Theory of Automata
Transition Graphs

Hafiz Tayyeb Javed

Week 5-Lecture-01

Contents

- Relaxing the Restriction on Inputs
- Looking at TGs
- Generalized Transition Graphs
- Non-determinism

Relaxing the Restriction on Inputs

Definition of A Transition Graph

- A **transition graph**, abbreviated **TG**, is a collection of three things:

1. A finite set of states, **at least** one of which is designated as the start state (-), and some (maybe none) of which are designated as final states (+).

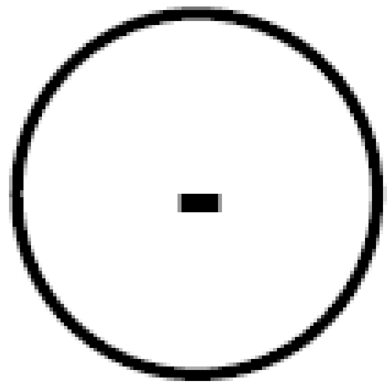
2. An alphabet **∑** of possible input letters from which input strings are formed.

3. *A finite set of transitions (edge labels) that show how to go from some states to some others, based on reading specified substrings of input letters (possibly even the null string* ***Λ****).*

Looking at TGs

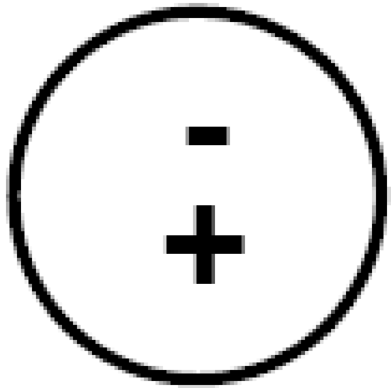
In this section, we will consider some more examples of TGs.

Example



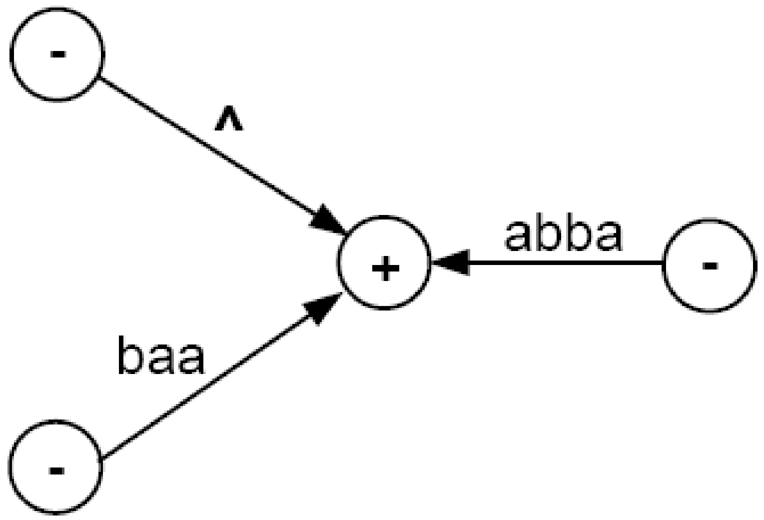
- This TG accepts nothing, not even the null string.
- To be able to accept anything, it must have a final state.

Example



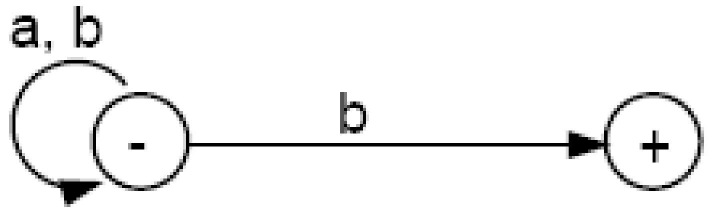
- This TG accepts only the null string **Λ**.
- Any other string cannot have a successful path to the final state through labels of edges because there are no edges (and hence no labels).
- Any TG in which some start state is also a final state will always accept the null string **Λ**. This is also true of FAs.

Example



- This machine accepts only the words Λ, baa, and abba.
- Anything read while in the final state will cause a crash, because the final state has no outgoing edges.

Example

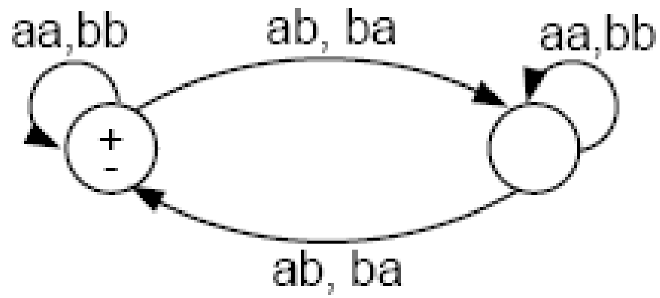


- We can read all the input letters, one at a time, and stay in the left-side state. When we read a b, if it is the very last letter of the input string, we can use it to go to the final state. Note that this b must be the very last letter, because once we are in the right-side state, if we try to read another letter, we will crash.

- It is possible for an input string ending with a b to follow an unsuccessful path that does not lead to acceptance (e.g., following the b-edge too soon and crash, or looping back to the **-** state when reading the last b).

- However, all words ending with a b can be accepted by some path. Hence, the language accepted by this TG is (a + b)*b.

Example



- In this TG, every edge is labeled with a pair of letters. Thus, for a string to be accepted, it must have an even number of letters.
- Let's call the left state the **balanced state**, and the right state the **unbalanced state**.
- If the first pair of letters that we read is a double (aa or bb), then we stay in the balanced state. While in the balanced state, we have read an even number of a's and an even number of b's.

EVEN-EVEN

- When a pair of unmatched letters is read (ab or ba), the machine flips over to the unbalanced state, meaning that it has read an odd number of a’s and an odd number of b’s.

- We do not return to the balanced state until another unmatched pair is read. The discovery of two unmatched pairs makes the total number of a’s and the total number of b’s read from the input string even again.

- This TG accepts exactly the language EVEN - EVEN.

- Recall that EVEN - EVEN is the language of all words containing an even number of a’s and an even number of b’s, including the null string **Λ**.

Generalized Transition Graphs (GTG)

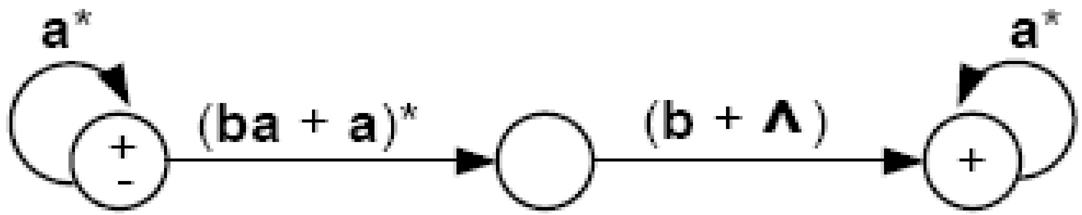
Definition

A **generalized transition graph (GTG)** is a collection of three things:

1. A finite set of states, of which at least one is a start state and some (maybe none) are final states.
2. An alphabet ∑ of input letters.
3. Directed edges connecting some pairs of states, each labeled with a regular expression.

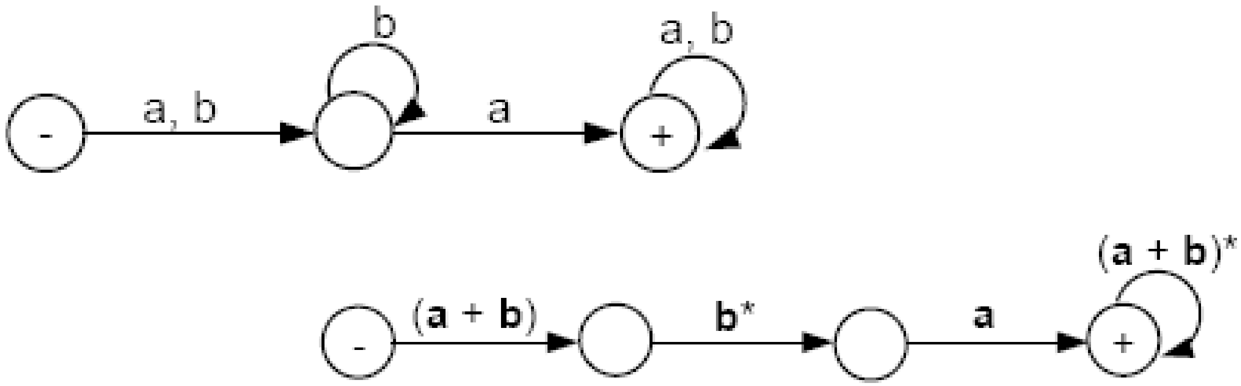
Example

- Consider this GTG:



- This GTG accepts all strings without a double b.
- Note that the word containing the single letter b can take the free ride along the Λ-edge from start to middle, and then have letter b read to go to the final state.
- Typo in textbook: The first edge should be labeled (ba + a)* as in the figure above, not (ab + a)*.

- Note that there is no difference between the Kleene star closure for regular expressions and a loop in transition graphs, as illustrated in the following figure:

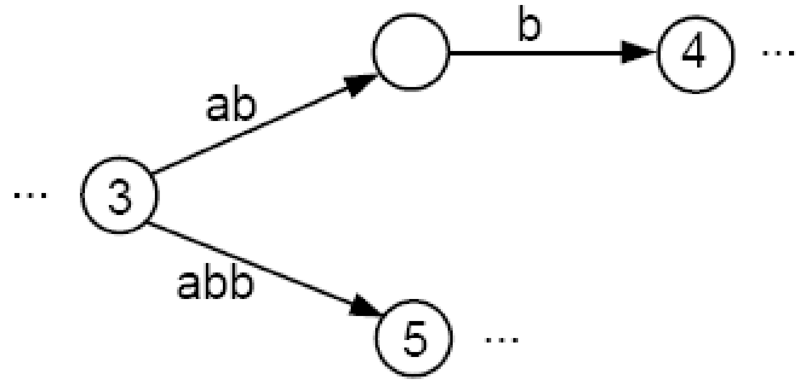


- In the first picture we may loop in the middle state or go to the third state. For not to loop, corresponds to taking the Λ choice from the b*-edge in the second picture.

NonDeterminism

NonDeterminism

- We have already seen that in a TG, a particular string of input letters may trace through the machine on different paths, depending on our choice of grouping.



- This figure shows part of some TG.

- The input string *abb* can go from state 3 to state 4, or to state 5, depending on whether we read the letters two and one, or all three at once.

NonDeterminism

- The ultimate path through the machine is NOT determined by the input alone. Human choice becomes a factor in selecting the path. The machine does not make all its own determination.

- Therefore, we say that TGs are **nondeterministic**.