# CSC 339 – Theory of Computation
Fall 2023

## Tutorial 8
Decidability, Complexity

# Exercise 1

Show that the class of decidable languages is closed under complementation, union, intersection and concatenation.

# Solution

Suppose that $L_1$and $L_2$ are decidable languages. Let $M_1$ and $M_2$ be TM's that decide these languages.

TM that decides $\overline{L_1}$:

1. Run $M_1$ on the input.
2. If $M_1$ accepts, reject. If $M_1$ rejects, accept.

# Solution

Suppose that $L_1$and $L_2$ are decidable languages. Let $M_1$ and $M_2$ be TM's that decide these languages.

TM that decides $L_1 \cup L_2$:

1. Copy the input to a second tape.
2. Run $M_1$ on the first tape.
3. If $M_1$ accepts, accept.
4. Otherwise, run $M_2$ on the second tape.
5. If $M_2$ accepts, accept. Otherwise, reject.

# Solution

Suppose that $L_1$and $L_2$ are decidable languages. Let $M_1$ and $M_2$ be TM's that decide these languages.

TM that decides $L_1 \cap L_2$:

1. Copy the input to a second tape.
2. Run $M_1$ on the first tape.
3. If $M_1$ rejects, reject.
4. Otherwise, run $M_2$ on the second tape.
5. If $M_2$ accepts, accept. Otherwise, reject.

# Solution

Suppose that $L_1$ and $L_2$ are decidable languages. Let $M_1$ and $M_2$ be TM's that decide these languages.

TM that decides $L_1 L_2$:

1. If the input is empty, run $M_1$ on the first tape and $M_2$ on a blank second tape. If both accept, accept. Otherwise, reject.
2. Mark the first symbol of the input with #.
3. Copy the beginning of the input, up to but *not* including the symbol #, to a second tape. Copy the rest of the input to a third tape.
4. Run $M_1$ on the second tape and $M_2$ on the third tape.
5. If both accept, accept.
6. Otherwise, move the mark (#) to the next symbol of the input.
7. While the mark has not reached a blank space, repeat Steps 3 to 6.
8. Delete the mark from the first tape. Run $M_1$ on the first tape and $M_2$ on a blank second tape. If both accept, accept. Otherwise, reject.

# Exercise 2

Show that the acceptance problem for NFA's is decidable. The corresponding language $A_{NFA}$ is:

$$A_{NFA} = \left\{ \begin{array}{c} \langle N, w \rangle | N \text{ is an } NFA, w \text{ is a string over the input} \\ \text{alphabet of } N, \text{and } N \text{ accepts } w \end{array} \right\}$$

# Solution

1. Verify that the input string is of the form $\langle N, w \rangle$ where $N$ is an NFA and $w$ is a string over the input alphabet of $N$. If not, reject.
2. Convert $N$ into a DFA $M$.
3. Determine if $M$ accepts $w$ by using the algorithm for the acceptance problem for DFA's.
4. Accept if that algorithm accepts. Otherwise, reject.

# Exercise 3

Consider the language of strings of the form $\langle M \rangle$ where $M$ is a DFA that accepts at least one string of odd length. Show that this language is decidable.

# Solution

Let $L_{ODD}$ be the language of strings of odd length. If $M$ is a DFA, then $M$ accepts at least one string of odd length if and only if $L(M) \cap L_{ODD} \neq \emptyset$. The language $L_{ODD}$ is regular.

1. Verify that the input string is of the form $\langle M \rangle$ where $M$ is a DFA. If not, reject.
2. Construct a DFA $M'$ for the language $L(M) \cap L_{ODD}$.
3. Test if $L(M') = \emptyset$ by using the emptiness algorithm.
4. Reject if that algorithm accepts. Otherwise, accept.

# Exercise 4

Let $INFINITE_{DFA} =$
$\{\langle A \rangle |\ A$ is a $DFA$ and $L(A)$ is an infinite language$\}$.
Show that $INFINITE_{DFA}$ is decidable.

(see text textbook for a solution – page 213, ex. 4.10).

# Exercise 5

- Consider the TMs designed to recognize the languages defined in the tutorial 7. Determine their time and space complexity classes.