Greedy Algorithms for Finding Maximum Number of Then Step in Reasoning*

Karol Pąk[1] and Aleksy Schubert[2]

[1] Institute of Informatics, University of Białystok,
ul. K. Ciolkowskiego 1M, 15-245 Bialystok, Poland
pakkarol@uwb.edu.pl

[2] Institute of Informatics, University of Warsaw
ul. S. Banacha 2, 02–097 Warsaw, Poland
alx@mimuw.edu.pl

Abstract

As in the case of computer programs, proof scripts written according to declarative formats of proof assistants such as Mizar or< Isabelle can sometimes be more and sometimes be less readable. However, quite frequently, smart pieces of reasoning that are conducted in a less readable way are re-analyzed by human readers to make them stronger, more easily applicable and so on. Unfortunately, the existing optimization methods that try to improve the representation of the information flow in a given reasoning are NP-complete. We present an algorithm that improves proof readability in accordance with Behaghel's First Law. We show that our algorithm achieves satisfactory performance ratio, which results from the fact that the maximal degree of the proof graph generated for a reasoning is small.

Formal proof checking systems such as Mizar[1] or Isabelle/Isar[5] can verify the correctness of proof scripts regardless of their legibility. However, in the case of human readers, the legibility has substantial significance. Generally, shorter reasoning should be more readable. However, if we shorten a given proof script, using the proof construction schemes available in the proof assistant at hand or external tool developed with system in the Mizar case, the operation may strongly impair readability.

As an orthogonal direction, we can improve proof legibility, reshaping the linearization of information flow in a given reasoning to better fit into human short-term memory. This manipulation is based upon the observation that every premise used in a justification has to be assumed or derived before in the proof; however the particular location of the premise in a reasoning is not fixed. The Behaghel's law prescribes that elements intellectually close one another should be located in the text next to each other. This idea is realized when we keep the rule that for each step at least part of required premises is available in n directly preceding steps of the proof script, or just in the previous step where we can refer to using the *then* construction. It is important to note that the problem of finding the maximum number of such steps is NP-hard, for each positive n [4].

There have been several attempts to build a tool that can automatically enhance a given linearization of an information flow. We can develop a transformation method where the information flow and a selected determinant of legibility are together formulated as a solving goal for an SMT solver [3]. In an alternative approach we can create dedicated algorithms or adapt existing ones to realize multi-criteria optimization of determinants. However, our problems of legibility are relatively new and not enough explored.

We propose an algorithm that maximizes the number of *then* steps, one of possible legibility determinants. In our approach the main goal is obtained based on an arbitrary heuristics that

*The paper has been financed by the resources of the Polish National Science Centre granted by decision no DEC-2012/07/N/ST6/02147

approximates the optimal solution of the Maximal Independent Set problem (MIS). Note that the MIS problem is well explored and there exists algorithms that have a performance ratio of $\frac{\Delta+2}{3}$, $\frac{2\overline{d}+3}{5}$ for approximating independent sets in graphs with degree bounded by Δ and the average degree $\overline{d}$, respectively [2].

To sketch the idea of our algorithm, let R be a reasoning and denote by F_R the DAG that represent the information flow in R. A topological sorting τ of F_R is a τ-linearization of R if τ describes the linear ordering of proof steps in R. A fragment of a reasoning is a *τ-reasoning path* when it is a sequence of steps such that all steps excluding at most the starting one are *then*-steps in τ. Additionally, τ-reasoning path is maximal, means simply that it is not a subsequence of any other τ-reasoning path. Note that the number of *then*-steps in a given τ-linearization determines uniquely an acyclic partition of the information flow for maximal τ-reasoning path of reasoning. Moreover the number of non *then*-steps in τ is equal to the length of the maximal τ-reasoning path. Consequently we can view the problem to maximize the number of *then*-steps as the problem to minimize the acyclic partition of a graph into directed paths.

The main idea of our algorithm is that we concatenate pairs of directed paths that fulfill chosen criteria keeping as invariant the acyclicity of the resulting partition.

We start the computation of the algorithm with a partition π_0 of the graph into singleton sets. Subsequently, we generate in an iteration i a set V_i of pairs (a, b) of elements in the partition π_{i-1} such that the first step in b uses the last element of a as its premise, but at the same time joining the two elements of the partition does not introduce cycle (equivalently, the longest path that connects the joined elements in the partition graph has length 1). Then we construct an instance of the MIS problem based on V_i. We use V_i as the vertices in the input graph and its elements are connected with an edge when corresponding pairs of elements in π_{i-1} have common element (pairs as sets have non-empty intersection). The independent set of vertices that results from an MIS algorithm indicates the set of those pairs in π_{i-1} that are joined to obtain π_i. We continue this iteration until the set V_i gets empty.

It is worth mentioning here that after the first iteration we obtain at most every second edge in each *then*-sequence. If we use the known minimum degree-greedy algorithm on top of the above mentioned procedure, we obtain the maximal number of *then*-steps in 93.97% of one-level reasonings in MML, which offers a much satisfactory compromise between the complexity of the procedure and the quality of the final result.

References

[1] G. Bancerek, C. Bylinski, A. Grabowski, A. Korniłowicz, R. Matuszewski, A. Naumowicz, K. Pąk, and J. Urban. Mizar: State-of-the-art and Beyond. In M. Kerber, J. Carette, C. Kaliszyk, F. Rabe, and V. Sorge, editors, *CICM 2015, Washington, DC, USA, July 13-17, 2015, Proceedings*, volume 9150 of *LNCS*, pages 261–279. Springer, 2015.

[2] M. M. Hallorsson and J. Radhakrishnan. Greed is Good: Approximating Independent Sets in Sparse and Bounded-degree Graphs. *Algorithmica*, 18(1):145–163, 1997.

[3] K. Pąk. Automated Improving of Proof Legibility in the Mizar System. In S. M. Watt, J. H. Davenport, A. P. Sexton, P. Sojka, and J. Urban, editors, *CICM 2014, Coimbra, Portugal, July 7-11, 2014. Proceedings*, volume 8543 of *LNCS*, pages 373–387. Springer, 2014.

[4] K. Pąk. Improving Legibility of Formal Proofs Based on the Close Reference Principle is NP-Hard. *Journal of Automated Reasoning*, 55(3):295–306, 2015.

[5] Makarius Wenzel. The Isabelle/Isar Reference Manual, 2015.