Model Checking
Prof. B. Srivathsan
Department of Computer Science and Engineering
Indian Institute of Technology – Madras

Lecture - 25
Bchi Automata

In this module, we will see a kind of automata which can accept languages of infinite words and these automata are called Buchi automata. Buchi is the name of the person who introduced this form of automata.

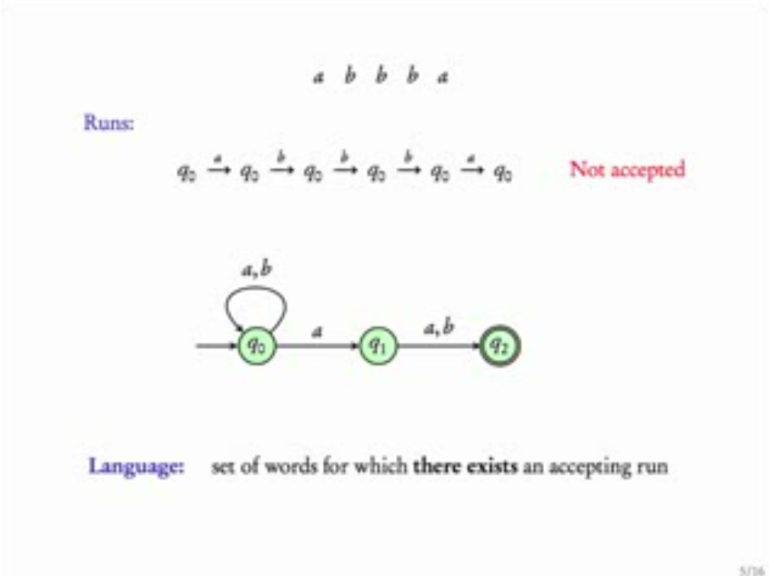
(Refer Slide Time: 00:21)

So here is the goal. We would like to give some kind of an automaton for omega regular expressions. The normal NFA, DFA work for regular expressions. Now can we come up with something like that for omega regular expressions. Once we have that, we want to take a synchronous product with the transition system of the model. Remember when we were checking regular safety properties.

We had a property in the form of an automaton and then we took a synchronous product with the automaton representing the transition system. We moved all the labels to the edges and then made all states accepting, if you remember and then we took a synchronous product. Once we did this, we checked for the emptiness of the product. So here we have a different story.

We want an automaton model for omega regular expression, that means for languages over infinite words and then we want to take a similar product with the transition system and then we want to be able to check the emptiness of this automaton. We will do a short recap of finite automata and I will give some more definitions, after which we will extend these definitions to get our Buchi automata.

(Refer Slide Time: 02:01)



Look at this NFA. This NFA accepts all words, where the second last letter is a. Suppose I ask this question, is this word accepted by this automaton, yes, indeed. How do we check this? We will run the automaton over this word. Let me explain what I mean by a run. This is one way of reading the word. You are at q0 which is the initial state. On an a there are two choices, either you can stay here or the automaton can go here.

Let us take the choice where the automaton stays here. So q0 a q0 and again on a b which stays in q0, well for a b there is not choice, it has to stay here. You read a b in q0 and on a there are two choices, either you got to the next state or you stay in the same state. This is run where the automaton stays in the same state and so on. In fact in this entire run, the word is read while the automaton stays in q0, q0 is not an accepting state.

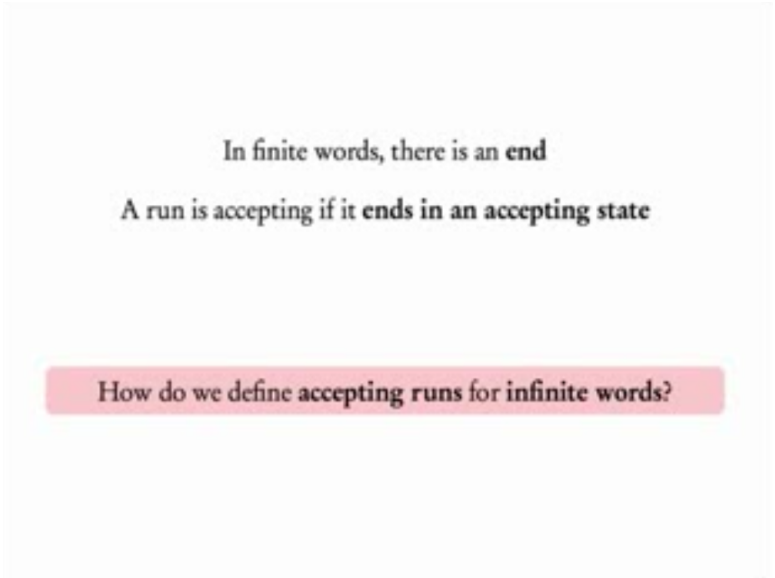
However, there is another run where the automaton stays in q0 till this point, till it reads b and on seeing this a, the automaton takes this transition goes to b1 and on b take this transition to go to q2. This run ends in an accepting state and hence this run is called an accepting run. A word is accepted if there exists an accepting run. So for this word, here is an

accepting run and the language of this automaton is the set of words for which there exists and accepting run.

Okay, the thing to note is that for this word there could be multiple runs depending on the non determination present here. This gives us a choice of taking the next transition. As long as there is one run which is accepting, the word is accepted by the NFA. Let us look at another word, a b b b a. So this word is not in the language, let us look at the runs. There is only one possibility.

The only possible run stays in q0 till the end and since the ending state is not an accepting state this run is non-accepting and the word is not accepted by this automaton. So this is giving us the criterion for accepting or not accepting a word. As long as there is one run which runs in an accepting state, the word is accepted by the automaton.

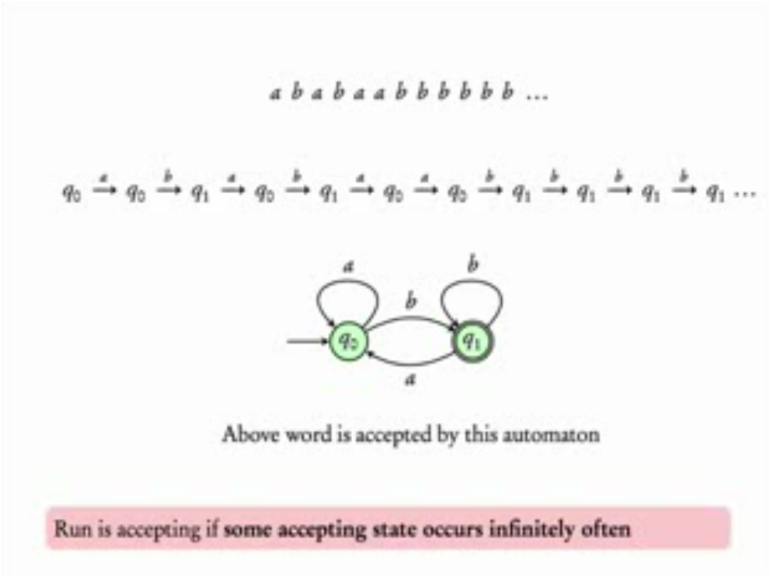
(Refer Slide Time: 05:40)



In the case of finite words, you have an end. For example, for this word the end is here at a and hence we would talk about runs which end after reading this letter. We can say that run is accepting if it ends in an accepting state. However, in the state of infinite words, there is no end. If we want to talk about automata for infinite words and we talk about runs, when would we call a run to be accepting. First of here, our runs would be infinite.

You would be running the automaton on the entire word. When do we call it an accepting run?

(Refer Slide Time: 06:36)



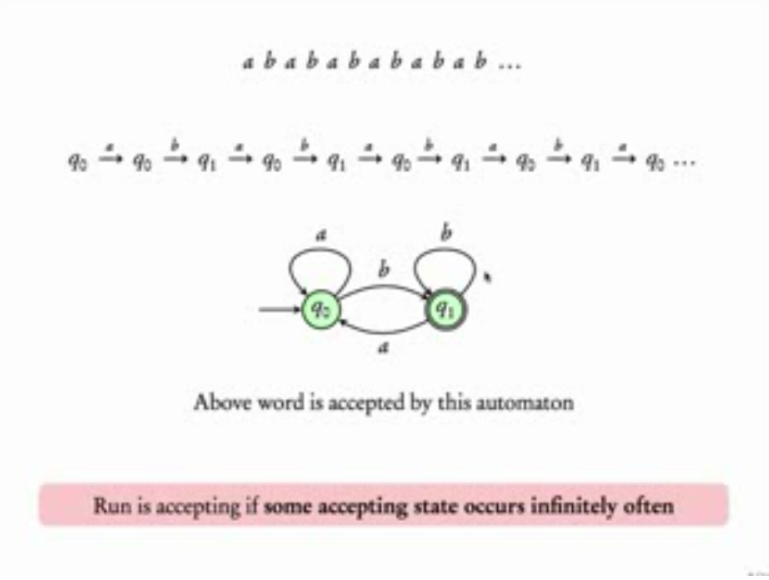
Let us see, take this automaton. And this take this word a b a b a a and after a while it sees only bs. So this is an infinite word. Let us look at run on this automaton on this word. So at q0 when the automaton reads an a it stays in q0, when it reads a b it goes to q1. On an a it comes back to q0, on a b it goes to q1. On an a it comes back to q0, stays in q0 when it sees an a and then on a b it goes to q1 and then it keeps staying in q1 forever.

So this is a word and this is an automaton and this is a run of the automaton of this word. When do we say that this run is accepting? Note that this is marked as an accepting state. When do we call this run to be accepting? We say that the run is accepting if some accepting state occurs infinitely often. Here there is only one accepting state which is q1, look at this run, q1 comes here, q1 comes here forever.

So q1 comes infinitely often in this run. So this run will be called accepting and since this word has an accepting run, this word would be accepted by the automaton. Note that we are now interested in finding automata for languages of infinite word. The first step would be define the notion of acceptance. This is an example illustrating that we took a word which is infinite and we considered a run of the automaton on it.

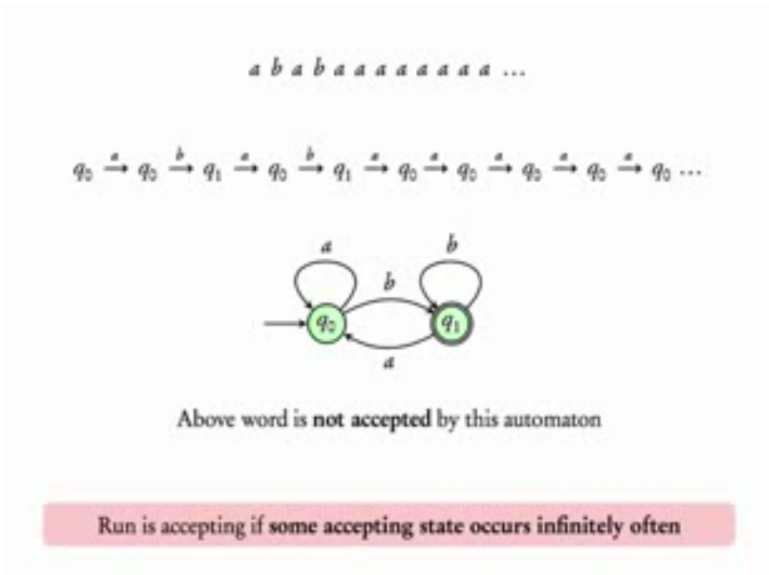
Here is a run and this run contained the accepting state infinitely of it and so we called this run as an accepting run.

(Refer Slide Time: 09:13)



This is the same automaton. Look at this word a b a b a b a b and so on. Let us now run the automaton on this word. You start at q0 on an a. The automaton stays at q0, on a b it goes to q1, on an a comes back to q0, goes to q1, q0 q1, q0 q1 and so on. Recall that run is accepting if some accepting state occurs infinitely often and this run Q1 occurs infinitely often. So this run is accepting and hence the above word is accepted by this automaton. We are slowly trying to understand the language of this automaton.

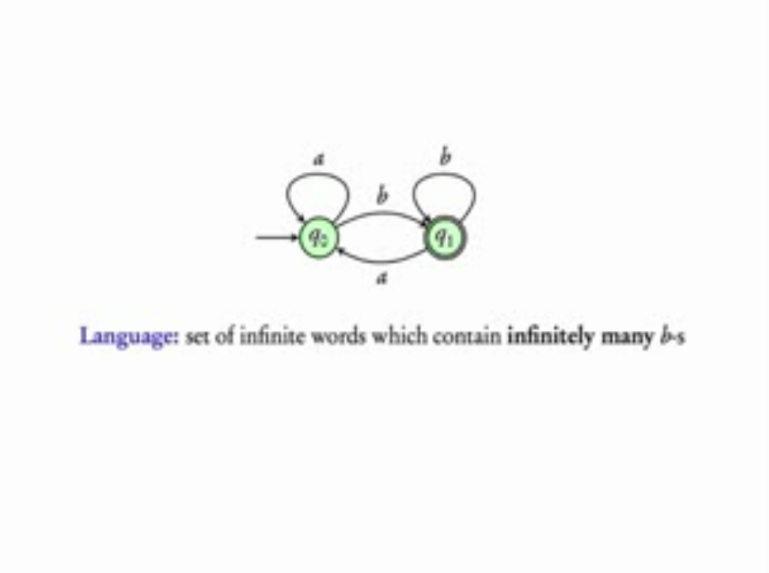
(Refer Slide Time: 10:12)



Let us see another example. Suppose we have the word, a b a b followed by only a. At q0 on an a you stay at q0, on a b go to q1, again when a come to q0 b go to q and then you keep seeing a. So for the first a you come to q0 and then you will stay at q0 for ever. Recall that the run is accepting if some accepting state occurs infinitely often. This is the only run for this word and this run q1 occurs up to this point only.

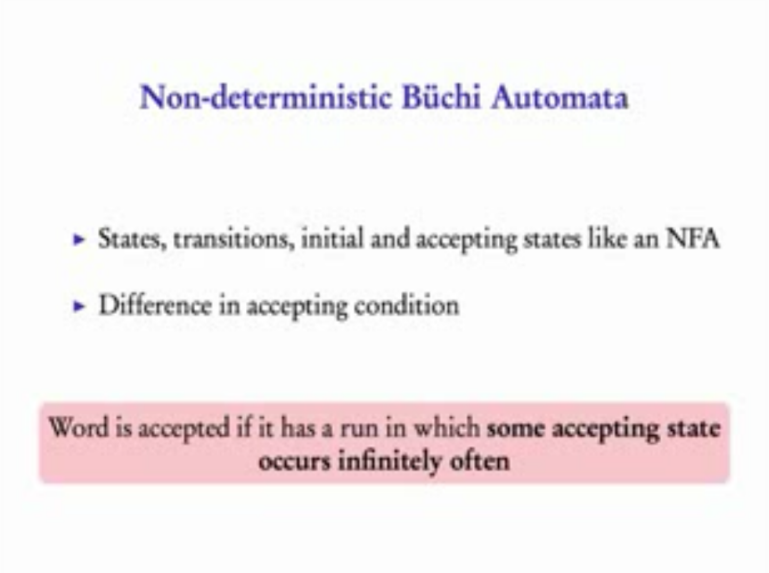
After this only q0 occurs. So q1 does not occur infinitely often. Hence this word is not accepted by this automata according to this condition. If you see q1 will occur infinitely often only when b occurs infinitely often. So this automaton considered as an automaton over infinite words would give you the language consisting of words where b occurs infinitely often.

(Refer Slide Time: 11:35)



This is what is said here. So this is an automaton over infinite words and it is called a Buchi automaton.

(Refer Slide Time: 11:46)

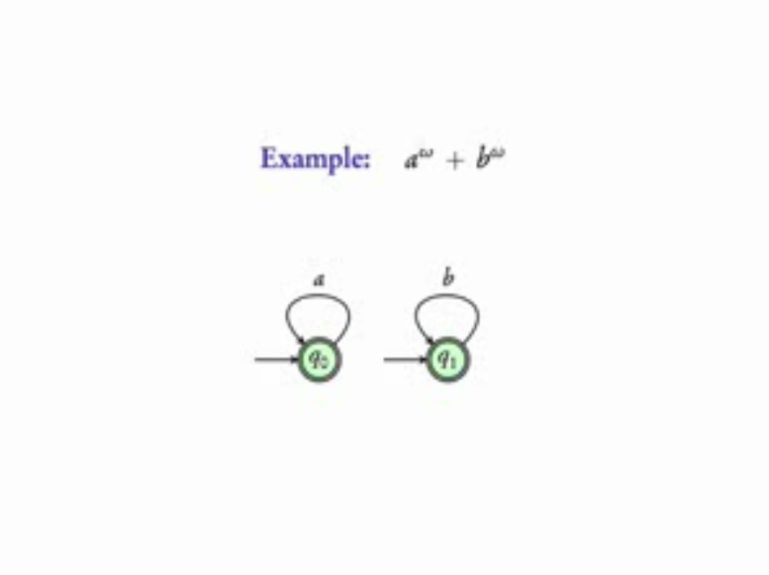


So like non deterministic finite automata we have non deterministic Buchi automata, you have state transitions, initial and accepting states like an NFA. So here in fact this is

deterministic even, but in general you have states transitions, initial and accepting states. The only difference is that, Buchi automata are for infinite words. So there is a difference in the accepting condition.

A word is said to be accepted if it has a run in which some accepting state occurs infinitely often. In the case of finite words, we said that if there is a run which ends in an accepting state here we say that here should be a run where some accepting state occurs infinitely often.

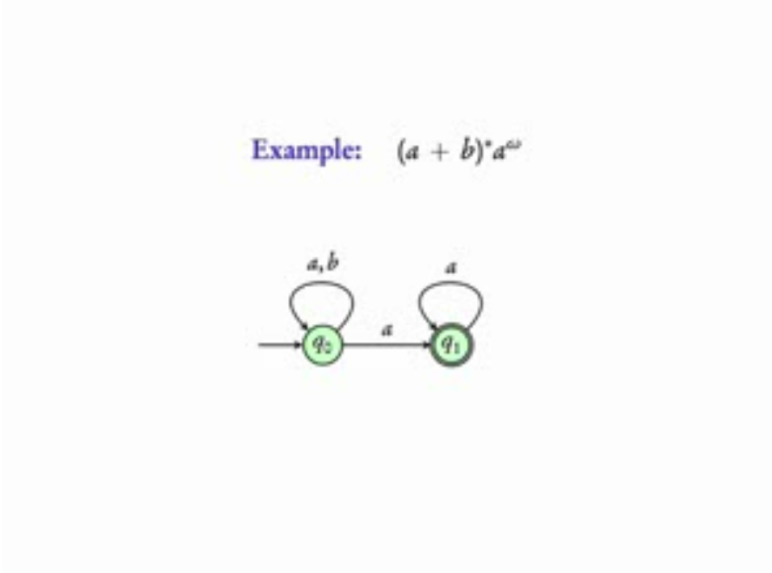
(Refer Slide Time: 12:41)



Let us see some examples. Suppose this is the language defined by the omega regular expression, a power omega plus b power omega. So there are only two words, the infinite word consisting only of a and the infinite word consisting only of b. This is the NDA, non deterministic Buchi automaton, accepting this language. If you look at a a a a a, it will start at q0 and keep seeing q0 again and again.

Since q0 is accepting, a power omega will be accepted. Once you are in q0 you cannot read a b. So no other word will be accepted apart from a power omega from q0, if the automaton chooses to start from q1, it can read only b and since q1 is an accepting state and it occurs infinitely often, b power omega will be accepted.

(Refer Slide Time: 13:41)



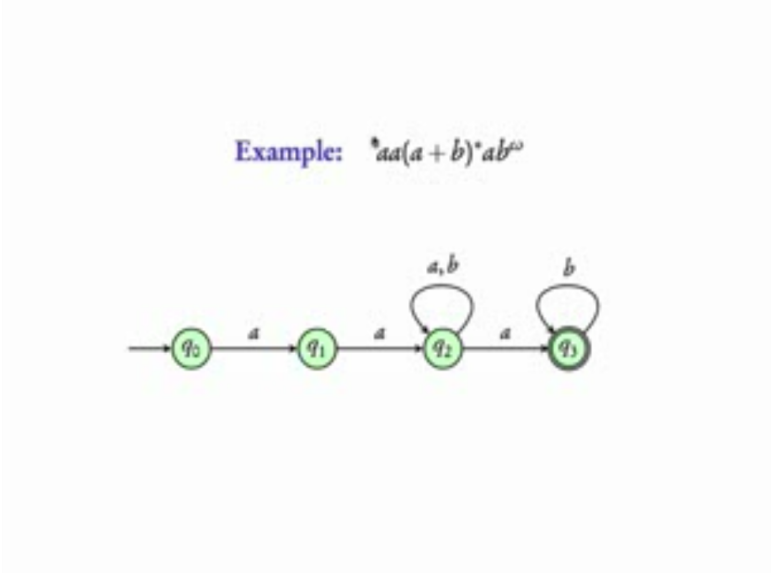
Let us look at this language, a plus b star a power omega. It says that you can read a b some finitely many times, after which you should switch to a state where you read only a. So this is what is happening here. You read a b, as long as you stay here, rather if you stay here forever, q1 does not occur at all. So if you keep staying here forever, that would be a non accepting run.

When can you go to q1, you can choose an a and go to q1 and once you do that you can stay only in a. So every word in this language the moment you shift to a power omega, you have to shift to this state and keep reading only a. And since q1 occurs infinitely often, the word will be accepted. For example, if I look at the word b power omega, it is not in this language. So it should not be accepted by this automaton, let us see.

What would a run on b power omega b, there is only one run, we will start from q0, since there is only this transition on a b, you have to keep seeing q0 again and again and again and since q0 is a non accepting state, b power omega would not be accepted. Similarly, if b occurs infinitely often, you have to keep staying in q0 because once you read an a, you cannot read a b.

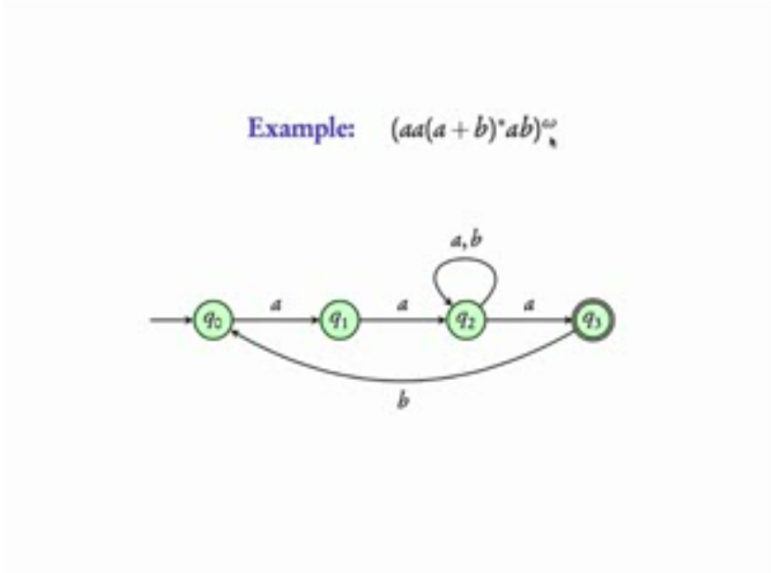
So this expression is giving us the language where b occurs only finitely often.

(Refer Slide Time: 15:30)



Let us look at another example, a a a plus b star a and then you jump to b power omega. So this is what is happening, give the automaton for a a a plus b square a and then you move to a state which we will be accepting and here you read only bs. So a word in this language would have an accepting run. This non deterministic Buchi automaton corresponds to this omega regular expression.

(Refer Slide Time: 16:06)



One more example, here this entire thing has to repeat infinitely often. You should see a word in a a a plus b star a b and again you should see a word in a a a plus b star a b and again you should see a word in a a a plus b star a b raise to omega, that is the meaning of this omega. So you draw the automaton, a a a plus b star a b, here I have made q3 to be the acceptance state. So the automaton has to keep looping around this if this wants to accept a word and if it loops around this, then it will be a a a plus b star a b.

(Refer Slide Time: 16:56)

We have seen this new concept called non-deterministic Buchi Automaton. This is a kind of automaton which is used for describing languages over infinite words. In particular, this automaton would be useful to talk about properties described by omega regular expressions. Now what is the difference from finite automata, here of course you are talking about infinite words.

And the word is accepted if there is a run which sees an accepting state infinitely often. Try to write your own regular expressions, omega regular expressions and try to form non deterministic Buchi Automaton for these expressions.