Validation of Stimuli for Studying Mental Representations Formed by Parallel Programmers During Parallel Program Comprehension

Leah Bidlake **Eric Aubanel** **Daniel Voyer**
Faculty of Computer Science, Faculty of Computer Science, Department of Psychology
University of New Brunswick
leah.bidlake@unb.ca, aubanel@unb.ca, voyer@unb.ca

Abstract

Research on mental representations formed by programmers during program comprehension has not yet been applied to parallel programming. The goal of this proposed study is to validate the stimuli that will be used in subsequent studies on mental representations formed by expert parallel programmers. The task used to stimulate the comprehension process will be verifying the correctness of parallel programs by determining the presence of data races. Responses to the data race question will be analysed to determine the validity of the stimuli. Participants will also be asked what components of the program they used to determine whether or not there was a data race and their responses will be collected for use in future work.

1. Introduction

In recent years, the research on program comprehension has declined dramatically and as a result, newly developed or popularized languages and paradigms including parallel programming have not been a part of the research (Bidlake, Aubanel, & Voyer, 2020). Parallel programming has introduced new challenges including bugs that are hard to detect, making it difficult for programmers to verify correctness of code. One type of bug that occurs in parallel programming is data races. Data races occur when multiple threads of execution access the same memory location without controlling the order of the accesses and at least one of the memory accesses is a write (Liao, Lin, Asplund, Schordan, & Karlin, 2017). Depending on the order of the accesses some threads may read the memory location before the write and others may read the memory location after the write which can lead to unpredictable results and incorrect program execution. Data races are difficult to detect and verify as they will not appear every time that the program is executed. To detect data races programmers must understand how a program executes in parallel on the machine and the memory model of the programming language.

2. Research Goal

To date, no empirical research on program comprehension or mental representations of parallel programmers has been conducted. The lack of research in this programming paradigm means that there are no existing data sets or stimuli to draw from. We will create a stimulus set to be used in subsequent research on mental representations formed by parallel programmers. The research goal of the proposed study is to validate the stimulus set.

3. Method

Given the feedback received when this proposal was presented at the PPIG Doctoral Consortium 2020, we will be conducting a pilot study with 10 participants. The results of the pilot will be analysed to determine if any of the parameters need to be adjusted for the validation study.

3.1. Participants

Participants will need to have experience programming in C and using OpenMP 4.0 directives to implement parallelization. One hundred participants will be recruited using social media including Facebook and LinkedIn and email. Participants will receive a $10 gift card as an incentive; this will be administered using Rybbon. Participants will be informed in the consent form that the incentive is only available in select countries. Participation will be voluntary and the protocol will be approved by the research ethics board at UNB.

3.2. Materials

The programs from the DataRaceBench 1.2.0 benchmark suite (Liao et al., 2017) were used as inspiration for the programs written by the first two authors, who are expert programmers. The programs will be written in C using OpenMP 4.0 directives with no comments or documentation. There will be 80 programs in total, all containing a parallel region. In our original study proposal the programs were divided into data race and no data race categories and then subdivided into reading input from a file and reading input from the command line. The feedback received from the doctoral consortium led us to simplify the programs so that all data was contained within the program. Forty of the programs will contain a data race and forty of the programs will not contain a data race. This will produce 4000 data points per data race (40 programs x 100 participants), exceeding the power recommendation proposed by Brysbaert and Stevens (2018) for our linear mixed model analysis.

The length of the programs will be measured by counting the lines of code. The mean length of the programs with data races will be the same as the mean length of the programs without data races.

3.3. Procedure

Participants will complete the tasks online. The experiment will be developed using PsychoPy 3, an open source software package, and Pavlovia will be used to host the experiment online. Qualtrics will be used to develop and administer the consent form at the beginning of the experiment and the questionnaire at the end of the experiment.

The stimuli will be presented to each participant in a random order. Participants will be given a time limit of 30 seconds for exposure to the stimuli. The exposure will end when the time limit has been reached or when the participant responds to the data race question. There were concerns raised at the doctoral consortium regarding the exposure time and mental strain of tracing code. To address these concerns we have ensured that variable names used in the stimuli match typical programming conventions (i.e.: variables i, j, and k are used for loop counters) and were consistent between stimuli to reduce the mental load (i.e.: variables used for arrays in all stimuli are a, b, and c). The stimuli were also simplified by removing the code to read in data from either a file or the command line.

The following measures will be taken for each stimulus: correctness of response, response time, and level of confidence in their response. Their level of confidence will be measured using a visual analogue scale representing a 100mm line with "not confident" as the left side extreme (score of 0) and "very confident" as the right side extreme (score of 100). Participants will click at the location of their answer and the program will record the distance along the line (0 to 100) as the measure of confidence.

Originally we proposed to ask a summary question for a subset of the stimuli. It was suggested at the doctoral consortium that a summary may not provide insight into the mental model of the participants. Instead, questions that specifically elicit how participants are thinking about the code, what parts of the code they are looking at, or what parts of the code are most relevant for the task, would provide more information that relates to their mental representations. In response to this we decided to ask the question "What cues or program components did you use to determine whether or not there was a data race?" instead of writing a summary. Thirty of the stimuli will include the additional task of answering this question. The participants will not be given a time limit for writing their answer. The question will take place after the data race question and the level of confidence rating are completed. After finishing the data race experiment, participants will complete a questionnaire to document their background and level of expertise (Feigenspan, Kastner, Liebig, Apel, & Hanenberg, 2012). Specifically, they will be asked about their: age, gender, year of study, level of education completed, years of programming experience, number of programming courses completed, self estimated level of programming expertise and parallel programming expertise, perceived level of programming expertise compared to their peers, and perceived level of parallel programming expertise compared to their peers.

3.4. Analysis

The results of the pilot study will be used to determine if any parameters of the study need to be adjusted. The exposure time and level of difficulty of the stimuli may need to be adjusted if the accuracy rate is

low and participants are using all of the allotted time exposure.

The participants' responses to the question for select stimuli will be subjected to an informal analysis as a preliminary examination of mental representations, however, the emphasis will be on stimulus validation. The responses to the data race task will be used to validate the stimulus set. Ideally we want to have an accuracy rate of approximately 90% for both data race types (yes, no). If the task is too difficult we expect there will be a higher number of no responses compared to yes responses to the data race question. We predict, with expertise as an independent variable, a positive correlation between level of expertise and confidence and a negative correlation with response time. The variables relevant to expertise will be used with the data race type (yes, no) as predictors in exploratory mixed linear models.

4. Conclusion

The results of the proposed study will indicate the validity of the stimulus set and provide direction for future studies. A valid stimulus set would allow us to move forward with our research on mental representations of expert parallel programmers.

5. References

Bidlake, L., Aubanel, E., & Voyer, D. (2020, July). Systematic literature review of empirical studies on mental representations of programs. *Journal of Systems and Software*, *165*, 110565. doi: 10.1016/j.jss.2020.110565

Brysbaert, M., & Stevens, M. (2018). Power analysis and effect size in mixed effects models: A tutorial. *Journal of Cognition*, *1*(1). doi: 10.5334/joc.10

Feigenspan, J., Kastner, C., Liebig, J., Apel, S., & Hanenberg, S. (2012, Jun). Measuring programming experience. In *20th IEEE International Conference on Program Comprehension (ICPC)* (p. 73–82). IEEE. doi: 10.1109/ICPC.2012.6240511

Liao, C., Lin, P.-H., Asplund, J., Schordan, M., & Karlin, I. (2017). Dataracebench: A benchmark suite for systematic evaluation of data race detection tools. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (p. 11:1–11:14). ACM. doi: 10.1145/3126908.3126958