Review of Bachelor's Thesis

Student: Malecová Tatiana
Title: Equivalence-Based Slicing of Programs (id 24038)
Reviewer: Vojnar Tomáš, prof. Ing., Ph.D., DITS FIT BUT

1. Assignment complexity — **more demanding assignment**

The difficulty of the work is slightly above average due to combining a need to work with low-level systems code, compiler technology, as well as methods of static analysis (which are not lectured at the bachelor level at FIT BUT). Moreover, the proposed approach had to extend a recent research work, and the implementation had to be combined with a so-far prototype tool.

2. Completeness of assignment requirements — **assignment fulfilled**

The assignment has been fully accomplished.

3. Length of technical report — **in usual extent**

The extent of the thesis is in the usual range.

4. Presentation level of technical report — **75 p. (C)**

The overall structure of the thesis is very logical, the contents of the particular chapters is appropriate, the chapters link well together. I appreciate that the author helps the reader to get oriented by systematically providing summaries of so far described ideas and outlines what is going to be presented next. Most of the text is quite well readable and clear. I appreciate that the author presents the core of the proposed approach in the form of pseudocode.

However, I have also hit several parts that were harder to follow for me and that possibly contain some smaller mistakes:

1. Page 7: The different sets like L_i, G_i, etc. should be required to be pairwise disjoint.
2. Page 10, Alg. 1, line 9: Q is not a couple, perhaps you should initialize it as <(...,...)>.
3. Page 10, Alg. 1, line 13: It is not clear to me what happens with the control flow after the check on this line (and I think that this point is important for the termination of the algorithm).
4. Page 13, last line: It is not clear to me why you require E' to be a strict subset of E.
5. Page 17, Alg. 2, line 2: A similar problem with the type of the assigned element as already mentioned above.
6. Page 17, Alg. 2, line 8: I do not see what the newly synchronised basic blocks are.
7. Page 18, point (c): At this point, it was not clear to me how the synchronisation is found.
8. Page 19, point 3: When mapping so far unmapped operands to each other, you create a sort of assumption on the code (you assume that the mapping is established), do not you? From the text, I do not see why this is safe, i.e., why you can afford to do so.
9. Page 20, Alg. 4, line 10: A missing bracket.
10. Page 20, Alg. 4, lines 12 and 13: I do not understand what the effect of these statements is when one passes several times through them. They seem to overwrite the results of each other.
11. Page 20, Alg. 4: I am not sure what the algorithm returns when no difference is detected.
12. Page 21, Alg. 5, lines 20 and 21: I do not see what happens if elements of Q1 and/or Q2 have some unprocessed successors. They may be missed by the algorithm. Indeed, what would happened if the node number 7 in the right part of Figure 4.2 had some further successors?

5. Formal aspects of technical report — **90 p. (A)**

As far as I can say, the work is written in very good English. The text contains just some typos and minor language errors (e.g., in the use of articles, some words are used in a wrong way, etc.). The typography is at a quite high level too with just some minor problems (e.g., the placement of labels of tables).

6. Literature usage — **90 p. (A)**

The way the student used the available literature is appropriate.

7. Implementation results — **90 p. (A)**

The approach proposed by the student has been implemented as an extension of the open-source tool DiffKemp. At the time of writing of this report, the code has not yet been merged into the main branch of DiffKemp, but---to the best of my knowledge---it is not far from that. I have seen the tool in action, and it indeed works as expected. The student evaluated the tool through a number of experiments that validate its correct functioning and efficiency.

What I am somewhat missing is an evaluation of the results produced by the tool in such a way that some heavy-weight tool for checking semantic equivalence would be tested on them. I have to stress that an experiment of this form is not an explicit part of the assignment. However, since the work on the tool is motivated by that it should help applicability of heavy-weight formal tools, it would be nice to try that out on at least some small examples.

8. Utilizability of results

The work contributes to the recently developed tool DiffKemp for checking sematic equivalence of different versions of systems code. DiffKemp was developed in collaboration of Red Hat and FIT BUT. The fact that Red Hat has invested into the project shows that the subject is important for the company. However, DiffKemp is still subject to further improvements aimed at its better practical applicability. The original approach proposed in the thesis and its implementation represent important steps in this direction.

9. Questions for defence

1. On page 9 of your thesis, you say you simplified the presentation wrt the original algorithm used by DiffKemp by considering an instruction-by-instruction comparison only. Is it really a simplification of the presentation only? This is, does your tool support dealing with more general comparisons than instruction-by-instruction?
2. Have you ever tried at least a simple experiment with applying some heavy-weight formal verifier, such as LLReve, on the code simplified by you? You mention that you reduced the changed code by 54.8 % overall. I am afraid that this needs still not be enough for the heavy weight tools to succeed. However, perhaps you saw some (many?) particular functions reduced much more so that LLReve would indeed catch up? May be this could even be tried for the defence?
3. If there is enough time, you could comment on some of the issues mentioned in the part of the review devoted to the writing of your thesis. Particularly, points 12, 8, 3, and 10 are interesting from my point of view (in this order).

10. Total assessment

87 p. very good (B)

My evaluation of the work, which is close to the upper edge of B, summarizes the facts that the student came with an original and non-trivial solution, the proposed solution is implemented and evaluated in a solid way, the thesis is written in very good English, the thesis is mostly quite well readable and correct, but I have found some parts which seem to contain smaller problems. In case the student presents the work in an excellent way and is able to react to my comments and questions in a clear way (indeed, I may have misunderstood something), the evaluation could be even better according to my opinion.

In Brno 2 June 2021

Vojnar Tomáš, prof. Ing., Ph.D.
reviewer