My HoTT notes

12 June 2020 - ongoing

1 Searchability

$$\texttt{Searchable}\ A :\equiv \prod p : A \to \mathbf{2}. \left(\sum x : A. p\ x = \texttt{tt}\right) + \prod x : A. p\ x = \texttt{ff}$$

In words: the type A is searchable if for every boolean predicate p we can find an element of A that satisfies p or else prove that no element of A satisfies p.

Theorem 1.1. The type `Searchable` A can be of any h-level.

Proof. $p\ x = \texttt{tt}$ and $p\ x = \texttt{ff}$ are propositions, the summands are disjoint, and dependent products preserve h-level (well, unless the domain is $\mathbf{0}$), so the h-level is determined by A.

If A is contractible, then `Searchable` A is contractible.

If A is a proposition, then `Searchable` A is a proposition.

If A is of h-level n, then so is `Searchable` A.

Hope my mental techniques for h-levels work :)

□

$$\texttt{MerelySearchable}\ A :\equiv \forall p : A \to \mathbf{2}. (\exists x : A. p\ x = \texttt{tt}) \lor \forall x : A. p\ x = \texttt{ff}$$

In words: a type is merely searchable if we can decide whether there merely exists an element that satisfies the given boolean predicate p, but in general there is no way to find such an element.

Theorem 1.2. `isProp` (`MerelySearchable`A)

Proof. We use the truncated logic. □

Theorem 1.3. Some instances of searchable types.

1. Searchable types include: $\mathbf{0}, \mathbf{1}, \mathbf{2}$, all finite types.

1.5. The interval is searchable.

2. If A and B are searchable, so are $A \times B$ and $A + B$.

3. If A is searchable and $B : A \to \mathcal{U}$ is a family of searchable types, $\sum x : A.B\ x$ is also searchable.

4. If A is searchable and $f : A \to B$, then the image of f is also searchable.

5. Very importantly, the type $\mathbb{N}_\infty$ is searchable.

6. There are arbitrarily big searchable ordinals.

Proof. 1. For finite types: we assume that "finite" means "there is a list of all the elements". So, use it to check every element, and poof – done.

1.5. The interval is contractible, thus equivalent to $\mathbf{1}$, so it's searchable.

2. To search $A + B$ first search A, then search B.

3. To search $A \times B$ and $\sum x : A.B\ x$ search for an a, and when looking for it, search for a b that can be paired with it.

4. To search the image of $f : A \to B$ using the predicate $p : B \to \mathbf{2}$, search the whole of A using the predicate fp (beware: I write composition of two functions f and g like fg!).

5. The searchability of $\mathbb{N}_\infty$ was established by Martin Escardó.

6. Proved by Escardó.

□

Theorem 1.4. Taboos

Searchability of $\mathbb{N}$ is taboo. In general, preservation of searchability for $\prod$, `W` and `M` and $=$ should be taboo.

Proof. $\mathbb{N}$ is the primordial searchability taboo – how can we search infinitely many numbers in a finite time (besides checking every next number twice as fast as the previous one)?

`W` can be used to define $\mathbb{N}$, so its searchability should be taboo.

What about `M`? Lists should in general not be searchable, so infinite lists shouldn't be too... but that's a dangerous line of reasoning.

As for Π, I'd guess that, even though $\mathbb{N}_\infty$ and $\mathbf{2}$ are searchable, the searchability of $\mathbb{N}_\infty \to \mathbf{2}$ is taboo.

TODO: check the two above paragraphs. □

A good question yet to be answered: does there purely exist a type that is not searchable? At first one might think about the universe $\mathcal{U}$, but at a second glance one may think the universe should satisfy some kind of Rice's theorem, so maybe all boolean predicates on it are constant... and thus the universe should be searchable. To answer this question, let's take a look at some classical axioms.

(Remark by Escardó: if there is $p : \mathcal{U} \to \mathbf{2}$ such that $p\ X \neq p\ Y$ for some $X, Y : \mathcal{U}$, then WLPO holds)

Theorem 1.5. Relation to excluded middle

1. $\prod A : \mathcal{U}.\texttt{Searchable}\ A \to A + \neg A$
2. Conclusion: $(\prod A : \mathcal{U}.\texttt{Searchable}\ A) \to \prod A : \mathcal{U}.A + \neg A$
3. Conclusion 2: $\neg \prod A : \mathcal{U}.\texttt{Searchable}\ A$
4. $\texttt{LEM} \simeq \prod P : \texttt{Prop}.\texttt{Searchable}\ P$
5. $\texttt{LEM} \simeq \prod A : \mathcal{U}.\texttt{MerelySearchable}\ A$

Proof. 1. Set $p\ _{:} \equiv \texttt{tt}$. Searchability then reduces to $(\sum x : A.\texttt{tt} = \texttt{tt}) + \prod x : A.\texttt{ff} = \texttt{tt}$, which is equivalent to $A + \neg A$.

2. Trivial.

3. From the contraposition of (2) using the fact that, given univalence, excluded middle for all types does not hold (see chapter 2 of the HoTTbook for this; TODO – give precise theorem number).

4. Both sides of the equivalence are propositions, so it suffices to give two implications. Right-to-left is the same as in point (2). Left-to-right: for a proposition P the type $\sum x : P.p\ x = \texttt{tt}$ is a proposition, so we can check if it's inhabited using LEM. If it is, we found what we we're looking for. If not, we can dervie $\prod x : P.p\ x = \texttt{ff}$.

5. Both sides are propositions, so we need two implications. Right-to-left: as in (4). Left-to-right: using LEM we can decide if $\exists x : A.p\ x = \texttt{tt}$ and if not, we can conclude $\forall x : A.p\ x = \texttt{ff}$ using classical logic. ☐

Having seen some instances of searchable types, we should be a bit dissatisfied: all types we considered we're sets and all type formers we saw preserved the property of being a set. What about types with more complicated path structure?

Theorem 1.6. The circle $\mathbb{S}^1$ is searchable.

Proof. We have $p : \mathbb{S}^1 \to \mathbf{2}$ and we need to find an $x : \mathbb{S}^1$ such that $p\ x = \texttt{tt}$ or prove that no $x : \mathbb{S}^1$ satisfies p.

Let's check the point base. If $p\ \texttt{base} = \texttt{tt}$, then we found our point. If not, it should be quite obvious that $\prod x : \mathbb{S}^1.p\ x = \texttt{ff}$, because all points are connected to base with a path. This path is truncated (i.e. $\prod x : \mathbb{S}^1.||x = \texttt{base}||$), but this doesn't matter – we can take it out because we're proving $p\ x = \texttt{ff}$ which is a proposition. ☐

The above proof looks a bit suspicious: there is nothing particular about the circle in it. It could just as well work for any type that has a point connected to all other points... such types are called connected (HoTTBook, remark 3.11.2).

$\texttt{Connected}\ A :\equiv \sum x : A. \prod y : A.||x = y||$

Theorem 1.7. Searchability of connected types.

1. $\prod A : \mathcal{U}.\texttt{Connected}\ A \to \texttt{Searchable}\ A$

2. Conclusion: the circle $\mathbb{S}^1$, the sphere $\mathbb{S}^2$, the n-sphere $\mathbb{S}^n$ are searchable.

2.5. There is a searchable type that is not connected.

3. If A is searchable, then it may be the case that the searchability of $x =_A y$ is taboo.

4. If A is inhabited, then the suspension $\Sigma\ A$ is searchable. If not, then it's also searchable.

Proof. 1. A is connected so we have $x : A$ and $H : \prod y : A.||x = y||$. Given a predicate $p : A \to \mathbf{2}$, let's check $p\ x$. If it's $\texttt{tt}$, then we're done. If not, use the path unpacked from $H\ y$ to show that all $y : A$ satisfy $p\ y = p\ x$ and thus $p\ y = \texttt{ff}$.

2. These types are connected.

2.5. $\mathbf{2}$ is one such type.

3. $\mathbb{S}^1$ is searchable, but it's fundamental group is $\mathbb{Z}$, whose searchability is taboo (just like for naturals).

4. If A is inhabited, $\Sigma\ A$ is connected and thus searchable. If A is empty, then $\Sigma\ A \simeq \mathbf{2}$, so it's also searchable. □

A question: can we prove that the suspension of A is always searchable, no matter what A is? Maybe we can use truncations for that.

Theorem 1.8. Searchability of truncations.

1. If A is searchable, $||A||$ is searchable.

2. The converse need not hold.

Proof. 1. To search $||A||$ with $p : ||A|| \to \mathbf{2}$, search A with the predicate $q\ x :\equiv p\ |x|$.

2. $||\mathbb{N}||$ is trivially searchable, but searchability of $\mathbb{N}$ is taboo. □

The proof of theorem 1.8.1 should hint us at an obvious generalization (if the fact that images of searchable types are searchable didn't hint us at it yet).

$\texttt{isSurjective}\ f :\equiv \forall b : B.\exists a : A.f\ a = b$

If there's a surjection from A to B, it means A has "more" points than B. More geometrically we could say that the points of A cover the points of B.

Theorem 1.9. Searchability and surjections

If A is searchable and $f : A \to B$ is surjective, then B is searchable.

Proof. If we want to search B with $p : B \to \mathbf{2}$, we can search A with $q\ a :\equiv p\ (f\ a)$. If we found an a, then we know that $f\ a$ is the element of B we're looking for. Otherwise for any $b : B$ we have an a such that $f\ a = b$, so from $\forall a : A.p\ (f\ a) = \texttt{ff}$ we can conclude $\forall b : B.p\ b = \texttt{ff}$.

□

We are now ready to prove that suspension is always searchable.

Theorem 1.10. Suspension is searchable

1. There is a surjection $\mathbf{2} \to \Sigma\ A$.
2. $\Sigma\ A$ is searchable.

Proof. 1. Let's define a function:

$$\begin{aligned} &f : \mathbf{2} \to \Sigma\ A \\ &f\ \texttt{tt} = \texttt{N} \\ &f\ \texttt{ff} = \texttt{S} \end{aligned}$$

It remains to prove that $\forall y : \Sigma\ A.\exists x : \mathbf{2}.f\ x = y$. We do this by induction on y:
If y is $\texttt{N}$, we choose $\texttt{tt}$.
If y is $\texttt{S}$, we choose $\texttt{ff}$.
If y varies along $\texttt{merid}\ a$ for some $a : A$, we need to give a path between proofs of $\exists$, but that's easy.

2. $\mathbf{2}$ is searchable and it covers $\Sigma\ A$, so $\Sigma\ A$ is searchable.

□

But there's more to it.

Theorem 1.11. Pushouts and pullbacks

1. If A and B are searchable, their pushout (for any $f : C \to A$ and $g : C \to B$) is searchable.

2. If A and B are searchable and C has decidable equality, then the pullback of $f : A \to C$ and $g : B \to C$ is searchable.

Proof. 1. The pushout is covered by the coproduct $A + B$.

2. To search the pushout with the predicate p, we may search $\prod AB$ with the predicate

$$q : A \times B \to \mathbf{2}$$
$$q\ (a, b) :\equiv p\ (a, b)\ \texttt{and}\ f\ a = g\ b$$

If we find some (a, b), we know it's ok, because if must satisfy $f\ a = g\ b$. If we didn't find anything, it means all (a, b) with $f\ a = g\ b$ fail to satisfy p, and these are all elements of the pullback.

□

2 Separatedness