

## A9.1 - Model Checking

Jaan Tollander de Balsch - 452056

December 3, 2018

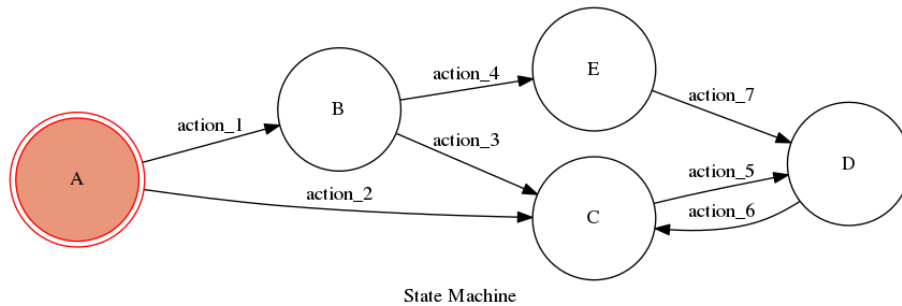


Figure 1: Example of a finite state machine.

The properties of given *finite state machine (FSM)* can analyzed by analyzing the underlying *directed graph*  $G = (V, E)$  where the vertices  $V$  are the *states* and the edges  $E$  are the *transitions*. The state machine starts from some initial state  $s \in V$ . A finite state machine can be implemented using Python library called *transitions*<sup>1</sup>. (Yarkoni, 2014)

**Weak task completeness:** *Can a user find some way to reach a goal from initialization?* This property can be tested by finding a path between initial state  $s$  and some goal state  $t$  using a graph search algorithm such as depth-first search or breadth-first search. (Mann, 2014)

**Visibility:** *User input is always associated with feedback.* This property can be tested by testing that the transition to all states yields some feedback. The feedback could be visual feedback or text output in the console. (ForeverWintr, 2016)

The attached code contains the test for these properties implemented in Python.

---

<sup>1</sup><https://github.com/pytransitions/transitions>

## References

ForeverWintr, 2016. *How to capture stdout output from a python function call? - stack overflow*. [online] Available at: <<https://stackoverflow.com/a/40984270/5433224>> [Accessed 3 Dec. 2018].

Mann, E., 2014. *Depth-first search and breadth-first search in python*. [online] Available at: <<https://eddmann.com/posts/depth-first-search-and-breadth-first-search-in-python/>> [Accessed 3 Dec. 2018].

Yarkoni, T., 2014. *Yet another python state machine (and why you might care)*. [online] Available at: <<https://www.talyarkoni.org/blog/2014/10/29/yet-another-python-state-machine-and-why-you-might-care/>> [Accessed 3 Dec. 2018].