Pan Chan
CS9163
Assignment 1.1

Turing Completeness

Turing completeness is the ability to compute every Turing-computable function. In other words, something is Turing complete if it can simulate another Turing machine.

In the case of this sandbox, it is Turing complete because it can be compared to the MIPS architecture, which has been proven to be Turing complete in the past. This sandbox has an instruction set which is a subset of the MIPS' instruction set. For Turing completeness, this sandbox can perform every integer operation instruction in the MIPS architecture. Add, subtract, multiply and divide are supported for all signed integers. There is no immediate arithmetic function in this sandbox, like there is in MIPS, but this is merely an inconvenience, rather than a limitation of capabilities. Basic copying, setting, and clearing of registers and memory are supported. The MIPS architecture gets more specific with instructions for certain registers, half-words, bytes, etc, but this is not necessary in the sandbox as it is not actually running machine language. Logical functions in MIPS can all be simulated in this sandbox by using the equality instruction (EQL) and loading/storing in memory and registers. This sandbox also supports a conditional jump on equal or not equal, and an unconditional jump. All MIPS integer functions can be simulated in this sandbox, and MIPS can be proven to be Turing complete using its integer functions, thus this sandbox is Turing complete.