

If we want to determine whether a_1 is equal to a_2 we make a list l of $n + 1$ instances of a_1, and get a pair of indices $i_1 < j_1$ with duplicates in this list. We then proceed to swap the element at $l[i_1]$ with a_2, giving a new list. The original list is equal the new list if and only if $a_1 = a_2$.

We then proceed to get two indices $i_2 < j_2$ of duplicate elements in this new list. If this process is assumed to be a function, and thus provide equal outputs for equal inputs, we get $\langle i_1, j_1\rangle = \langle i_2, j_2\rangle$ if and only if $a_1 = a_2$; and since equality on $\mathbb{N}$ is decidable, we are done.

Our proof turned on the facts that (1) the second projection of a witness of Bounded(A) is a function, (2) this function can be assumed to respect equality on its input, and (3) two lists are equal if and only if they are pointwise equal.

We will now mirror this with streamless sets. One major difference between lists and streams is the following: while lists are equal whenever their elements are equal, this only holds for streams if we assume so. It is consistent to assume an inhabitant of the following type in ITT, and if we do so for all types, we say that we have *functional extensionality.*

▶ **Definition 9** (FunExt(A))**.**

$$\mathrm{FunExt}(A) := \Pi_{f,g:\mathbb{N}\to A}(\Pi_{n:\mathbb{N}}(f(n) =_A g(n)) \to f =_{\mathbb{N}\to A} g)\,.$$

▶ **Lemma 10.** *In ITT with functional extensionality we have: If A is streamless then it has decidable equality.*

Proof. We assume an inhabitant of FunExt(A) and two elements $a, b : A$, and we proceed to determine their equality. Let the stream f_a be the constant A-stream consisting of only a, and let $\langle i, j\rangle$ be the indices returned by $M_A(f_a)$. We now make the stream f'_a which is constantly a, except at index i, where it is b:

$$f'_a(n) = \begin{cases} b & \text{if } n =_{\mathbb{N}} i \\ a & \text{otherwise} \end{cases}$$

Notice that if $a =_A b$ we have $\Pi_{n:\mathbb{N}} f_a(n) =_A f'_a(n)$, so from functional extensionality we then have $f_a = f'_a$. So, by functionality of M_A, we get $a = b \to M_A(f_a) = M_A(f'_a)$, and thus

$$M_A(f_a) \neq M_A(f'_a) \to a \neq b.$$

Concluding, if $M_A(f'_a) \neq \langle i, j\rangle$ then $a \neq b$, and if $M_A(f'_a) = \langle i, j\rangle$ then $a = b$ (as $f'_a(i) = b$ and $f'_a(j) = a$), and since equality on $\mathbb{N}$ is decidable we are done. ◀

Lemma 10 is relevant for the search of a counter-model to the general claim that streamlessness is closed under product. From section 3, we know that such a counter-model must have two streamless sets with undecidable equality. This section shows that the model must also reject functionality extensionality for us to have a streamless set with undecidable equality.

It also highlights some of the difficulty of defining finiteness for sets with undecidable equality in a computational setting, and since the other notions of finiteness given in [4] imply streamlessness, this result also covers them. All the definitions of finiteness have some sort of equality/duplication check at their core. Given this it seems plausible that a proof of finiteness can, in certain situations, lead to decidability. On the other hand, it is quite unsatisfactory that, in certain settings, we are unable to define finite sets of elements with undecidable equality.

In the next section we look at how to formalize both this and the previous results in Coq.

5 Formalization in Coq and HoTT

5.1 Coq: Prop and Set

In this section we will relate the above results to the proof assistant Coq [3], where we have to deal with the distinction between Prop and Set. Functions, which is how we defined streams, live in the universe Set, while there is a separate universe Prop for propositions. The intention is, roughly, to separate between types where we care about the internal structure of the inhabitants (Set) and where we care only about the existence of the inhabitant (Prop).

Given a inhabitant of a type in Prop one is generally not allowed to eliminate on it to construct elements in Set; thus we can not build the new stream g_2 of duplicates using indexes found from a witness of a type in Prop. This means that the constructions given in this paper can not be implemented in Coq *as they stand* if streamless is written as follows:

```
Definition StreamlessEx(A:Set):= forall g:nat → A,
          exists i j, i<j ∧ g(i) = g(j).
```

One way to remedy the situation is to define the notion of a set being streamless in the following way, closer to the way it was encoded in ITT. The notation "$\{x : nat \mid P(x)\}$" is Coq's notation for $\Sigma_{x:\mathbb{N}} P(x)$.

```
Definition StreamlessSig (A:Set):= forall g:nat→ A,
          {ij : nat*nat | fst ij < snd ij ∧ g(fst ij)=g(snd ij)}.
```

StreamlessSig enables us to use the proof of a set being streamless in a computation; in particular it enables us to construct the stream g_2 needed to prove Corollary 8 in Coq. The disadvantage is that it can make it harder to prove sets to be streamless in the first place. There is reason to believe that there are fewer sets satisfying StreamlessSig than StreamlessEx.

In general, whether one wants the statement in Prop or in Set reflects whether one wants to work proof relevant or not; formalizing it as StreamlessSig enables us to use the proof (of a set being streamless) in a computation.

StreamlessSig A implies StreamlessEx A, while the provability of the converse implication is unknown. Interestingly, it *is* know for sets with decidable equality, since we are able to prove the following lemma in Coq for A with decidable equiality, making the two notions of streamless coincide in those cases.

```
Lemma streamlessExToStrSig(A:Set)(A_dec: DecidableEq A) :
                         StreamlessEx A → StreamlessSig A
```

Essential for the proof is the following lemma, holding for decidable predicates P on $\mathbb{N}$, and shown in the Coq library *Coq.Logic.ConstructiveEpsilon*[1].

```
Lemma constructive_indefinite_ground_description_nat :
      (exists x : nat, P x) → {x : nat | P x}.
```

With the indefinite ground description the proof is straightforward. We assume that we have some pairing/decoding functions enabling us to encode pairs of natural numbers as single natural numbers. We then define versions of both StreamlessEx and StreamlessSig using single numbers, prove that the single and paired versions are equivalent, and then it is a simple application of the indefinite ground description given above.

The conclusion is the following corollary:

▶ **Corollary 11.** *In Coq we can prove that StreamlessEx (and StreamlessSig) of sets with decidable equality is closed under Cartesian products.*

A natural question is whether we can strengthen this to say that StreamlessEx is closed under Cartesian products as long as *at least one* of the sets have decidable equality. Unfortunately, this does not follow from the current construction. To see this, assume an $A \times B$-stream g. The construction in Proof 3 uses $(g_2)^n$ to find n-indices with B-equal

[1] http://coq.inria.fr/library/Coq.Logic.ConstructiveEpsilon.html

elements. But for this to be definable in Coq using the technique above, B needs decidable equality. The proof then uses the decidability of A to eliminate on whether there are duplicates among the resulting A-elements or not. It does not seem possible to manipulate the construction such that it is enough for only one of the sets to have decidable equality.

We are also able to reproduce Lemma 10 in Coq for StreamlessSig, the proof is simply a direct Coq formalization of the proof given in Section 4.

```
Lemma strSigAndFuncExtImpliesDecA (A:Set) (Ma:StreamlessSig A)
   (fext: functional_extensionality nat A): forall a b :A, {a=b}+{not(a=b)}.
```

Again, we are not able to simply adapt the proof to StreamlessEx, since the proof crucially uses the indexes returned from M_A in the construction of new functions.

All the Lemmas in this section have been formalized and proved in Coq[2].

5.2 HoTT

Closely related to the Prop/Set distinction is the truncated and non-truncated statements one encounters in Homotopy Type Theory (HoTT). Truncation is a type former which "truncates" a type – removing all information contained in the inhabitants of that type except their existence – and it is written as $||A||$ for a type A. (For more information we refer the reader to the freely available book [10].) We will not go further into HoTT here; but what is relevant for us is that we have a HoTT version of the indefinite ground description above. For decidable predicates P we have

$$||\Sigma_{n:\mathbb{N}} P(n)|| \to \Sigma_{n:\mathbb{N}} P(n)$$

as stated by exercise 3.19 in [10]. One should be able to reproduce a version of Corollary 11 in this setting, getting that for the non-truncated version of streamless it is enough for one of the sets to have decidable equality for streamlessness to be closed under Cartesian products.

With our current knowledge we need both sets to have decidable equality for the truncated version to be closed under Cartesian products without further assumptions, and we conjecture that this is in fact a strict requirement. If we choose to assume the HoTT-version of the axiom of choice,

$$(\Pi_{x:X} ||\Sigma_{a:A(x)} P(x,a)||) \to ||\Sigma_{g:\Pi_{(x:X)} A(x)} \Pi_{x:X} P(x, g(x))||,$$

we can show that truncated-streamless sets are closed under products as long as one of the sets has decidable equality.

In HoTT we can also assume the Univalence axiom, giving that isomorphic structures can be identified. Importantly, the univalence axiom implies functional extensionality. Lemma 10 makes it clear that – unless we want every streamless set to have decidable equality – we must use the truncated version of streamlessness in this setting.

6 HA$^\omega$

It is natural to ask how closely coupled the above results are to the particular constructive setting we are working in, and whether we can reproduce them in a different setting. We will now look at how the results fit in the system HA$^\omega$, an extension of Heyting Arithmetic to the language of finite types, see [9] for more information on HA$^\omega$.

[2] The Coq-script can be found at `https://github.com/epa095/streamless-in-coq`.

HA^ω is proof-irrelevant and does not have strong Σ elimination; instead, we have to use the axiom of choice to extract a function, giving the witnesses which we can then use as terms in the logic.

The set of finite types $\mathcal{T}$ is built from the basic type 0 ($\mathbb{N}$) and is closed under $\times$ and $\to$. HA^ω is "neutral" in the terminology of [9]; we do not assume decidability of $=_\tau$ for any other types than 0, nor do we assume that equality between functions is extensional.

Sets are not a primitive notion in HA^ω, so when talking about sets we mean functions of the type $A : \tau \to 0$; such functions represent the set of elements on which it returns 1. This means that all sets will have decidable membership and sets can only contain elements of one and the same type. For a set $A : \tau \to 0$, we call τ the enclosing type of A. Following [9] we will write $a < b$ in place of $< (a, b) = 1$, where the latter is the characteristic function of the less-than relation. With "a stream over A" we mean a function $f^{0\to\tau}$ where τ is the enclosing type of A such that $\forall n^0(A(f(n)) = 1)$.

Streamlessness of A_τ in this setting is expressed as

$$\text{Streamless}(A_\tau) := \forall g^{0\to\tau}((\forall n^0 A(g(n)) = 1) \to \exists i^0 j^0(i < j \land g(i) = g(j))).$$

In order to formalize our results in HA^ω, we first need to define some axioms. $AC_{\sigma,\tau}$ is the following axiom schema,

$$AC_{\sigma,\tau} := \forall x^\sigma \exists y^\tau \phi(x, y) \to \exists z^{\sigma\to\tau} \forall x^\sigma \phi(x, zx),$$

and AC is the axiom schema consisting of $AC_{\sigma,\tau}$ for all types $\sigma, \tau \in \mathcal{T}$. $EXT_{\sigma,\tau}$ is the following axiom schema,

$$EXT_{\sigma,\tau} := \forall y^{\sigma\to\tau} z^{\sigma\to\tau}((\forall x^\sigma, yx = zx) \to y = z),$$

and if we add $EXT_{\sigma,\tau}$ for all types $\sigma, \tau \in \mathcal{T}$ we get the system E-HA^ω.

To reproduce the proof of Lemma 5 in HA^ω, we need to construct the function g^2 of duplicates, and for this we need access to, for every stream, a pair of indexes with duplicate elements in that stream. The following instance of AC for every type τ enclosing a streamless set is enough to mirror Lemma 5 in HA^ω.

$$AC_{0\to\tau,0} := \forall x^{0\to\tau} \exists y^0 \phi(x, y) \to \exists z^{(0\to\tau)\to 0} \forall x^\sigma \phi(x, zx)$$

Let the $\phi(x, y)$ stand for the predicate "$(\forall i^0 A(x(i)) = 1) \to y$ encodes a pair of indexes $i < j$ such that $x(i) = x(j)$". Then the antecedent of $AC_{0\to\tau,0}$ follows immediately from A being streamless, and the result is the function M_A, needed to reconstruct the machinery in the proof of Lemma 5.

▶ **Corollary 12.** *In* $HA^\omega + AC$ *we have that streamless sets are closed under products.*

Encoding sets by their characteristic functions yields decidable membership, but in general not decidable equality. The extensionality of E-HA^ω, giving that streams are equal when they are pointwise equal, enables us to mirror Lemma 10:

▶ **Corollary 13.** *In* E-$HA^\omega + AC$ *we have that streamless sets have decidable equality.*

Note that E-$HA^\omega + AC$ does not prove the law of excluded middle, as it is conservative over HA. For further details, see [1].

7 Related work

One of the first investigations of streamlessness known to the author is by Richman and Stolzenberg [8]. In their terms, a streamless set is called **2**-good, where **2** is the set of two-element subsets of the natural numbers. They show that the sum of two **B**-good sets, of which **2**-good is an instance, is **B**-good, but leave it open for products. This paper does not resolve any of their open questions, as they work in a more general setting than equality. They also give another notion, that of a set being *bar-good*, and they show that the Cartesian product of a bar-good set with a **B**-good set is **B**-good. It is not clear what the relation between streamlessness and bar-good is, and whether there are natural axioms one can assume to make a streamless set bar-good.

Veldman and Bezem [11] investigate the constructive content of the Ramsey theorem [7], giving a constructive proof of a reformulation of it. For this, they use what they call *almost-full* binary relations; relations R on $\mathbb{N}$ where, for every *increasing* function $f : \mathbb{N} \to \mathbb{N}$,

$$\exists m, n : \mathbb{N}, m < n \wedge R(f(m), f(n)).$$

They postulate the axiom of bar-induction, and with it they prove that almost-full relations are closed under intersection. They name this the Intuitionistic Ramsey Theorem, and show that it is classically equivalent to Ramsey's Theorem.

Using equality as the relation R, one gets a notion which comes quite close to streamlessness, apart from Veldman and Bezem's requirement that the functions are increasing, and the fact that streamlessness is a concept applicable for any type (not only $\mathbb{N}$), possibly with undecidable equality.

In light of this, it is natural to ask whether the proofs in this paper can be generalized to relations other than equality. We define what it means for a reflexive and transitive relation R on A to be a well-quasi-ordering:

$$\mathrm{Wqo}_A(R) := \forall g : \mathbb{N} \to A, \exists i, j : \mathbb{N}, i < j \wedge R(g(i), g(j)).$$

Note that a set A is streamless exactly when we have $\mathrm{Wqo}_A(=_A)$. We can ask if the intersection of two such relations is itself a Wqo and whether the proof of Lemma 6 suggest how this could be shown. Unfortunately, we do not see how. We are still able to use the construction g^n to find n elements $a_1, \ldots a_n$ such that $a_1 R a_2 \ldots R a_n$, but we do not have the property that with n elements $b_1, \ldots b_n$ such that *none* of them are R-related, and $n-1$ elements $b'_1, \ldots b'_{n-1}$ such that none of them are R-related, there must be one of the $b_1, \ldots b_n$ which is not R related to any of the $b'_1, \ldots b'_{n-1}$. We have this property when the relation R is equality, and this is used in the proof of Lemma 6.

If we *did* have that Wqo relations were closed under intersection we would immediately get that streamless sets are closed under products: define the relation R_1 on $A \times B$ as $(a_1, b_1) R_1 (a_2, b_2)$ if and only if $a_1 =_A a_2$, and likewise for R_2, looking at the second projections. If A and B are streamless sets, then R_1 and R_2 are Wqo relations and their intersection is equality on $A \times B$.

Vytiniotis, Coquand and Wahlstedt [12] provide an inductive formulation of almost full relations on arbitrary types. They show – if we instantiate their proofs with the relation being equality – that it implies streamlessness, and show that almost-full relations are closed under intersection.

Streamlessness works in a quite general setting, with few assumptions on the underlying set. Bezem et al. [2] impose further restrictions, and the result is a interesting hierarchy of finiteness notions. The restrictions imposed are that equality is decidable; that the subset is defined by some decidable predicate; and that the set is a subset of some set that can be enumerated. This holds for decidable subsets of natural numbers in particular. The authors find six different formalizations and put them into a hierarchy.

8 Remaining questions

There are several questions remaining. The main one is whether one can show that streamlessness is closed under Cartesian products in ITT without assuming decidable equality. Secondly, to what degree can one show similar results in systems without strong Σ elimination – for example, for StreamlessEx in Coq or the truncated statement in HoTT? And what is the relationship between StreamlessEx and StreamlessSig for sets with undecidable equality?

We conjecture that there exists a model showing that, in ITT, the product of two streamless sets with undecidable equality is not necessarily streamless. From Lemma 10 we know that such a model must reject functional extensionality, and from Lemma 3, we know that neither of the sets can be bounded.

At this point there are, to this author's knowledge, only two sets which are known to be streamless but not bounded. One is the set presented in [4], originally suggested by F. Richman, showing that not all noetherian sets are bounded. As noetherian sets are streamless, this is also a streamless set. But this set has the interesting property that, once one looks at any of the elements in the set, one knows the size of the set! So it is not bounded *a priori*, but if one is given a stream of elements from the set, one can deduce its size and then continue as in the proof of Lemma 3.

The second set, presented in the still unpublished article by Bezem et al. showing that not all streamless sets are noetherian, does not have this property. On the other hand, it has decidable equality, rendering it useless as a counter-model. There does not seem to be an easy way to tweak the model to get rid of this decidable equality; it is essential for the proof that the set is streamless as the authors use Markov's Principle to find the duplicate pair, and Markov's Principle is only applicable for decidable predicates.

To conclude, we currently have no good candidate for a streamless set with a non-streamless Cartesian product. Constructing a suitable streamless set, non-bounded and with undecidable equality, appears to be quite complicated. Neither of the ways used to prove a set streamless – that is, by gathering information about the size of the set encoded in the elements themselves, or using Markov's principle – is likely to work. It seems the most promising route to a counter-model involves finding novel ways to construct streamless sets.

Lastly, we would like to encourage other to look for new notions of finiteness, especially trying to find notions that works nicely and robustly for sets with undecidable equality.

9 Conclusion

We showed that, in Martin-Löf intensional type theory, if at least one of the streamless sets A and B has decidable equality or is bounded, then the Cartesian product $A \times B$ is streamless. We also saw that adding functional extensionality to ITT gives streamless sets decidable equality; and we mirrored these results in both (E-) $\mathrm{HA}^\omega + \mathrm{AC}$ and in Coq.

Acknowledgements. I want to thank Arnaud Spiwack and Thierry Coquand for their valuable feedback and questions after my presentation on this topic at TYPES 2014, and Marc Bezem for both introducing this problem to me and discussing it with me. I also wish to thank the anonymous reviewers for their challenging feedback which led to significant changes to this paper, and Maja Jaakson which kindly proofread it.

References

1. Michael Beeson. Goodman's theorem and beyond. *Pacific Journal of Mathematics*, 84(1):1–16, 1979.
2. M. Bezem, K. Nakata, and T. Uustalu. On streams that are finitely red. *Logical Methods in Computer Science*, 2011. `http://www.cs.ioc.ee/~keiko/papers/finiteness.pdf`.
3. The Coq Development Team. *The Coq Reference Manual, version 8.4*, August 2012. Available electronically at `http://coq.inria.fr/doc`.
4. Thierry Coquand and Arnaud Spiwack. Constructively finite? In *Contribuciones científicas en honor de Mirian Andrés Gómez*, pages 217–230. Universidad de La Rioja, 2010.
5. Per Martin-Löf. An intuitionistic theory of types: predicative part. In H. E. Rose and J. C. Shepherdson, editors, *Proceedings of the Logic Colloquium 1973*, volume 80 of *Studies in Logic and the Foundations of Mathematics*, pages 73–118. North-Holland, 1975.
6. Per Martin-Löf and Giovanni Sambin. *Intuitionistic type theory*, volume 17. Bibliopolis Naples, 1984.
7. F. P. Ramsey. On a problem of formal logic. *Proceedings of the London Mathematical Society*, 2(1):264–286, 1930.
8. F. Richman and G. Stolzenberg. Well quasi-ordered sets. *Advances in Mathematics*, 97(2):145–153, 1993.
9. Anne Sjerp Troelstra and Dirk Van Dalen. *Constructivism in mathematics*, volume 2. Elsevier, 1988.
10. The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. `http://homotopytypetheory.org/book`, Institute for Advanced Study, 2013.
11. Wim Veldman and Marc Bezem. Ramsey's theorem and the pigeonhole principle in intuitionistic mathematics. *Journal of the London Mathematical Society*, 2(2):193–211, 1993.
12. Dimitrios Vytiniotis, Thierry Coquand, and David Wahlstedt. Stop when you are almost-full. In *Interactive Theorem Proving*, pages 250–265. Springer, 2012.