

Project Proposal

- A title reflecting the project
 - Javascript Fuzzing
- Team Members
Thomas Perez, Sarkis Mikaelian, Prabhat Gehlot, Eduardo Preciado
- Any background necessary to understand the project
 - Familiarity with Javascript semantics, and syntax.
 - Familiarity with Constraint Logic Programming, and an understanding of its inherent advantage for testing software systems.
 - Grammar, and how they are used to formal specify programming languages.
 - An understanding of prolog's syntactic, and semantic definition.
- What exactly you need to do for this project, along with a timeline of when you're planning to do it:

What	When
Find some specific functions(language elements) to test within javascript	(Same week)
Review existing literature for testing languages using CLP, and reviewing other language fuzzers like JSFuzz, and other tools.	3 weeks
Design, and implement a testing scheme.	6 weeks
Reviewing our results, and improving	9 weeks

- Find some specific functions (language elements) to test within javascript(Same week).
- Review existing literature for testing languages using CLP, and reviewing other language fuzzers like JSFuzz, and other tools.
- Design, and implement a testing scheme.
- Reviewing our results, and improving.

- The sort of data you're planning to collect for the project for the report. This can include things like:
 - Benchmarking data showing how fast it runs, or how complex of problems it can handle
 - Data comparing code complexity between programming techniques
 - Basically, anything interesting that can be learned by either looking at the implementation or running the implementation
 - We will collect how many interesting cases we make.
 - If our results are valid
 - Reasoning behind our choices
- An explanation of how the project relates to logic programming
 - This project relates to logic programming because it uses one of the major concepts of logic programming. Defining validity to generate programs that fit that validity or using outputs to determine inputs
 - Logic Programming is useful for testing
- Why this project is sufficiently large
 - This project is unlimited by definition of testing where we can test for days yet still not be able to prove validity. However, we can analyze the data collected from testing to view interesting cases.
 - Also if we feel that after our testing and analysis of some functionality has proven insufficient than we may go ahead and select another functionality and repeat our process.
 - This project is sufficiently large because using CLP we can test infinitely until we find an interesting test case.