Jada Cumberland
Nov 7, 2025

Turing Completeness of Architectural Gadgets

In the course of conducting research into Weird Machines, the team at ODU utilized a Python library (https://github.com/rossgore/weird_machine_gadgets) created by faculty advisor Ross Gore to mimic the architectural gadgets available in cyber-physical systems. This example in particular models the Wind Energy Testbed (https://sics-c.org/cybersecurity-center-for-offshore-wind-energy/) at ODU's Center for Secure and Intelligent Critical Systems (SICS-C). With these recreated gadgets, we were able to pseudocode several programs that demonstrated the usage of typical gadgets and functions in additional ways, creating additional states in the intended Turing Machine. This, by definition, represents a weird machine. Our programs included a segmented and full message system, a semaphore, a counter, and more, as seen in the examples in the GitHub repository.

In the world of computers, the most powerful machine representation of a system is a Turing Machine. Turing machines must have the ability to read/write to an arbitrary amount of memory, make conditional decisions, and traverse/jump through the memory as necessary. This traversal is also a fundamental aspect of looping structures, so it could be said that the presence of loops is a property of a Turing-complete language, along with sequential execution and selection/conditionals.

We concluded that the pseudo gadgets in the library constitute a Turing Machine; this includes loops, integer variables, binary variables, and conditional writes. Programmers can "get" and "set" these variables, representing the read and write operations necessary to be considered Turing-complete. Additionally, different parts of the memory could be accessed based on the memory address provided to these functions, and loops make use of this function. They are implemented through the ControlGadget structure; as typical, they check a given value or expression to determine which code should be run/jumped to.

The availability of these gadgets in a cyber-physical system provides opportunities for additional features from simple operations. Since we can consider the weird machines to be Turing-complete, any problem could be solved using them. This also proves that a specifically crafted input could control the architectural gadgets present within a CPS and utilize them for positive or negative reasons.