Review for ESOP2024 of Finiteness in Cubical Type Theory

November 3, 2023

1 Summary

The reviewed article presents a library formalising mathematical results about finiteness in Cubical Agda. Cubical Agda is a proof assistant rooted in Homotopy Type Theory (abbreviated HoTT), a field in which dependent types are studied from a homotopical perspective and homotopy theory is studied from a type-theoretic perspective. The main features of Cubical Agda for this paper are that it implements the univalence axiom (stating that equivalent types are equal) as well propositional truncations (allowing to assert or enforce that no constructive information can be extracted from an inhabitant of a type). The library presented in the article contains the following:

- It contains formalisations of various results about five notions of finiteness, relating them to each other and giving several equivalent definitions for each. The various ways in which they differ from each other, due to both constructiveness and truncatedness, are laid out. While some of the formalised results are known, the following two notable results are new:
  - Theorem 2.7 proves that a type merely equivalent to $\mathrm{Fin}(n)$ (they are called cardinally finite type here) with a total order is in fact equivalent to $\mathrm{Fin}(n)$.
  - Theorem 3.4 proves that cardinally finite types form a Π-pretopos. A Π-pretopos is a standard analogue for topos in a predicative context.
- It contains a tool automating reasoning on cardinally finite sets designed around the fact that they are stable by dependent sums/products and interact well with decidable propositions. This tool is demonstrated using the countdown problem, which consists in finding all arithmetic expressions build from some finite set of numbers evaluating to a target number.

2 Review

The paper is well-written and easily understandable. It is a thought provoking and enjoyable experience to read it. Its scientific merits are as follows, in decreasing order of importance:

1. First and foremost, it provides a formalised library of results about finite sets, with an emphasis on cardinally finite sets. This is clearly valuable as finite sets are ubiquitous in mathematics, so it is bound to be used in many following developments. A lot of these results are nicely packed in Theorem 3.4, stating that cardinally finite sets forms a Π-pretopos.

2. Its focus on cardinally finite sets is well-justified in two ways. Firstly we think it is the most natural definition for finiteness in a homotopical context. Secondly it has a computational flavour, leading to the presentation of a tool for automated reasoning about cardinally finite sets. It is clear to us that this tool will be useful as it allows to turn many boring exhaustive proofs into one-liners, at least for small examples.

3. We think this paper is a good introduction to finiteness in a HoTT context, giving a nice and clear overview of how some natural finiteness notions relates to each other. Using HoTT (in particular propositional truncation) has the nice advantage that it allows to clearly distinguish issues related to information leaking out of constructive finiteness proofs from issues related to the potential failures of classical principles.

4. This paper claims that it can serve as a good introduction to HoTT (mainly univalence and propositional truncation) to functional programmers. While we are not in a position to assert the validity of this claim as we are already familiar with HoTT, we find it believable since functional programmers are likely to already have a good intuitive understanding of finite types.

With this in mind we recommend to accept the paper, assuming the minor corrections from the next section are implemented.

3 Corrections

While the proofs in the paper are perfectly rigorous, the intuitive explanations sometimes implicitly rely on the law of excluded middle for propositions. This should not be done in a constructive context, at least not without explicitly mentioning that LEM is used. The clearest example is line 333 where it is claimed that any proposition is isomorphic to $\top$ or $\bot$. See Section 5 for more details.

The introduction wrongly claims that Kuratowski finite sets form a Π-pretopos, whereas the article proves that cardinally finite types forms a Π-pretopos. This should be corrected. This is a typo rather than a mathematical mistake, but its a very misleading typo. By the way we think that Kuratowski finite sets forming a Π-pretopos is equivalent to the law of the excluded middle, see Section 4.1 for more info.

Some idea of how much time the proof search algorithm takes should be given, for both trivial examples like associativity of the conjunction of booleans and the running example of countdown problem.

4 Suggestions for improvements

Fig 1 is great! Since all notions of X being finite in the paper correspond to having a map from $\mathrm{Fin}(n)$ to X with certain property (sometimes as data and sometimes by asserting their mere existence), I think it would be very helpful to give a table summarising this information as a complement to Fig 1.

The proof of Theorem 2.7 is super nice! I think it can be extended for any X to an equivalence between the types (Cardinal finiteness X + total order on X) and (Manifest Bishop finiteness of X). Do this require a lot of work to formalise? If not it would be a nice addition!

I think it should be mentioned that manifest enumerable sets are precisely quotients of finite sets by any (non necessarily decidable) equivalence relation. See Section 4.3 for more thought on that.

The circle being finite seems counterintuitive to me, as its identity types are provably infinite. Even worst the delooping BG for G any infinite group is manifestly enumerable, so calling this a finiteness condition is a bit misleading to me. Maybe point that out as a remark, or restrict finiteness to sets.

4.1 If Kuratowski finite sets form a Π-pretopos then LEM holds

I think that Kuratowski finite sets forming a Π-pretopos (or even simply forming a lex catgeory) implies that all propositions are decidable (and therefore that a type is Kuratowski finite if and only if it is cardinally finite).

Indeed, assuming this, we would be able to find pullbacks of diagrams of the shape $1 \to X \leftarrow 1$ in the category of Kuratowski finite sets. Then identity types of a Kuratowski finite set would have to be Kuratowski finite sets.

But for any proposition P we have that the suspension ΣP is a Kuratowski finite set (via the inclusion of the two poles $1 + 1 \to \Sigma P$), and $N =_{\Sigma P} S$ is equivalent to P so then any proposition would be Kuratowski finite. But we can prove that any Kuratowski finite proposition is decidable, indeed it is merely equivalent to $\| \mathrm{Fin}(n) \|$ for some $n : \mathbb{N}$, and therefore merely equivalent to $n > 0$ for some $n : \mathbb{N}$, so it is decidable.

I think that if the author(s) agree with this proof it should be included somewhere in the article, as a subsection or as a remark.

4.2 Cardinally finite types form a Boolean topos

We think that monomorphisms in the category of cardinally finite types are precisely morphisms which fibers are decidable propositions.

Indeed it is clear that a monomorphism is injective so since it is a map between sets we know that its fibers are propositions. We know this fibers are cardinally finite type using stability of cardinally finite types by dependent sums. But cardinally finite propositions are decidable.

So monomorphisms in the category of cardinally finite sets are classified by *Bool* which is equal to $1 + 1$. This together with it being a Π-pretopos should imply that it is Boolean topos more or less by definition.

We don't think that formalising this would require a lot of work (we might be wrong...), and if so we think it would be a lovely addition to the paper. We think this at least warrant a remark, if the author(s) agree that our sketch of proof is correct.

By the way the topos of cardinally finite types validates choice, i.e. any epimorphism is split.

4.3 Toward a genuinely constructive notion of finite sets

This section contains suggestions for future work rather than suggestions for the submitted article, I hope it interests the author(s).

We think that the paper does a wonderful job at convincing that being cardinally finite is a suitable definition of finiteness, in particular it can be leveraged to automate proofs. We think that split enumerable, manifest enumerable and manifest Bishop finite should be avoided in a HoTT context as notions of finiteness in the sense that they do not have the correct automorphisms, for example the category of manifest Bishop finite sets is equivalent to $\mathbb{N}$ seen as a discrete category, which is not what we expect as a category of finite sets.

More generally it seems reasonable to us that a 'good' notion of finiteness should be a proposition on types such that:

- The type $\mathrm{Fin}(n)$ should be finite for all $n : \mathbb{N}$.
- Finite types should form a Π-pretopos.
- Some non-triviality assumptions like $\mathbb{N}$ should not be finite, and maybe some others.
- In a classical context they are equivalent to the usual finite types, meaning that LEM (or perhaps choice) should imply that being finite for this 'good' notion is equivalent to being cardinally finite.

With this in mind the paper demonstrates that being cardinally finite is a 'good' notion of finiteness. But as explained in 4.2 this notion is classical in the sense that cardinally finite types form a Boolean topos. A natural question to us is whether there is a 'good' notion of finiteness that is genuinely non-classical? As an example is there a 'good' notion of finiteness such that every proposition is finite? This would at least guarantee that if LEM fails then this notion is different from cardinal finiteness.

We have argued in Section 4.1 that Kuratowski finiteness is not good in the sense that it does not give a Π-pretopos.

One might think of considering the dual notion to Kuratowski finiteness, i.e. subfinite types, i.e. types that embeds into $\mathrm{Fin}(n)$ for some $n : \mathbb{N}$. Do they form a Π-pretopos? We have that all propositions are subfinite (subfinite types are in fact precisely the types merely of the form $\Sigma\mathrm{Fin}(n)P$ with P a family of propositions). But all subfinite type have decidable equality so we do not expect them to be stable by quotient by non-decidable equivalence relation, so they probably do not form a Π-pretopos (more precisely I expect that if subfinite type form a Π-pretopos then LEM holds and subfinite is equivalent to to cardinally finite, with a proof similar to the one for Kuratowski finiteness).

My best guess is then to consider types that merely embeds in a quotient of some $\mathrm{Fin}(n)$ by any equivalence relation (i.e. types that merely embeds in a Kuratowski finite set). Are they stable by dependent products? Do they form a Π-pretopos? They seems to obey all the other requirement for being a 'good' notion of finiteness.

5 Small corrections and suggestions for improvement

Line 79: This paragraph is confusing, my guess is that it means that any finiteness predicate has two versions: a list based and a natural number based, and that these two versions are always equivalent. It should be clarified.

Line 86: The article prove that cardinally finite sets forms a Π-pretopos, not Kuratowski finite sets. This should be corrected.

Line 160: We find the informal description "Lists are datatypes [...] its shape" of Lists as containers confusing. Maybe say "An inhabitant of $List(A)$ consists of a shape given by a natural number a an inhabitant of A for each number smaller than its shape".

Line 204: Should replace $[false, true]$ by $(2, [0 \mapsto false, 1 \mapsto true])$, using the container definition, or even better mention that the usual list notations is used to represent container-based lists.

Line 250-258: In HoTT, a contraction is an inhabitant of $isContr(A)$. A contractible type is a type with a contraction. So contraction $\to$ contractible types.

Line 320: I do not know what enumeration-based and relation-based mean. Maybe it should be explained, or the remark should be erased.

Line 327: I think the problem is not that we have many representative for each finite set as these representatives are all equal. The problem is rather that the category of manifest bishop finite type has no non-trivial automorphism, and therefore does not model finite sets in the usual mathematical sense. In fact it models $\mathbb{N}$ seen as a discrete category.

Line 333: Propositions are not equivalent to $\perp$ or $\top$ without the excluded middle for propositon, which is not assumed here.

Line 360: I think the do-notation should be explained a bit.

Line 373-378: It is incorrect to say that $\neg\neg P$ has no algorithmic content, see for example CPS translations. With LEM we have that $\neg\neg P$ and $\| P \|$ are equivalent so their difference is more in how they relate to classical principles. I would suppress this paragraph, or perhaps replace by a remark explaining that $\neg\neg P$ and $\| P \|$ have different computational meaning.

Line 388: $x = y$ being a proposition implies $Dec(x = y)$ being a proposition is not completely trivial. The way it is formulated here might be confusing to beginners.

Line 439: If it is mentioned that it is explored elsewhere, a reference should be given.

Line 488: It is wrong to say that cardinal finite and manifest Bishop finite refer to the same class of types, as I think you can find models where they do not agree. We have $\neg(CA \wedge \neg BA)$ so we cannot build an explicit counterexample, but we still cannot conclude $CA \leftrightarrow BA$. Might be worth pointing out that $CA \leftrightarrow BA$ is equivalent to the choice principle saying that any cardinal finite type have a total order.

Line 513: 'It has finitely many points' I do not know what this means. Maybe replace it by 'it is a finite cell complex'?

Line 589: 'is something of the standard formal definition of finiteness.' I do not know why this is claimed, especially since cardinal finite seems to me as the standard definition for finiteness in a HoTT context.

Line 606: giving a definition of Π-pretopos and some intuition about them (roughly: they are predicative toposes) would be welcome. Perhaps it should be mentionned that the notion of topoi is wrong in a constructive context, in the sense that Grothendieck topoi are not topoi constructively, so using Π-pretopos is rather natural.

Line 608: discrete Kuratowaski finite type → cardinal finite type. This seems to me as a recurring problem, meaning that discrete Kuratowski sets should be replaced by cardinal finite sets everywhere including in the introduction (which by the way wrongly claim a proof that Kuratowski finite sets form a Π-pretopos).

Line 618: Should indicate which proofs in this section are new.

Line 625: Should say that no informal account for sets forming a Π-pretopos will be given, and refer to the formalisation for more details.

Line 631: 'Since sets in HoTT [...] then the closure proofs.' I think this sentence sentence is confusing and should be suppressed.

Line 634: I think the subsection should be given a more informative title like 'Cardinal finite types form a Π-pretopos' or something like this.

Line 638: Should maybe explain whether using split enumerability rather than manifest Bishop finiteness leads to easier proofs.

Line 648: I do not know the do notation so I don't understand the definition of $\sup -\Sigma$. While I can infer what it is from context, maybe the do notation should be explained a bit.

Line 691: $\| \_ \|$ being a monad is not enough, you need it to be a modality. A weaker sufficient condition is: since $\| \_ \|$ is a functor commuting with finite products.

Line 729: I might be a bit confused by what you mean by monad, as I don't know Haskell. To me a monad do not have to commute with finite products (e.g.

the exception monad sends X to $X + 1$ and does not commute with product). However propositional truncation definitely does commute with products, as it is a modality, so this is not a problem here.

Line 768: ‘First and foremost, our proof of finiteness [...] the same cardinality’ I am not sure what this means. Perhaps it should be reformulated.

Line 807: should be explained more, ! as well as $\stackrel{?}{=}$ and $from - true$ are not defined.

Line 923: Should explain that not all expressions are valid because substraction and division are not total functions from natural numbers to natural numbers.

Line 1120: Seems worthwhile to give the precise definition of $\mathcal{E}!\langle Solutions \rangle$.

6 Typos

General: The symbol $\exists$ in HoTT is usually reserved for the propositional truncation of a sigma type, i.e. $\exists[x : A]B$ is defined as $\| \; \Sigma[x : A]B \; \|$. The paper seems to use both interchangeably, which is confusing to me. I suggest removing the uses of exists.

General: contraction $\rightarrow$ contractible types almost everywhere. A contraction of A is an inhabitant of $isContr(A)$, a contractible type is a type with a contraction.

General: to say that the propositional truncation of A in inhabited, people usually says that we merely have A. As an example a cardinal finite type could be be defined as a type X such that we merely have $n : \mathbb{N}$ and $\mathrm{Fin}(n) = X$.

General: Citation should all look the same, they are sometimes [name date], sometimes name [date].

General: Section 4.4 always says finite without specifying which notion is used. I think it does not matter much, but it should be specified either by replacing everywhere and stating in the beginning of the section what finite means here (which is split enumerablility if I understand well).

Line 6: Kuratowski finiteness form $\rightarrow$ Kuratowski finite types form

Line 13: finiteness in constructive $\rightarrow$ defining finiteness in constructive

Line 62: is quite a complex $\rightarrow$ is a quite complex

Line 62: it behaves something like $\rightarrow$ it behaves somewhat like

Line 63: it is more general $\rightarrow$ it does not necessarily satisfies excluded middle (or something similar, more precise)

Line 63: finite sets $\rightarrow$ Kuratowski finite sets

Line 64: demonstrate the library with $\rightarrow$ apply the library to

Line 76: (from cardinal finiteness to $\rightarrow$ (cardinal finiteness implying

Line 77: avoid nested parenthesis

Line 93: would not function $\rightarrow$ would not compute

Line 124: Manifest enumerability $\rightarrow$ manifest enumerability

Line 124: Section 2.X $\rightarrow$ (Section 2.X)

Line 165: Should use Sigma rather than Exists to define fibers.

Line 191: The syntax $\equiv \langle\rangle$ do not occur in the above proof.

Line 206: A slightly more complex example → Another example (I think it is even simpler)

Line 207: we have to provide → it is enough to provide

Line 242: I have never seen the word 'slop'. Maybe 'slop' → superfluous information.

Line 255: it must be unique → the fiber must have a unique inhabitant

Line 258: that x is not just in xs, but it → that not only x is in xs, but moreover that it

Line 272: turns out this is not the case → turns out that this is not the case

Line 273: Both predicates imply the other. → Each predicate implies the other.

Line 274: from manifest Bishop finiteness → from manifest Bishop finiteness to split enumerability

Line 276: $! \in$ is defined as a contraction of $\in$ → $x! \in xs$ is defined as the type of contraction of $x \in xs$.

Line 282: decidable equality, via Lemma → decidable equality via Lemma

Line 297: they are of equal power. → they have the same propositional truncation. (Or maybe something else?)

Line 307: $\exists \to \Sigma$.

Line 319: $\exists \to \Sigma$.

Line 329: a proof of finiteness that is a proposition. → a notion of finiteness such that being finite is a proposition.

Line 332: mere propositions → propositons

Line 332: contractions → contractible types

Line 335: types whose path are contractions. → types whose equality types are contractible.

Line 348: the first constructor here ($|_|$), or → the constructor $|_|$ here or

Line 349: "point" → point

Line 354: some A exists, without revealing which A → some inhabitant of A exists, without choosing one.

Line 362: there exists a propositionally truncated proof that A is → there exists a proof of the propositional truncation of A being

Line 389: all paths are propositions → all identity type are propositions

Line 410: Exists → Sigma

Line 419: Exists → Sigma

Line 424: that that → that this

Line 426: Exists → Sigma

Line 430: truncated equivalence proof → proof of truncated equivalence

Line 430: truncated proofs are trivially equal by the truncation, → proofs of a truncated type are always equal,

Line 431: now has been reduced → has now been reduced

Line 449: thereby satisfying the eliminator in Equation 12. → so that we can apply the eliminator in Equation 12 to conclude.

Line 454-456: enforcing that the given list is sorted, → Sorted is a predicate stating that the given list is sorted. The type $x \leftrightarrow y$ states that x is a permutation of y, the precise definition comes from [...]: two lists are related if they have equivalent membership predicates.

Line 457: Clarify that we use actual equivalences and not just functions back and forth, as an example [a,a] is not a permutation of [a].

Line 459: with our finiteness predicates. → with our definition of lists.

Line 460: extremely straightforward → straightforward

Line 467: are both permutations of each other they→ are permutations of each other, then they

Line 471: support lists of BA are permutations of each other this is → support lists of inhabitants of BA are permutation of each other, this is

Line 472: can eliminate from within a truncation. → induces a map out of the truncation.

Line 473: The second component of the output pair [...] definitions of permutations. → Now we need to provide the second component of the output pair, namely to prove that for all x in the cardinal finite type we have that x being in the sorted list is contractible. This follows from the fact that the sorted list is a permutation of the original one.

Line 482: the question arises → a question arises

Line 484: $\neg\Sigma[A : Type]CA \times \neg BA \rightarrow \neg(\Sigma[A : Type]CA \times \neg BA)$

Line 493: apply to the same types. → cannot be distinguished through explicit examples.

Line 501: contractions → contractible types

Line 503: path are members → path types are members

Line 509: the extra path → an extra path

Line 524: position information → information

Line 525: position is → information is

Line 535: was the listed form of a → gives a

Line 536: higher homotopies than sets → higher homotopies

Line 537: a surjection that is not necessarily → surjections that are not necessarily

Line 542: were split surjections → corresponds to split surjections

Line 543: to the proper surjections → to surjections

Line 547: trivially easy → easy

Line 553: decidable, because of the presence of decidable equality. → decidable when the type has decidable equality.

Line 586: itself commutative → is commutative

Line 591: equivalent → related

Line 591: seen already. → already seen.

Line 599: there the → their

Line 601: Lemma 2.10 → Lemma 2.10 and Lemma 2.9

Line 613: categories that avoids issues with higher homotopy structures. → categories with higher homotopy structure on their types of arrows.

Line 614: pre-categories, i.e. categories such that all arrows are sets. → 1-categories, i.e. categories which arrow types are sets.

Line 616: The formal proof we provide [. . .] our own. → Some proofs in this section are formalisation of proofs from [Univalent, section 9] as well as from [Iversen] and [Hu and Carette].

Line 620: we will follow → we followed

Line 622: the latter of which → the latter

Line 623: treatment (and definition) → treatment

Line 630: This will be quite similar to our objects for finite sets. → Our objects for the category of finite sets are defined similarly.

Line 635: discrete Kuratowski sets → cardinal finite sets

Line 655: makes use directly → makes directly use

Line 656: makes use furthermore → makes furthermore use

Line 662: type family from A → type family over A

Line 667: E!(Fin(n)) can be removed.

Line 682: made up of [. . .] which is also split enumerable. → a product of finitely many split enumerable type.

Line 684: Lemma 3.1 shows [. . .] proving our goal. → Applying Lemma 3.1 n times allows to conclude.

Line 688: closure proofs on B → closure proofs for B

Line 692: However [. . .] same trick does not work → However the same trick does not work for the dependent type formers Pi and Sigma.

Line 712: the following function exists: → there exists a function of the following type:

Line 715: on finite types → for cardinally finite types

Line 716: finite sets → cardinally finite types

Line 728: is closed under finite products → commutes with finite products

Line 731: ‘This lemma is a well-known folklore theorem.’ Should be moved to line 715, before the lemma.

Line 733: With the closure properties proven [. . .] complete: → It is explained in the formalisation how to go from these closure properties to the following theorem:

Line 734: Decidable Kuratowski sets → Cardinally finite sets

Line 738: interesting theoretical things → interesting theoretical result

Line 738: in some way related to that thing → related to that result.

Line 744: I like the righteous indignation. This is indeed unacceptable ;)

Line 748: can be leveraged to provide → have been leveraged to provide

Line 747: “logic programming”-esque → logic programming flavoured

Line 749: Using the machinery [. . .] proof of its correctness. → The machinery presented in this section will moreover accompany its solutions with formal proofs of their correctness.

Line 754: “more complex statement” → more complex statement

Line 755: problem, which we have been using throughout: → problem:

Line 760: the associativity → the previously mentioned associativity

Line 776: useful of which → usefulness of this example

Line 792: type of proofs which can be “reduce completely”: decidable proofs → proof obligations which can reduce completely: decidable proof obligations

Line 794: decision procedure over → decision procedure for

Line 811: needed of us → needed from us

Line 824: Exists → Sigma

Line 905: that that type → that this type

Line 906: "solution" → solution

Line 916: later on for this problem. → for this problem later on.

Line 921: that that → that this

Line 928: assist us to write → assist us in writing

Line 977: construct decidable equality for functions between Fin(n)s → prove that equality of endofunction of Fin(n) is decidable:

Line 978: such functions are themselves finite. → the type of such functions is finite.

Line 979: that the isomorphism, and by extension the permutation, is finite: → that the type of isomorphism of Fin(n) is finite:

Line 994: in the neighborhood of $O(n^n)$ → around $O(n^n)$

Line 996: blazing fast → blazingly fast (and I'm not sure if this is sarcastic or not, after all $O(n!)$ is big but there are $n!$ permutations so we cannot really hope to go faster)

Line 1015: the following: → an expression looking like:

Line 1020-1023: I don't think giving examples proving that parenthesising change the value of an expression have to be given, as it is a fact everyone reading this paper will be familiar with.

Line 1025: That's what we will [...] prove finite. → We will figure what this data types is and prove it is finite.

Line 1028: This doesn't get us much closer [...] rely on Dyck words. → However it is not convenient to prove that the type of such trees is finite, so we will rather use Dyck words.

Line 1048: Dyck words look easier [...] than straight binary trees, → the type of Dyck words looks easier to prove finite than the type of binary trees.

Line 1049: Dyck words are → the type of Dyck words is

Line 1049: to binary trees → to the type of binary trees

Line 1050: show this relation in one direction: from Dyck to → give a function in one direction: from Dyck words to

Line 1051: To demonstrate the algorithm → To define the algorithm

Line 1051: a simple tree definition: → a simple definition of binary trees:

Line 1090: we should not [...] not divide evenly → we should not be able to divide or substract when the result is not a well-defined natural number.

Line 1092: solutions: those which evaluates to the target, in other words. → solutions, in other words those which evaluates to the target.

Line 1093: these things → these constraints

Line 1093: expression like so: → expression:

Line 1109: With this all together, → With all if this together,

Line 1128: this gave → it gave

Line 1129: the elimination is coherently → the function is coherently

Line 1142: they are used

Line 1150: those of → those for

Line 1150: surjections, on Kuratowski finite sets. → surjections.

Line 1151: those sets form a topos → these sets form a Π-pretopos (and its good because there is no such proofs, see remark)

Line 1153: would not be possible. → is not possible in their context.

Line 1155: finite sets form a topos → cardinally finite sets form a Π-pretopos

Line 1156: quite closely along with the structure → quite closely the structure

Line 1158: finite sets → cardinally finite sets

Line 1162: category of finite sets → category of cardinally finite sets

Line 1159: the proofs there → their proofs

Line 1167: (or exhaustibility), without → (and exhaustibility) without

Line 1172: types, proof perspective. → types, formal perspective.

Line 1178: dependently-typed programming (and Agda in particular) → dependently-typed programming in general (and Agda in particular)

Line 1179: There, also, partial proof search is examined. → Partial proof search is also examined there.

Line 1193: "finite" → finite

Line 1194: of those predicates to the finite predicates. → between those predicates and the finiteness predicates.