Finite basis property revisited

Mizuhito Ogawa[1]

Japan Advanced Institute of Science and Technology
1-1 Asahidai Nomi Ishikawa, 923-1292 Japan
mizuhito@jaist.ac.jp

We discuss differences between the well quasi ordering (WQO) and the finite basis property of an ideal. A quasi ordering $\leq$ is a WQO, if, for an infinite sequence $a_1, a_2, \cdots$, there exists i, j with $i < j$ and $a_i \leq a_j$. A quasi ordering $\leq$ has the finite basis property, if, each ideal (upward closed set) I has finitely many minors (i.e., the minimal elements $b_1, \cdots, b_n$ in I such that, for each $c \in I$, there exists b_i with $b_i \leq c$). Classically they are equivalent, but constructively not. The former is described as a Π^0_2 formula, whereas the latter requires deeper nesting of quantifiers. For Π^2_0 formulae, we can apply A-translations and open / update / bar inductions [11, 1, 3] to extract computational contents from classical proofs with dependent choice [4, 2, 14, 12]. Typical examples are classical proofs of Dickson's and Higman's lemmas [6, 8]. Higman's lemma has several inductive proofs [5, 7, 13], which enable us program extractions. They are essentially the reduction to an exhaustive finite (but huge) case analysis.

However, program extraction from proofs of the finite basis property is not direct, since A-translation does not work anymore. An example is a linear time algorithm extraction of monadic disjunctive query processing on certain temporal database, which is based on the finite basis property by Higman's lemma [10].

This presentation returns to the constructive proof by Murthy and Russell [9], which gives an over approximation of a next candidate to continue a bad sequence in terms of regular expressions. Then, with the classical proof of well-foundedness of an ordering on regular expressions, we show termination of extraction procedures. This suggests a computable proof rather than a constructive proof. We will discuss about the difference starting from Dickson's lemma, followed by Higman's lemma, and possible generalizations.

References

1. U. Berger. A computational interpretation of open induction. In *Proc. 19th IEEE Symposium on Logic in Computer Science*, pages 326–334, 2004.
2. U. Berger, W. Buchholz, and H. Schwichtenberg. Refined program extraction from classical proofs. *Annals of Pure and Applied Logic*, 114:3–25, 2002.
3. U. Berger and P. Oliva. Modified bar recursion. *Mathematical Structures in Computer Science*, 16(2):163–183, 2006.
4. U. Berger, H. Schwichtenberg, and M. Seisenberger. The warshall algorithm and dickson's lemma: Two examples of realistic program extraction. *Journal of Automated Reasoning*, 26(2):205–221, 2001.
5. T. Coquand and D. Fridlender. A proof of Higman's lemma by structural induction. available at `ftp://ftp.cs.chalmers.se/pub/users/coquand/open1.ps.Z`, 1993.

6. L.E. Dickson. Finiteness of the odd perfect and primitive abundant numebrs with n distinct prime factors. *Amer. J. Math.*, 35, 1913.
7. A. Geser. A proof of Higman's lemma by open induction. Technical Report MIP-9606, Passau University, April 1996.
8. G. Higman. Ordering by divisibility in abstract algebras. *Proc. London Mathematical Society*, 2:326–336, 1952.
9. C.R. Murthy and J.R. Russell. A constructive proof of Higman's lemma. In *Proc. 5th IEEE Symposium on Logic in Computer Science*, pages 257–267, 1990.
10. M. Ogawa. A linear time algorithm for monadic querying of indefinite data over linearly ordered domains. *Information and Computation*, 186(2):236–259, 2003.
11. J.-C. Raoult. Proving open properties by induction. *Information Processing Letters*, 29:19–23, 1988.
12. H. Schwichtenberg and S. Wainer. Extracting computational content from proofs. In *Proofs and Computations*, Perspectives in Mathematical Logic. Springer-Verlag, *coming soon*.
13. M. Seisenberger. An inductive version of nash-williams' minimal-bad-sequence argument for higman's lemma. In *Types for Proofs and Programs, TYPES'00*, pages 233–242, 2000. Springer Lecture Notes in Computer Science, Vol. 2277.
14. M. Seisenberger. Programs from proofs using classical dependent choice. *Annals of Pure and Applied Logic*, 153(1–3):97–110, 2008.