Project Proposal

Justin Frank Pierce Darragh

Our project plan is to develop a new language built for the specification of the checkpointed style of computation commonly used in intermittent computing systems. While a fair amount of work has been done in this area, there does not yet seem to be a single work that combines all the features we intend to develop.

We are interested in this work because it is an approach at solving an architecture problem from a PL perspective. Specifically, we get to develop a new formal semantics, which is always a good time.

Background

Unfortunately, we only very recently changed topic direction and have not yet had time to collect a body of related works.

We can look at papers covered in class for examples of uses of related techniques. For example, "A Simpler, Safer Programming and Execution Model for Intermittent Systems" discusses many of the concerns around checkpointing, which will be very relevant to our interests. Meanwhile, the Crash Hoare Logic paper provides a framework for developing a logical system that we may wish to compare against. And "Revamping Hardware Persistency Models" shows an approach to formalizing hardware semantics that may prove useful to us.

Approach

Our project will be split into four primary components:

1. A new, small language based on IMP but with checkpoint operations.
2. A static analysis to identify sets of memory locations that need to be saved at checkpoints to maintain consistency with fully powered execution.
3. A formalization of the checkpoint semantics, derived from the CESK style.
4. A mechanization of the formal model of (3).

The language and the associated proofs will be implemented in Rocq, since both authors are already familiar with that system.

We will start with the development of both the core semantics as well as a preliminary approach to the static analysis. The semantics are necessary to hammer

out first to ensure we don't make any missteps in the implementation, and the design of the analysis will likely inform some elements of the semantics. The mechanized formalization will be developed in concert with the implementation of the language.

Additionally, we have begun speculation on the degree to which we can increase the complexity of the language as time permits. We will start with a small language first to ensure we can complete all of our goals by the end of the semester, but we would like to develop this system to be as fully featured as is reasonably possible.

Plan for Evaluation

The evaluation of our project can best be measured by the completion of its separate deliverables as outlined in the approach; namely, we intend to have (1) a language, (2) a static analysis of memory needs, (3) a semantics, and (4) a mechanized proof of the whole system.

A completely successful project would mean that all the proofs are accepted by Rocq, and that the necessary theorems are all covered. (This last part is a bit more up in the air, since we do not yet know the exact scope and nature of all the proofs we will need. We intend to stay in communication with the course instructor as needed to ensure we cover our bases.s)