
CS 639A Progress Report 1

Sumit Lahiri
19111274

Group No
1

Amit Kumar Sharma
20111012

First & Second Sprint : Brief Progress

We started working on the [debloating project](#) in **three splits** for our **Pre-RL Stage Work**. The most **time-consuming** part was to migrate the outdated packages used in some the tools chosen by us for delivering the project. We gained a fair idea on how **embedding spaces** are used and the novelty in [inst2vec](#) tool as modelling an LLVM IR as **XFG** & mapping to an **NLP** like **skip-gram** model. We started running the [inst2vec](#) tool to generate an embedding for our debloating **C++** sample space, like **call-site** information, **function arguments**, **type parameters** etc as described in [Paper 2](#). We are reporting the highlights from sprint of [Commit-1 01/11/2020](#) to [Commit-22 09/11/2020](#) which is **Week-1 : November 2020**.

Stage 1 : Split 1

- Completed the setup for LLVM-IR generation, clang tool, [inst2vec](#) tool.
- Completed the setup for [OCCAM](#) tool, [Trimmer](#) tool & [Chisel](#) tool. All setup & build related issues were **resolved** and **ran** once on one sample example.
- We are in the middle of running the [inst2vec](#) tool and using the **public data-sets** available to train it.
- We are using the pre-trained embeddings from [inst2vec](#) repository for the starting point for the **DeepOCCAM** tool we are developing.

Stage 1 : Split 2

- We found **2** more features where **program debloating** may be possible.
- We understood how to write specification for the script used by [Chisel](#) tool as in [Paper 1](#) in terms of the three parts of the scripts namely `compile()`, `desired()` & `undesired()`.
- [Chisel](#) tool works by learning a **policy** for **delta debugging** by **reinforcement learning** which guarantees **1-minimal P^*** & $O(|P|^2)$ runtime. The abstraction is a **markov decision process** for meaningful **guidance**.
- We spend time understanding how [OCCAM](#) tool, [Trimmer](#) tool & [Chisel](#) tool are working.

Stage 1 : Split 3

- Bloated Sample collection and preparation started, We are targeting [bitcoin1-src](#) & [eth-solidity](#). Past experience tells that the code is most probably **bloated**. We may have to change it later to publicly available samples.
- We found this [video](#) useful to understand how **program debloating** can be improved using **Stochastic Optimization**, which we intend to explore as **extra work** if time permits.