

Theory of Computation, CSCI 438 spring 2022
Exam 3 review, April 13

New material is in yellow.

Regular Languages

- Know the definition of a DFA.
- Be able to convert:

Conversion	Using What
Picture \Rightarrow $M = (\dots)$, i.e. the formal definition	Definition
Picture \Leftarrow $M = (\dots)$	

- Be able to convert:

Language description \Rightarrow DFA	Problem solving
Language description \Leftarrow DFA	

- Know the signature of δ^* for a DFA and be able to use it to define what it means for a DFA to accept a string, and therefore for the DFA to recognize a language
- Know the definition of a regular language
- Be able to prove that regular languages are closed under complementation, union and intersection

Nondeterminism

- Know the definition of an NFA
- Be able to convert:

DFA \Rightarrow NFA	Theorem: A language is regular iff some NFA recognizes it
DFA \Leftarrow NFA	

- Be able to prove that a language is regular iff some NFA recognizes it.
- Be able to prove closure of regular languages under concatenation and star-closure

Regular Expressions

- Know the definition of regular expressions
- Know situations where regular expressions are used
- Be able to convert:

Language description \Rightarrow Reg-ex	Problem solving
Language description \Leftarrow Reg-ex	

- Be able to convert:

NFA \Rightarrow Reg-ex	Problem solving. Once you have a solution you could prove that your solution works via induction on the length of the string, but I won't ask you to do this.
NFA \Leftarrow Reg-ex	

- Know the pumping lemma for regular languages and be able to prove that a language is not regular using the pumping lemma for regular languages

Context-Free Languages

- Know the definition of a context-free grammar
- Know situations where context-free grammars are used
- Know what it means for a variable to be able to “**derive**” a string
- Know the definition of a context free language
- Know what it means for a grammar to be “**ambiguous**” and what it means for a language to be “**inherently ambiguous**”
- Be able to prove that context-free languages are closed under union, concatenation and star-closure
- Be able to convert:

Language description \Rightarrow CFG	Problem solving
Language description \Leftarrow CFG	

- Know the definition of a PDA.

- Be able to convert:

Language description \Rightarrow PDA	Problem solving
Language description \Leftarrow PDA	

- Know the format of a grammar which is in Chomsky Normal Form.

- Be able to convert:

CFG \Rightarrow CFG in Chomsky normal form	Theorem 2.29: Any context-free language is generated by a context-
-------------------------------------------------	--------------------------------------------------------------------

	free grammar in Chomsky normal form.
--	--------------------------------------

- Be able to convert:

PDA \Leftarrow CFG	Theorem 2.20: A language is context-free iff some PDA recognizes it. For both directions you only need to be able to do the conversion
----------------------	----------------------------------------------------------------------------------------------------------------------------------------

- Know the pumping lemma for context-free languages and be able to prove that a language is not context-free using the pumping lemma for context-free languages

Turing machines are describable at various levels:

- Know the definition of a TM
- Know what it means for a language to be Turing-decidable and Turing-recognizable.
- Be able to convert:

Language description \Rightarrow TM	Problem solving.
---------------------------------------	------------------

- Given a variation on the definition of a TM, be able to show that it computes the same class of languages as a TM (that is, it computes the class of Turing-recognizable languages).

Variation of a TM \Rightarrow TM	Use constructive proofs
Variation of a TM \Leftarrow TM	

(You should know that a non-deterministic TM can be converted into a regular TM and you can describe the proof from a high level, but I will not ask detailed questions on the proof.)

- Know the Church-Turing Thesis and its implications.
- Know what is meant by an algorithm and know the format in which we will be writing algorithms in this class.