# Theory of Automata
# Properties of Regular Languages

Week 5

# Contents


- Closure Properties
- Complementation
- Intersection

# Closure Properties

- A language that can be defined by a regular expression is called a regular language.
- Not all languages are regular, as we shall see in the next lecture.
- In this lecture we will focus on the class of all regular languages and discuss some of their properties.

# Theorem 10

**If $L_1$ and $L_2$ are regular languages, then $L_1 + L_2$, $L_1L_2$, and $L^*_1$ are also regular languages.**

Notes:

- $L_1 + L_2$ is the language of all words in either $L_1$ or $L_2$.
- $L_1L_2$ is the product language of all words formed by concatenating a word form $L_1$ with a word from $L_2$.
- $L_1$ is the language of all words that are the concatenation of arbitrarily many factors from $L_1$*.
- The result stated in Theorem 10 is often expressed as **"The set of regular languages is *closed* under union, concatenation, and Kleene closure".**

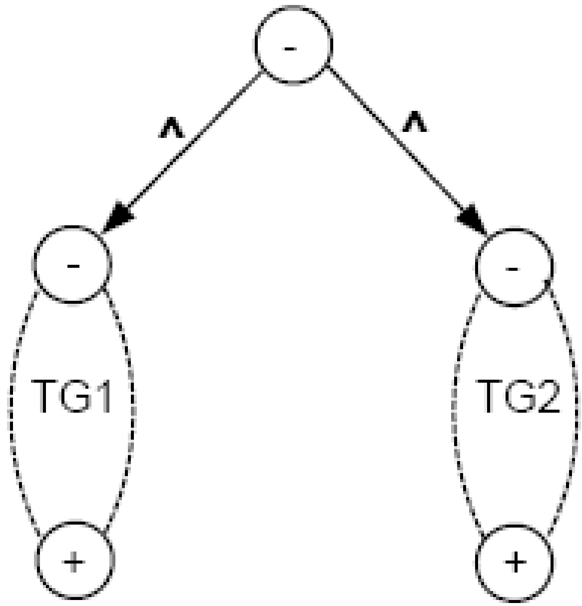
# Proof by Machines

- Because $L_1$ and $L_2$ are regular languages, there must be TGs that accept them (by Kleene's theorem).
- Let $TG_1$ accepts $L_1$ and $TG_2$ accepts $L_2$.

- Assume that $TG_1$ and $TG_2$ each have a **unique start state** and a **unique separate final state**. If this is not the case originally, then we can modify the TGs so that this becomes true as in Kleene's theorem,Part 2 of the proof (page 93).

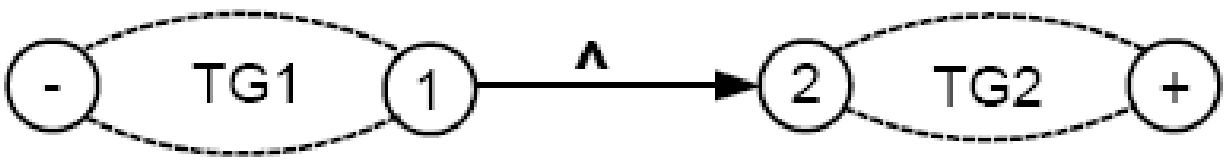
# Proof contd.

- Then the TG described below accepts the language $L_1 + L_2$.



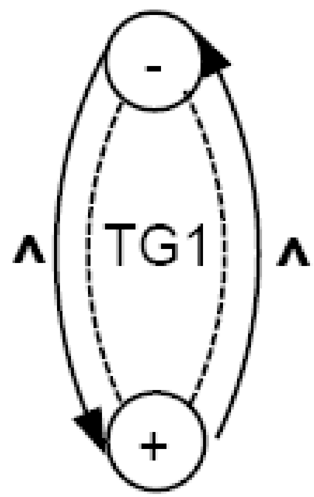

- By Kleene's theorem, since $L_1 + L_2$ is defined by this TG, it is also defined by a regular expression and hence is a regular language.

- The TG described below accepts the language $L_1L_2$ where state 1 is the former + of $TG_1$ and state 2 is the former - of $TG_2$.




- Since $L_1L_2$ is defined by this TG, it is also defined by a regular expression by Kleene's theorem, and therefore it is a regular language.

- The TG described below accepts the language $L_1$*.




- We begin at the - state and trace a path to the + state of TG1. At this point, we could stop and accept the string or jump back, at no cost, to the - state and run another segment of the input string.

# Complements

**Definition:**

- If L is a language over the alphabet , we define its **complement** L` to be the language of all strings of letters from that are not words in L.

Example:

– Let L be the language over the alphabet ∑ = {a; b} of all words that have a **double** a in them.

– Then, L` is the language of all words that do **not** have a double a in them.

- Note that the complement of L` is L. That is

(L`)` = L

# Theorem 11

**If** L **is a regular language, then** L\` **is also a regular language. In other words, the set of regular languages is closed under complementation.**

# Proof of theorem 11

- If L is a regular language, then by Kleene's theorem, there is some FA that accepts the language L.
- Some states of this FA are final states and some are not. Let us reverse the status of each state: If it was a final state, make it a non-final state. If it was a non-final state, make it a final state. The start state gets reversed as follows: - ↔ ±
- If an input string formerly ended in a non-final state, it now ends in a final state, and vice versa.
- The new machine we have just built accepts all input strings that were not accepted by the original FA, and it rejects all the input strings that used to be accepted by FA.
- Therefore, this machine accepts exactly the language L\`. So, by Kleene's theorem, L\` is regular.

# Intersection: Theorem 12

**If** $L_1$ **and** $L_2$ **are regular languages, then** $L_1 \cap L_2$ **is also a regular language. In other words, the set of regular languages is closed under intersection.**

# Proof of Theorem 12

- By DeMorgan's law (for sets of any kind):

  $$L_1 \cap L_2 = (L`_1 + L`_2)`$$

- This means that the language $L_1 \cap L_2$ consists of all words that are **not** in either $L`_1$ or $L`_2$.
- Because $L_1$ and $L_2$ are regular, then so are $L`_1$ and $L`_2$ by Theorem 11.
- Since $L`_1$ and $L`_2$ are regular, so is $L`_1 + L`_2$ by Theorem 10.
- Now, since $L`_1 + L`_2$ is regular, so is $(L`_1 + L`_2)`$ by Theorem 11.
- This means L1 ∩ L2 is regular, because

  $L1 \cap L2 = (L`_1 + L`_2)`$ by DeMorgan's law.