**CPSC 311 – Term Project**

*Proposal: Logical Expression -> Logical Circuit (LETLC language)*

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Student # | CS ID | E-Mail |
| Zhe Li | 88792486 | d3q1b | lizhe918@alumni.ubc.ca |
| Maxwell Yang | 96245428 | r0g2b | shukany@alumni.ubc.ca |
| Eric Yan | 22548424 | g5h2b | ericy676@students.cs.ubc.ca |
|  |  |  |  |
|  |  |  |  |

The Logical Expression -> Logical Circuit language (or simply LETLC, pronounced “Litek”) is a domain specific language designed to autonomously generate logical circuit diagrams from user-provided logical expressions. To elaborate, the user can enter a logical expression in a concrete syntax similar to the Racket as a Racket program. The LETLC will then parse the given expression into an abstract syntax and proceed to interpret it into a corresponding circuit diagram. The Racket language and its pict library will be the foundations for the creation of this language. The primary motivation behind the development of this language is to provide a clearer visual representation of unfamiliar Boolean algebra and circuit design concepts for students. Both of which are heavily emphasized in lower level undergraduate Computer Science courses. The language may also be used by course staff to help the development of new exam/homework questions, as well as solutions to some logical statement/circuit design problems.

To realize the language, we will split it into multiple milestones and have specific goals for each one.

**Exact project to be determined**

1. **The Project Topic and Type**
2. The Project Topic
3. How the Project Belongs to the Proposed Type
4. What the Project Should be after Finished
5. **Plan for Subsequent Milestones.**

**References**