

**I. IDENTIFICATION DATA**

<b>Thesis name:</b>	Memory Safety Analysis in Rust GCC
<b>Author's name:</b>	Jakub Dupák
<b>Type of thesis :</b>	master
<b>Faculty/Institute:</b>	Faculty of Electrical Engineering (FEE)
<b>Department:</b>	Department of Measurement
<b>Thesis reviewer:</b>	Jan Hubika
<b>Reviewer's department:</b>	Department of Applied Mathematics, Charles University

**II. EVALUATION OF INDIVIDUAL CRITERIA**

**Assignment**

**extraordinary**

*Evaluation of thesis difficulty of assignment.*

The goal of thesis is to implement borrow checking to Rust front-end in GCC (which is still in early stages of development and borrow checking is among the most complex missing parts of the infrastructure). This represents several challenges including the need to gain in-depth understanding of Rust language, GCC architecture and intermediate languages (GIMPLE and GENERIC), intermediate languages used in Rust front-ends and LLVM. This by itself exceeds usual requirements made on master thesis. In addition to that student needed to solve problems of interfacing C++ implementation of Rust frontend to Polonius solver implemented in Rust. The problem of borrow checking itself can be seen as a variation of alias analysis (or a points-to analysis) which is quite difficult part of every production quality compiler. Polonius itself is still in development and uses algorithms that differs significantly from current implementation of borrow checking in the official Rust compiler.

**Satisfaction of assignment**

**fulfilled**

*Assess that handed thesis meets assignment. Present points of assignment that fell short or were extended. Try to assess importance, impact or cause of each shortcoming.*

Student implemented working prototype of borrow checker. Over 60 of his patches has been already accepted to official GCC release. Limitations of current implementation of borrow checking are discussed in Section 6.1 of the thesis and seems very reasonable.

**Method of conception**

**outstanding**

*Assess that student has chosen correct approach or solution methods.*

The main technical challenges involved deciding on intermediate language on which GCC Rust should interact with the Polonius checker. In one direction borrow checking needs the control flow graph, which is constructed after translating parsed program to GIMPLE representation. In the opposite direction however borrow checking is tightly tied with details of Rust programming language which are not represented in GIMPLE (that is programming language agnostic representation designed for code optimization). The decision taken to implement a lowering step to new intermediate language BIR seems long-term maintainable. Decisions were discussed with senior GCC developers and it seems that the newly implemented borrow checking will be accepted to mainline compiler during development period of GCC 15.

**Technical level**

**A - excellent**

*Assess level of thesis specialty, use of knowledge gained by study and by expert literature, use of sources and data gained by experience.*

As discussed above, I consider the topic to be technically challenging and students ability to provide a working prototype is certainly a success.

**Formal and language level, scope of thesis**

**B - very good**

*Assess correctness of usage of formal notation. Assess typographical and language arrangement of thesis.*

Thesis is written clearly and documents all main aspects of the problem at hand. Formatting of references (with no space between sentence and reference) is somewhat non-standard. Author also sometimes omits to mention what he is referring to "After the initial experiments described in 5.1" instead of "After the initial experiments described in Section 5.1". Similarly the numbers of chapters in introduction have off-by-one error. Those are however minor issues.

**Selection of sources, citation correctness**

**A - excellent**

*Present your opinion to student's activity when obtaining and using study materials for thesis creation. Characterize selection of sources. Assess that student used all relevant sources. Verify that all used elements are correctly distinguished from own results and thoughts. Assess that citation ethics has not been breached and that all bibliographic citations are complete and in accordance with citation convention and standards.*

I noticed no problems in this area. All references seems correctly cited.

**Additional commentary and evaluation**

*Present your opinion to achieved primary goals of thesis, e.g. level of theoretical results, level and functionality of technical or software conception, publication performance, experimental dexterity etc.*

In addition to 60 patches accepted to GCC source-base, Jakub Dupák also authored a technical report "Contribution to the Rust front-end for the GCC compiler".

**III. OVERALL EVALUATION, QUESTIONS FOR DEFENSE, CLASSIFICATION SUGGESTION**

*Jakub Dupák implemented major part of new GCC Rust front-end and his work is from part already accepted into official GCC source-code. I believe that his thesis is of exceptional quality and clearly demonstrates his ability to solve challenging technical problems and communicate with large community of developers.*

*I would like to ask is why it was decided that Rust front-end in GCC may not use features of C++ language beyond C++11. While C++ dialect used by the middle-end and C++ front-end is limited to make it possible to build GCC in platforms with old compilers, the front-ends generally can use different languages (such as Ada FE which is implemented in Ada).*

I evaluate handed thesis with classification grade **A - excellent**.

Date: 4.2.2024

Signature:

