Theory of Computation

النظرية الاحتسابية

المحاضرة الثانية عشر

كلية التربية للعلوم الصرفة / جامعة ديالى

اعداد

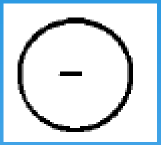
م.د. محمد سامي محمد

قسم علوم الحاسوب

المرحلة الثانية

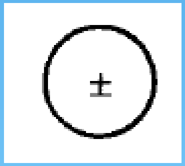
Some notes

Let us consider some more examples of TG's.



The picture above represents a TG that accepts nothing, not even the null string Λ. To be able to accept anything, it must have a final state.

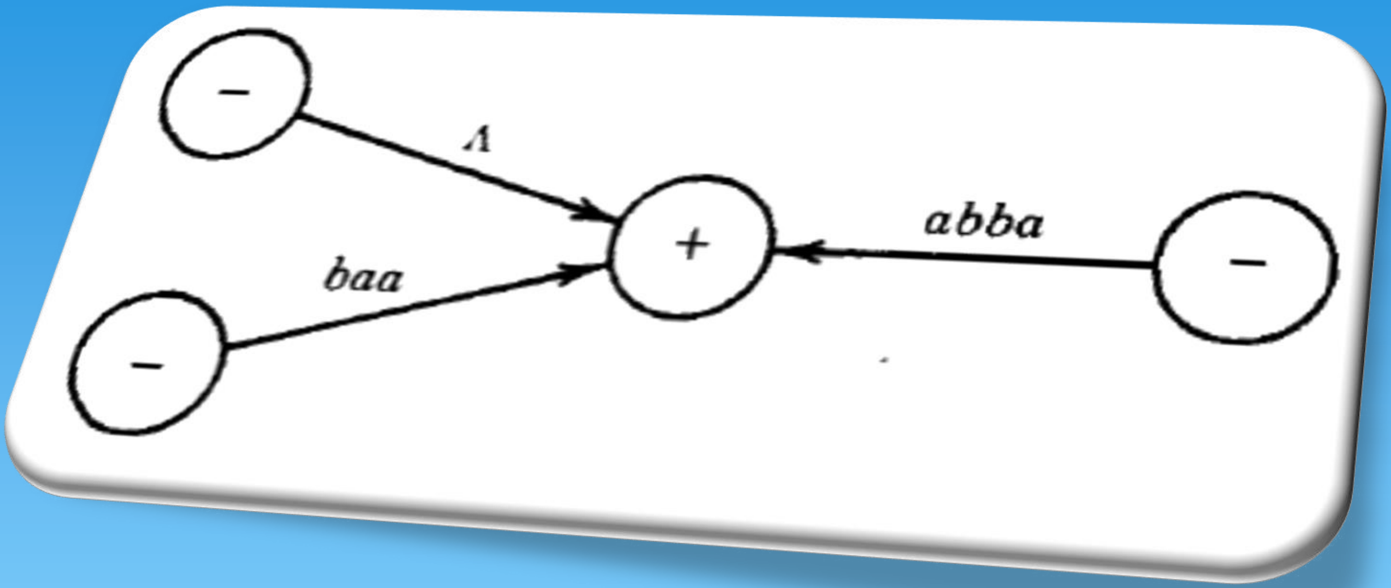
The machine



accepts only the string Λ. Any other string cannot have a successful path to the final state through labels of edges since there are no edges (and hence no labels).

Any TG in which some start state is also a final state will always accept the string Λ; this is also true of FA's. There are some other TG's that accept the word Λ, for example:

Another Example



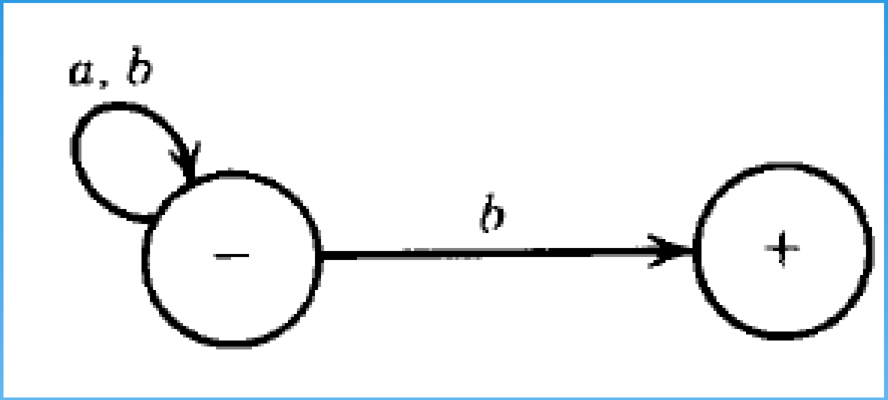
This machine accepts only the words Λ, *baa,* and *abba.* Anything read while in the + state will cause a crash, since the + state has no outgoing edges.

Explain

Design the previous example with more than one state

Another Example

Consider the following TG:



We have two conditions for b so ???????

It must be the very last letter, since once in the right-side state if we try to read another letter we crash.

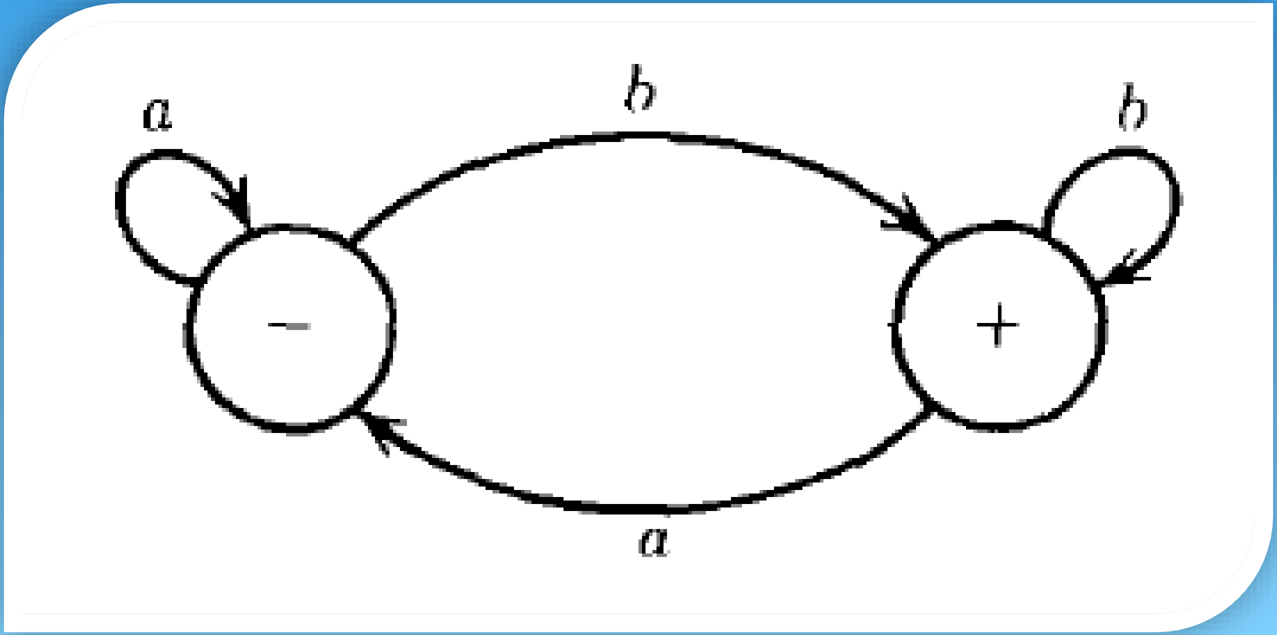
Previous Example Notes

Notice that it is also possible to start with a word that does end with a b but to follow an unsuccessful path that does not lead to acceptance.

We could either make the mistake of following the non-loop b-edge too soon (on a non-final b) in which case we crash on the next letter; or else we might make the mistake of looping back to - when we read the last b, in which case we reject without crashing. But still, all words that end in *b can* be accepted by some path, and that is all that is required.

Another Example

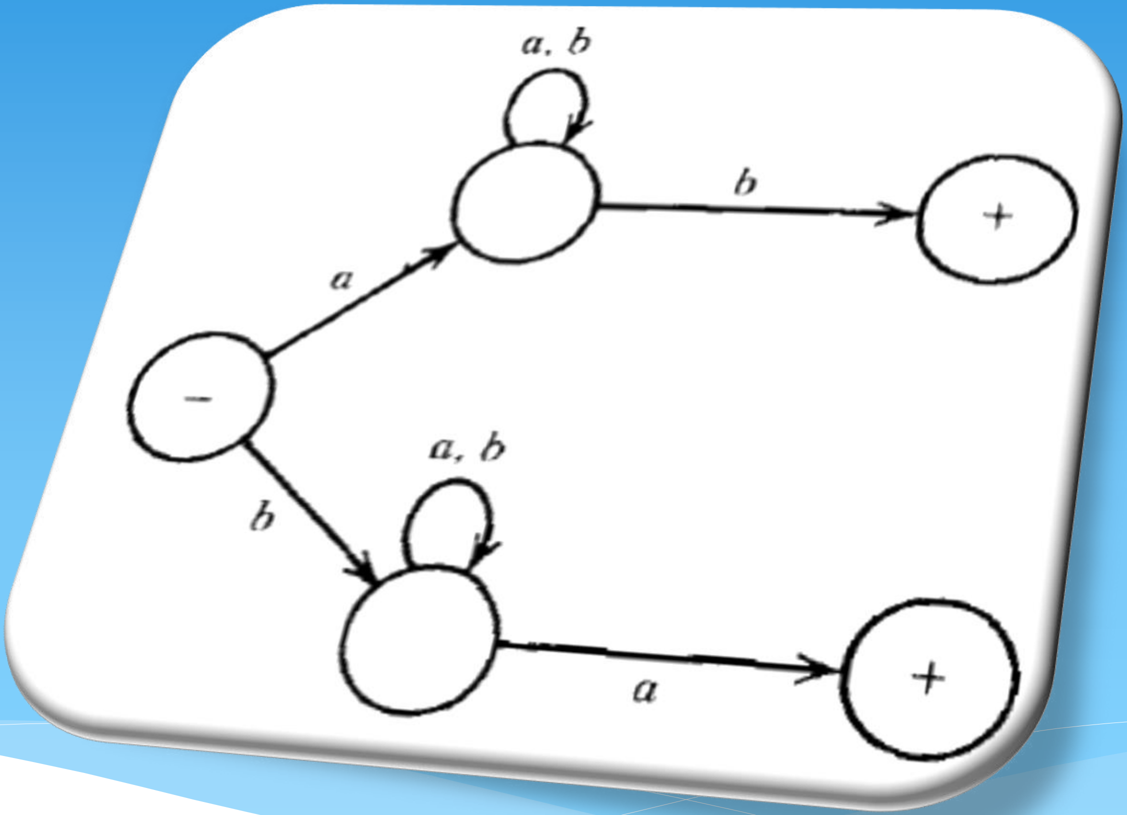
The language accepted by this TG is all words ending in b. One regular expression for this language is (a + b)*b and an FA that accepts the same language is:



Another Example

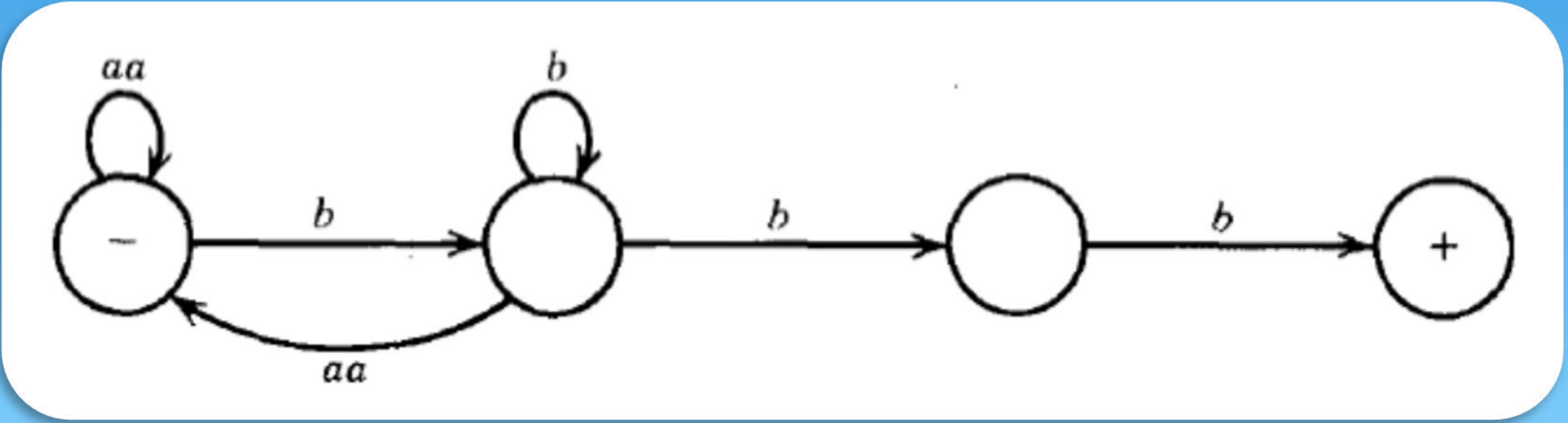
The following TG:

accepts the language of all words that begin and end with different letters.



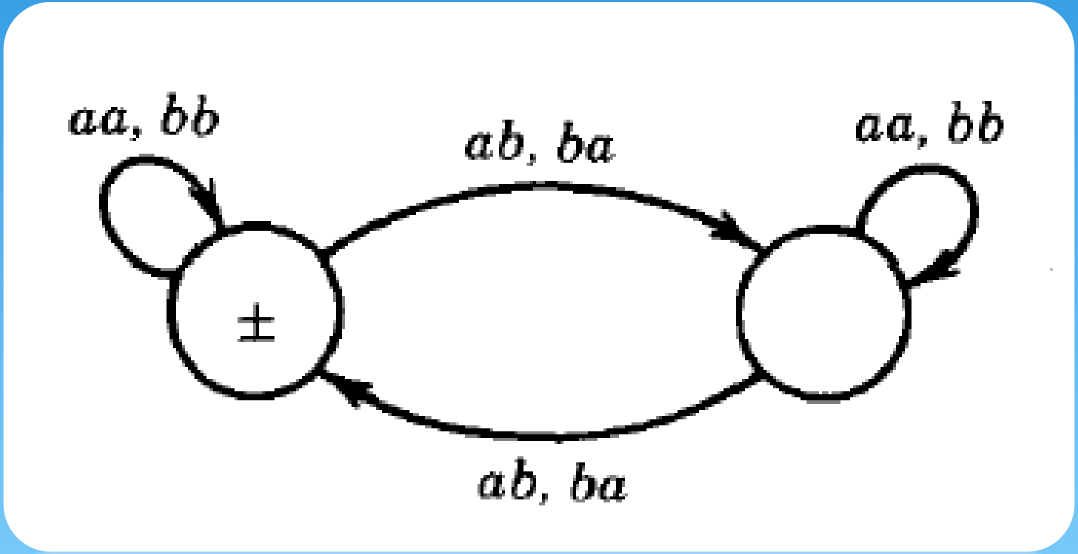
Another Example

accepts the language of all words in which the a's occur only in even clumps and that end in three or more *b's.*



Another Example

Consider the following **TG**:

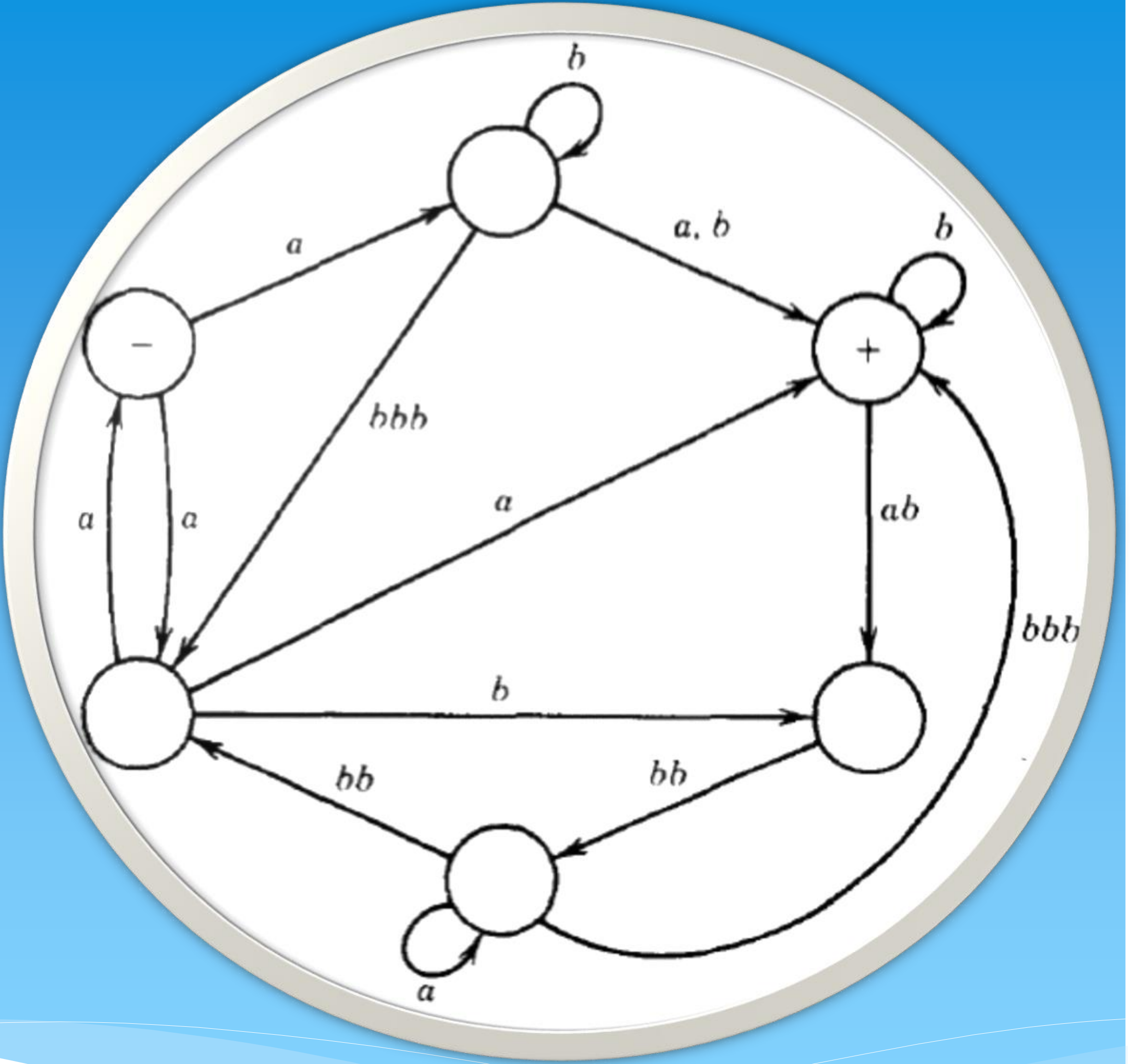


There is a practical problem with TG's. There are occasionally so many possible ways of grouping the letters of the input string that we must examine many possibilities before we know whether a given string is accepted or rejected.

EXAMPLE

Consider this TG:

abbbabbbabba



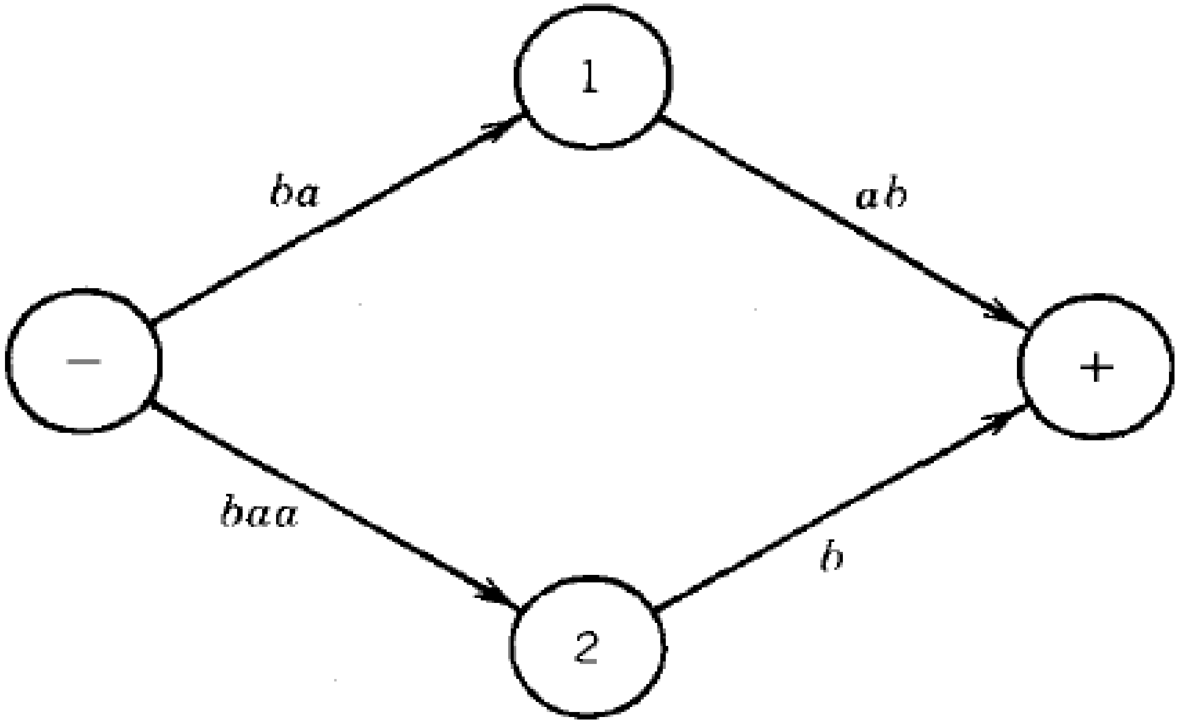
Is the word *abbbabbbabba* accepted by this machine?

Is the word *abbbabbbabba* accepted by this machine? (Yes, in two ways.)

When we allow Λ-edges we may have an infinite number of ways of grouping the letters of an input string. For example, the input string *ab* may be factored as:

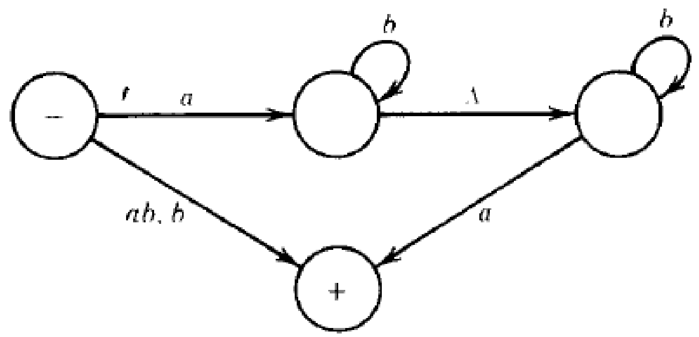
(a) (b)
(a) (Λ) *(b)*
(a) (Λ) (Λ) *(b)*
(a) (Λ) (Λ) (Λ) *(b)*

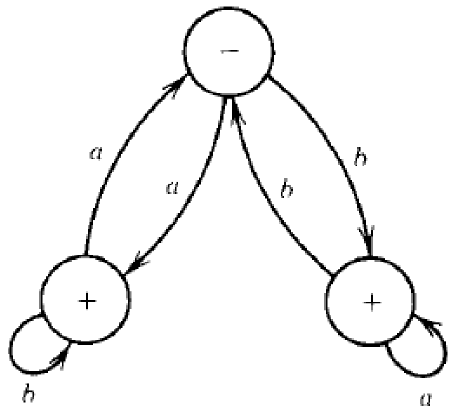
Problems



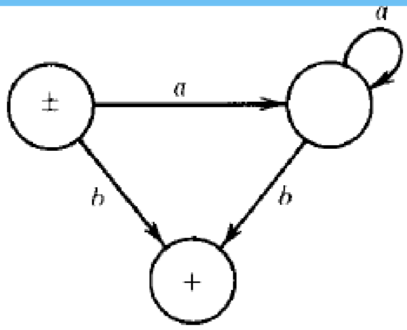
	b	ab	ba	baa
−			1	2
1		+		
2	+			
+				

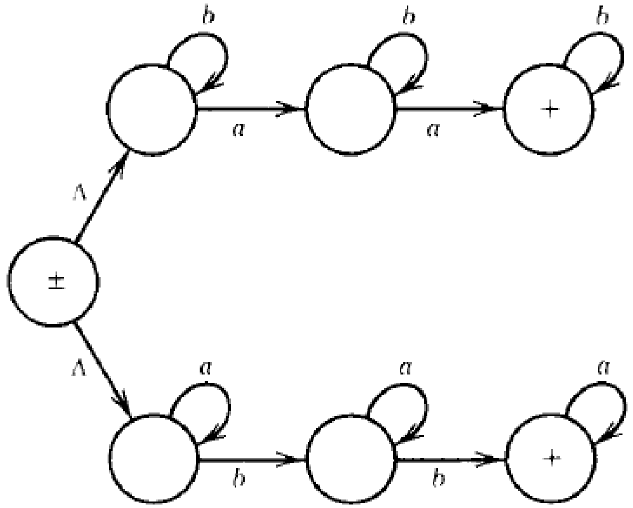
Here are six TG's. For each of the next 10 words decide which of these machines accepts the given word.

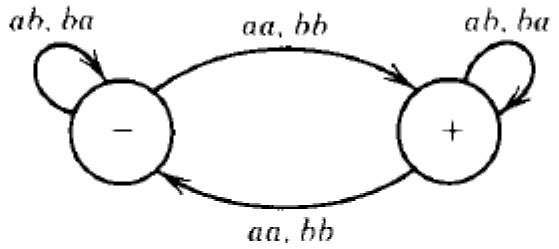


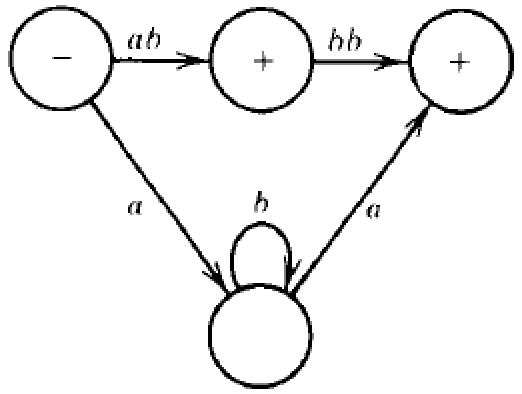


Find regular expressions defining the language accepted by each of the six TG's above.









(i) Λ
(ii) *a*
(iii) *b*
(iv) *aa*
(v) *ab*
(vi) *aba*
(vii) *abba*
(viii) *bab*
(ix) *baab*
(x) *abbb*