CS 360
Fall 2020
Practical Applications of CS 360

Jeffrey Shallit
School of Computer Science
University of Waterloo
shallit@uwaterloo.ca
https://cs.uwaterloo.ca/~shallit
Efficient tokenization of computer languages: the first step in compilation is breaking the input up into tokens. The best methods rely on the theory of regular expressions.

Efficient compilation of computer languages: we want to write our computer programs in a high-level language and have them parsed automatically translated to machine code. The best methods for this rely on the theory of deterministic context-free languages and LR grammars.
String searching: more and more, we must search vast databases of text for data matching a given pattern, which can have wildcards and other distinguished symbols. Many string-searching methods are ultimately based on finite automata and regular expressions.

Recognizing difficult problems: in real life, one is sometimes confronted with the necessity of solving a problem that, in full generality, is undecidable. The theory of computation helps you recognize when the problem you want to solve has no general solution, and lets you look for alternatives to solve the problem (approximate solutions, special cases, etc.)
Applications to other areas: formal language theory has proven useful in many other fields, such as

- circuit verification
- model-checking (formal verification of finite-state concurrent systems)
- XML document checking
- economics and game theory (finite automata as strategy models in decision-making);
- theoretical biology (L-systems as models of organism growth; DNA as a string)
- computer graphics (L-systems)
- linguistics (modeling by grammars)