Notes on Sipser: Chapter 8, “Space Complexity” (through § 8.2)
CSC 341, “Automata, Formal Languages, and Computational Complexity”
Department of Computer Science · Grinnell College
August 31, 2020

Just as a Turing machine $M$ that halts on every input has a time-complexity function $t(n)$ mapping every natural number $n$ to the maximum number of transitions that $M$ makes on any input of length $n$ before halting, so too $M$ has a space-complexity function $s(n)$ mapping every natural number $n$ to the maximum number of distinct squares that $M$’s tape head scans during the processing of any input of length $n$. (A technical point: If $M$ moves onto a previously unscanned square during the transition to $q_{\text{accept}}$ or $q_{\text{reject}}$, that square is not counted in the space complexity because $M$ never reads its contents. Similarly, the space complexity of a Turing machine that halts without making any transitions is 0, not 1.)

It’s clear that, for any such Turing machine $M$, $s(n) \leq t(n)$, since the tape head can reach at most one new square on any single transition.

The definitions of the $\text{SPACE}(f(n))$ and $\text{NSPACE}(f(n))$ classes are analogous to the definitions of $\text{TIME}(f(n))$ and $\text{NTIME}(f(n))$, but the space-complexity functions associated with particular languages are not necessarily even of the same order as their time-complexity functions.

In the proof of Savitch’s Theorem (Theorem 8.5), the number of configurations of $N$ is an upper bound on the number of transitions that $N$ makes before halting because any more transitions would result in a repeated configuration, and a Turing machine that repeats a configuration will never halt. This would contradict the assumption (in the first sentence of the proof) that $N$ is a decider, which means that it halts on every input. Space complexity and time complexity are both undefined for Turing machines that don’t always halt.

The remaining sections of Chapter 8 introduce some more complexity classes, defined in terms of the operation of automata similar to Turing machines but restricted in various ways, that are active research areas in theoretical computer science. This is one possible route for graduate study.

Copyright © 2020 John David Stone
This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit
http://creativecommons.org/licenses/by-sa/4.0/deed.en_US
or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.