

# A brief visit with the theory of automata

The first concept for an analytical framework for a system of lexicography is that of the automaton that walks the source text. In walking we may see reflections of a tape machine. An equivalence is made between the traversal of a tape of instructions, and the traversal of a sequence of characters. Both affect the performance of the traversal by the automaton, while the character of purpose is very different.

First, we need to interrogate the commonality as more interesting than the difference in either one replicating the other by abstraction. The loss of the distinction of mechanical detail has substance, but is it significant to a theory of automata?

Second, we may postpone that question to examine alternative models of lexicographical processors under the same cloud of thought.

The stateless lexicograph maps each character code, for example ASCII, to a state class. For example, “a” is an “alphabet” and “.” is a “special”. The stateless classifier walks a source text to produce a target

lexicograph most efficiently. Likewise, any stateless automaton traverses a process sequence in order.

At this point we may identify the intent of abstraction as the capture of commonality. The abstract automaton is inclusive of the common properties of any discreet, mechanical, stream processor. This is a frame of conception. The frame of analysis decorates the concept with mechanical detail sufficient to represent an object of study.

Therefore, the issues of commonality and comparison at the cloud were faulty by combination. A development proceeded the conception.

A stateful lexicograph may coalesce the represented state of subsequences, and may span the represented state of groups. For example, the subsequence “http://www/” may be coalesced to a “literal” despite having members from distinct classes, “alphabet” and “special”. Similarly, the group “(a < b)” may be spanned despite being a composition.

This automaton differs from an instruction processor in the decoration of its description, but in both cases we are concerned with issues we may term reachability and coverage. What problem spaces are entered and how are they covered? As designers and users we want to open problem spaces to awareness and knowledge, and to produce and use machines

that cover those problem spaces uniformly and effectively. That is, we want solutions that close onto corresponding problems crisply.

This is the purpose of the theory of the computing machine, the theory of automata.

John Pritchard

<https://twitter.com/syntelos>

<https://github.com/syntelos>

Saturday, 1 September 2018

<https://drive.google.com/file/d/1XXgxFSr0DlefZxbf1Pfn7wVsbLYnyTqg/view>

This work is licensed under a Creative Commons  
Attribution- NonCommercial- ShareAlike 4.0  
International License

<http://creativecommons.org/licenses/by-nc-sa/4.0/>

/

