University of Nevada, Las Vegas Computer Science 456/656 Fall 2021

Answers to Assignment 4. Due Wednesday October 6, 2021

There are more classes of languages (problems) than I can remember. Go to the internet if you are curious about this. But I expect you to learn only a few of them.

You will need to know following classes of languages (problems) by the end of the semester.

- $\mathcal{NC}$ (Nick's Class.)
- $\mathcal{P} = \mathcal{P}$-TIME.
- $\mathcal{NP} = \mathcal{NP}$-TIME.
- co-$\mathcal{NP}$.
- $\mathcal{P}$-SPACE.
- $\mathcal{NP}$-COMPLETE.
- $\mathcal{P}$-*complete*.
- $\mathcal{P}$-SPACE-*complete*.
- Recursive (decidable).
- Undecidable.
- $\mathcal{RE}$ (Recursively Enumerable.)
- co-$\mathcal{RE}$.
- Regular.
- Context-free.
- Context-sensitive.

You also will need to know inclusions, *i.e.* which classes are subclasses of others. Theoretically, you could draw a gigantic Euler diagram showing all inclusions, but it's better to draw Euler diagrams showing inclusions for only a few classes.

You will need to know about the following classes of grammars.

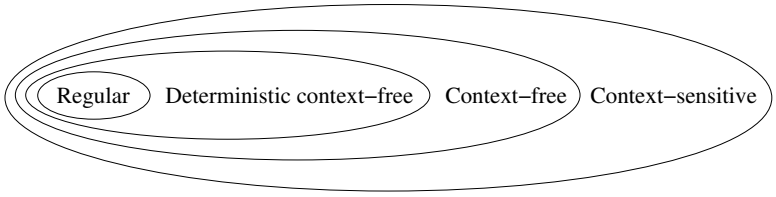
- Regular.
- Context-free.
- Context-sensitive.
- Unrestricted.

You will need to know the following concepts as well.

- Enumeration of a language.
- Recursive enumeration of a language.
- Canonical enumeration of a language.
- Recursive function.
- Countable, uncountable.

Note: *Recursive* means computable, but also means decidable. It is not the same as the definition used in programming.

1. Draw an Euler diagram showing the inclusions among the classes regular, context-free, deterministic context-free, and context-sensitive.



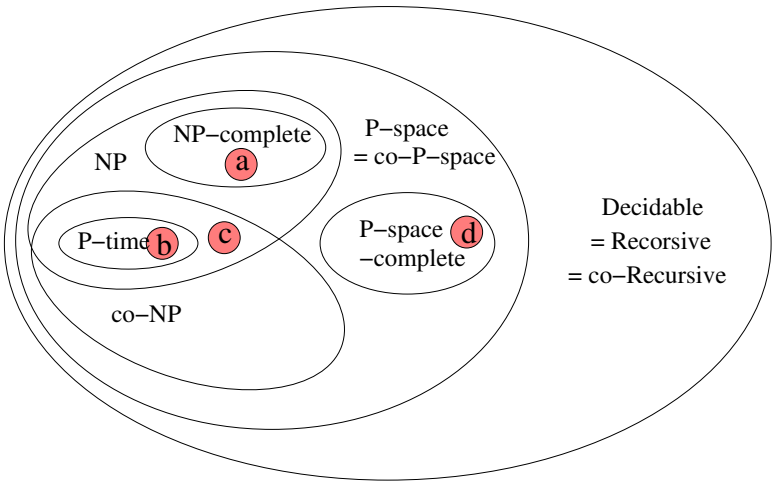
   As the semester goes on, I will expect you to be able to draw Euler diagrams for other classes.

2. Fill in each blank with either T(true), F(false), or O(open), showing which classes are closed under various operations.

Closed under:	union	intersection	concatenation	complementation	Kleene closure
Regular	T	T	T	T	T
Context-free	T	F	T	F	T
Context-sensitive	T	T	T	T	T
$\mathcal{P}$-TIME	T	T	T	T	T
$\mathcal{NP}$	T	T	T	O	T
$\mathcal{NP}$-complete	O	F	O	O	O
Decidable	T	T	T	T	T
Undecidable	F	F	T	F	F

   The website https://cs.stackexchange.com/questions/24264/are-np-complete-languages-closed-under-any-regular-operations gives proofs that $\mathcal{NP}$-complete is not closed under union, concatenation, and Kleene closure. These proof are only valid if $\mathcal{P} \neq \mathcal{NP}$, and the correct answer is "O" in all these cases. However, it can be proved that the class is not closed under intersection without assume that $\mathcal{P} \neq \mathcal{NP}$.

3. The following diagram shows inclusions of some of the complexity classes we've talked about.



Assuming that $\mathcal{P} \neq \mathcal{NP}$:

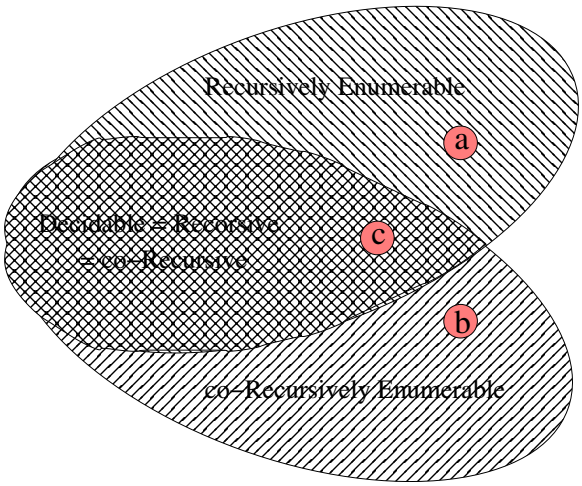
(a) Where in the diagram would you put the traveling salesman problem?

(b) Where in the diagram would you put the membership problem for a context-free language?

(c) Where in the diagram would you put the factoring problem for binary numerals?

(d) Where in the diagram would you put the equivalence problem for regular expressions?

The following diagram shows the relationship between recursive, recursively enumerable, and co-recursively enumerable languages.



(a) Where in the diagram would you put the halting problem?

(b) Where in the diagram would you put the equivalence problem for context-free grammars?

(c) Where in the diagram would you put the equivalence problem for regular expressions?