Project Ideas for Verification of Reactive Systems

The goal of a project is to understand an area in more depth, to replicate known results, or to perform some small research project. Projects should be done in groups of 2 or 3. Each project will have a mentor. Here are some ideas for (more research oriented) projects. You are welcome to come up with an idea on your own, but please talk to course staff before you start to work on it.

1. "Open sourcing MSR". Microsoft Research has a large number of researchers working in verification and analysis of software. In recent years, they have worked on many interesting verification projects. Unfortunately, most of these projects have proprietary or very restrictive licenses. The goal of this class of projects is to look at MSR projects and developing open-source implementations based, e.g., on LLVM or Frama-C.
I can give a more comprehensive list, but here are a few starting points:

- MSR has a language called Boogie that is an intermediate representation for software. Implement a Boogie compiler and its verification condition generator (search for Boogie MSR to find more details about this project)
- Implement context-bounded reachability (see the CHESS project at MSR)
- Implement the Corral software analysis tool (see the Corral web page)

You can search the MSR web pages http://research.microsoft.com/en-us/groups/rise to get more examples

2-3 persons per project
Mentor: Johannes / Filip / rupak

2. Bounded partial order reduction:
Implement the algorithm from the OOPSLA 2013 paper of that name by Musuvathi et al. and evaluate it on some set of benchmarks.

Mentor: Susanne van den Elsen and Rupak Majumdar
Suitable for 2-3 persons

3. [Theoretical] There is a famous theorem in first-order logic that states that every first order logic formula that is closed under sub-models is equivalent to a universal first-order logic formula. There is a similar result for a temporal logic called the mu calculus. The goal of this project is to study the proof for the mu-calculus and come up with a bound on how large the equivalent formula can be.
Conjecture: the equivalent formula is at most doubly exponential in the size of the original formula.
Can you find a good lower bound?

Suitable for 2 persons but requires a strong theory background
Mentor: Rupak Majumdar and Filip Niksic

4. Implement a bounded model checker like CBMC for C programs over LLVM. There is already an implementation called LLBMC that does it. Unfortunately, it is not open source. Compare your implementation against CBMC and LLBMC.

Mentor: Rupak Majumdar and Min Gao
Suitable for 2-3 persons

5. Develop a mocking framework for Amazon web services.

Suppose you want to test code that uses AWS, e.g., by doing symbolic execution. Clearly, you do not want to connect to actual amazon services (each call can cost you money). What you should develop is a mocking framework that replaces the actual amazon calls to calls to your framework and integrate with a symbolic execution tool. You might be able to use already existing packages that do part of this.

Mentor: Rupak Majumdar
Suitable for 2-3 persons

6. There is a way to solve "exists-forall" formulas by reducing them to checking the number of solutions to a SAT formula. Implement this algorithm (on top of an existing tool to count the number of solutions of a SAT formula) and see how well it performs.

Mentor: Rupak Majumdar / Dmitry Chistikov / Rayna Dimitrova
Suitable for 2 persons

7. Uniform generation: In constrained random testing, one gives some constraints on the inputs and then randomly picks values from the set of valid inputs. There is a lot of work on generating random values uniformly from the set of satisfying assignments for Boolean formulas. The goal of this project is to do it for constraints in richer languages, such as those involving linear arithmetic.

Mentor: Rupak Majumdar / Rayna Dimitrova / Dmitry Chistikov
Suitable for 2-3 persons

8. Write a BDD package on top of a key-value store like redis

Mentor: Rupak Majumdar / Rayna
Suitable for 2-3 persons

9. In addition to these suggestions, please feel free to come up with your own ideas. I will augment this list of projects over the next few days. Also, if you would like to work on more theoretical topics, please talk to me.