Hackable FSM

Abstract
FSM (Finite State Automaton) is a primary computer science abstract, computer science would not be possible without it. It defines "state device", that can go through states via multiple "paths" in graph. This is done typically in hardware level, but it is possible to construct such "device" in higher level computer programming languages. This work shows, that in higher level languages it is possible to construct automaton, wich can be set to other states via other paths than those defined in automaton graph.

Hackable FSM:

1. Automaton in higher programming languages (JavaScript here) is defined as a set of states and a set of connections between states. It typically performs some action, when set to some state, when undergoing transition via connection.
2. So two arrays exist: States (nodes in graph) and connections (paths of length 1 between states of graph)
3. If the transition happens, automaton typically fires action.

... todo

4. This implementation shows, that it is possible to implement finite state automaton functionality in a way, that lets programmer transition from state to state using custom callback in a way, that is not defined in automaton definition, it allows programmer work with "hackable automaton".
5. Using hackable automaton, it's possible to create automaton graphs with principle called "test driven development", which is one of the key principles for functionality and security. Example would be transition to "nonsense" states via nonsense paths and check if automaton behaves still correctly or opposite (not correctly). This allows coverage of automaton with tests that are the proof of correctness.

Basic implementation is on http://ceneksvoboda.eu/mat.html

This „paper“ is not complete, it's todo (draw graphs, explain code,...)

8.3.2020
Čeněk mooph Svoboda