Compilers

CMPT 432

Lab 2

Goals	More token making, now with Finite Automata
Notes	We're still figuring out how the characters that comprise the source code get turned into tokens that are (hopefully) valid in our language. Now we're down to the token-making nitty-gritty: Regular Expressions and Finite Automata
Resources	*Crafting a Compiler* • Read chapter 3 again • Do exercises 3.3 and 3.4 (regular expressions and DFAs) *Dragon* • Read chapter 3 again • Do exercises 3.3.4 (case insensitivity in regular expressions) Have a look at this Regular Expression maker/tester online at https://www.debuggex.com
Submitting	Commit a PDF of your work to your GitHub repository and I'll take a look at it.

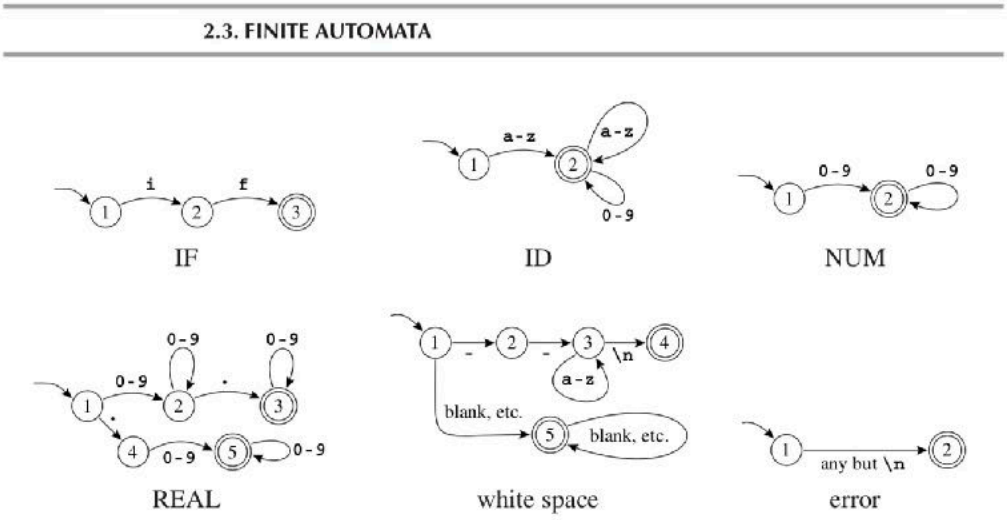

FIGURE 2.3. Finite automata for lexical tokens. The states are indicated by circles; final states are indicated by double circles. The start state has an arrow coming in from nowhere. An edge labeled with several characters is shorthand for many parallel edges.

from *Modern Compiler Implementation in Java* by Andrew Appel