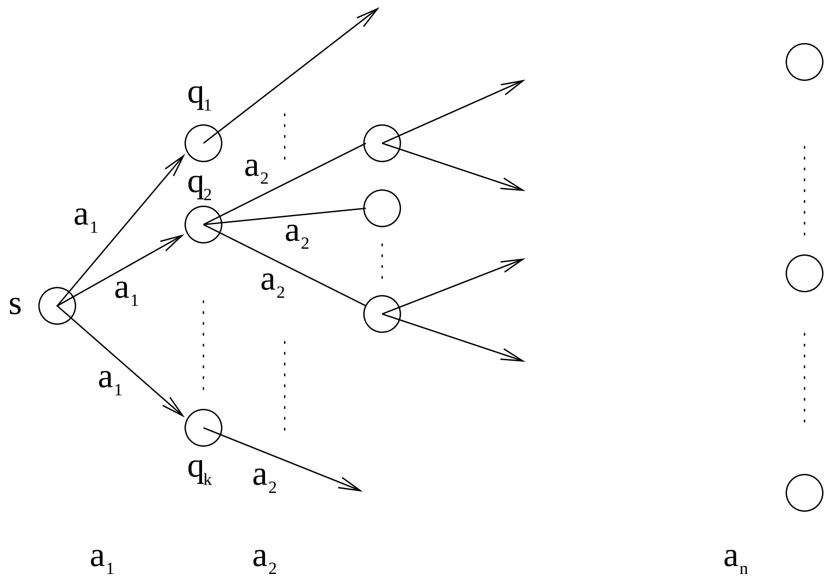
Lecture 6: Alternating Automata

In this lecture we shall examine a new technique to prove that nondeterministic finite automata can be complemented. We have already seen two techniques, one via Myhill-Nerode classes and another via the powerset construction. In the powerset construction we show that a subclass of NFAs, namely deterministic automata, are closed under complmentation and equivalent in expressive power to NFAs. In this lecture we shall go the other way. We shall generalize NFAs to Alternating Finite Automata and show that this larger class, for which closure under complementation is easy to establish, is equal in expressive power to NFAs.

The motivation comes from the following idea. In order to check that a given NFA does not accept a word w, it suffices to do the following: Suppose $w = a_1 w'$ and $\delta(s, a_1) = \{q_1, q_2, \ldots, q_k\}$. Then it suffices to start k copies of the automaton, one from q_1, one from q_2 and so on and verify that none of these has an accepting run on w'. Of course, to verify that q_i has no accepting run on $w' = a_2 w''$ we shall do the same thing, i.e., start one copy for each state in $\delta(q_i, a_2)$ and verify that all these copies do not accept w'' and so on. In this process, we get a tree of depth $|w|$ where the edges at level i are all labelled by a_i, and w is not accepted iff every leaf node in this tree is labelled by a state from $Q \setminus F$.

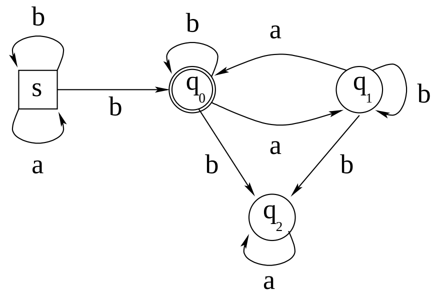


Here is a different way of looking at this idea: Usually, to check that the automaton accepts a word aw starting at a state q, we inductively proceed by evaluating $\mathsf{Acc}(q, w)$ where $\mathsf{Acc}(p, \epsilon)$ is true whenever $p \in F$ and $\mathsf{Acc}(q, aw) = \bigvee_{p \in \delta(q,a)} \mathsf{Acc}(p, w)$. This corresponds to an interpretation of the transition function δ as a logical or. On the other hand, the above construction interprets $\mathsf{Acc}(q, aw)$ as $\bigwedge_{p \in \delta(q,a)} \mathsf{Acc}(p, w)$ with $\mathsf{Acc}(p, \epsilon)$ being true whenever if $p \in Q \setminus F$. It interprets the transition function as a logical and.

Once we permit the interpretation of the transition function as a or in one automaton and as a and in another, we can go one step further and permit both of these choices coexist in the same automaton. This leads us to alternating automata.

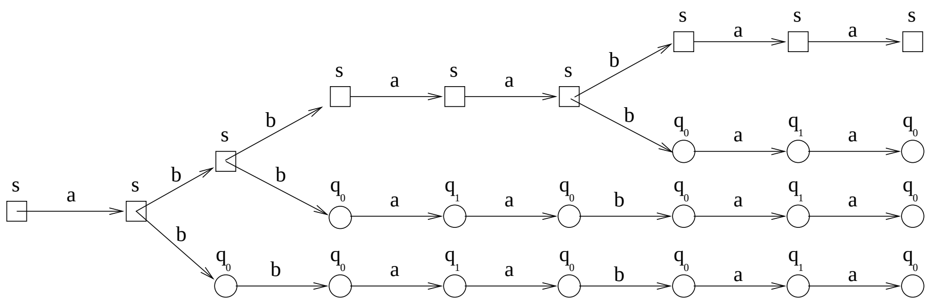
The set of states of an alternating automaton is divided into two sets $Q_\exists$ and $Q_\forall$. (The idea being that the transitions out of states in $Q_\exists$ are to be interpreted as or transitions and the transitions out of states in $Q_\forall$ are to be interpreted as and transitions.) Syntactically, this is the only difference between an NFA and an AFA. So, an AFA is of the form $(Q_\exists, Q_\forall, \Sigma, \delta, s, F)$ and we shall write Q for $Q_\exists \cup Q_\forall$.

Here is a an alternating automaton $(\{q_0, q_1, q_2\}, \{s\}, \{a, b\}, s, \delta, \{s, q_0\})$ where we have drawn the $\forall$ states as squares and the $\exists$ states as circles.

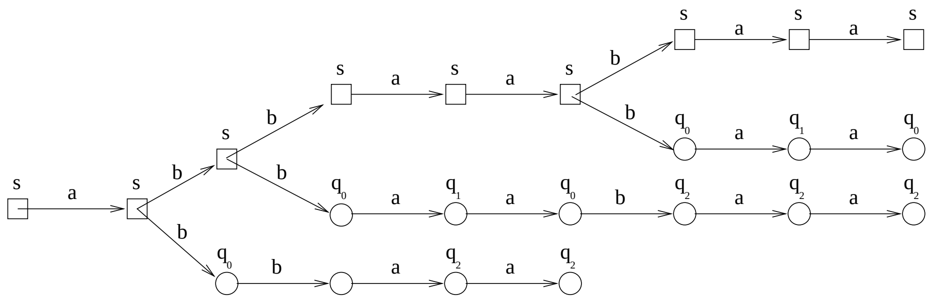


How does this automaton work? From the state s the automaton reads the entire input (since on both a as well as b there is a self-loop back to s), but since it is a $\forall$ state, on reading each b, it starts of an additional copy at state q_0 to read the rest of the input. It is quite easy to check that from state q_0 a word is accepted precisely when it has even number of a's. Thus, the language accepted by this automaton consists of all those words in which the number as to the right of each b is even.

Here is an accepting run of this automaton on the word *abbaabaa*. Notice, how the run branches in the state s on input b.



An alternating automaton can have multiple runs on a given input, because of the presence of nondeterminism. Here is another (partial) run of the above automaton on *abbaabaa*



The run is partial as along one of the paths the entire input is not read (since there is not transition on b at the state q_2.) Thus, this is not an accepting run. (Moreover, there is another path along which the entire input is read but the resulting state is not accepting.)

Formally a run of an alternating automaton $(Q_\exists, Q_\forall, \Sigma, \delta, s, F)$ on an input $a_1a_2\ldots a_n$ is a Q labelled rooted tree where

1. The root is labelled by s.
2. If an interior node, at some level i, is labelled by some $q \in Q_\exists$ then it has single child which is labelled by a p where $p \in \delta(q, a_i)$.
3. If an interior node, at some level i, is labelled by some $q \in Q_\forall$ and $\delta(q, a) = \{q_1, \ldots q_k\}$ then it has k children, labelled q_1, q_2 $\ldots$ q_k.

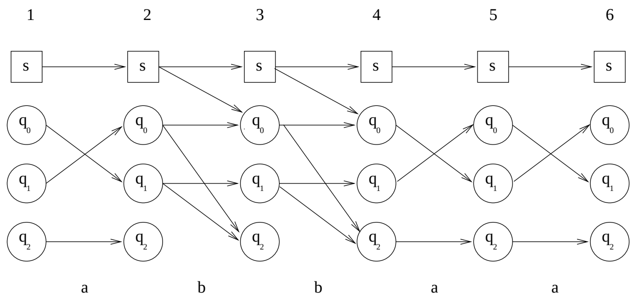
A run is accepting if all the leaf nodes are labelled by states in F.

In the next section we outline a notion of a game played on finite graphs (as a matter of fact finite DAGs suffices for this lecture) and relate it to alternating automata.

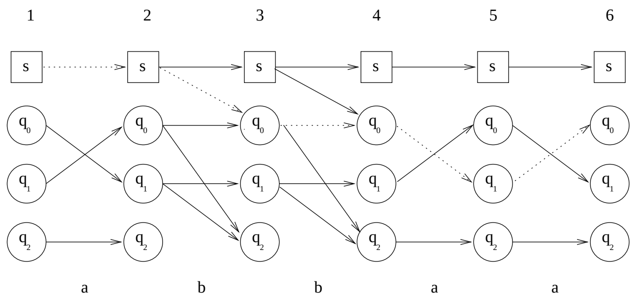
1 Games on Finite DAGs

With every alternating automaton $A = (Q_\exists, Q_\forall, \Sigma, \delta, s, F)$ and word $w = a_1a_2\ldots a_n$, we can associate a directed (acyclic) graph with vertex set $Q \times \{1, 2, \ldots, n+1\}$. Edges go from level i to level $i+1$. If $\delta(q, a_i) = \{q_1, \ldots, q_k\}$ then there are edges from (q, i) to $(q_j, i+1)$ for $1 \leq j \leq k$. We shall refer to this game as $\mathcal{G}(A, w)$.

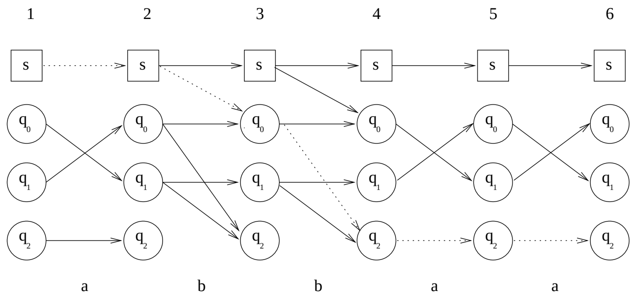
Here is the graph corresponding to the automaton described in the previous section and the word *abbaa*.



Given such a graph we define a game as follows: There are two players, *automaton* and *pathfinder*. The set of nodes is divided among the two players. The automaton owns all the nodes labelled with states from $Q_\exists$ and the pathfinder owns the nodes labelled by states from $Q_\forall$. A token is placed on the node $(s, 1)$ (i.e. the copy of the start state in the first level). In each round the player who owns the node with the token makes the move. He chooses one of the neighbours (via outgoing edges) and moves the token there. The play ends when the token reaches a vertex with no outgoing edges. The aim of the automaton is to ensure that the token reaches a node labelled by an accepting state in level $n+1$. We say that the automaton wins the play if this happens (i.e. the token lies on a level $n+1$ node labelled with a final state.) Otherwise, the pathfinder wins the play. For example, the following play (marked by dotted edges in the figure below) is winning for the automaton:



On the other hand, the following play is winning for the pathfinder.



Observe that we do NOT require that the players alternate in making moves. So much so that, one player may NOT make any moves at all during a play. For instance, if the pathfinder were to always move from s to s in the above game the automaton would never get an opportunity to make a move. However this would be very clever on part of the pathfinder since he would lose such a play!

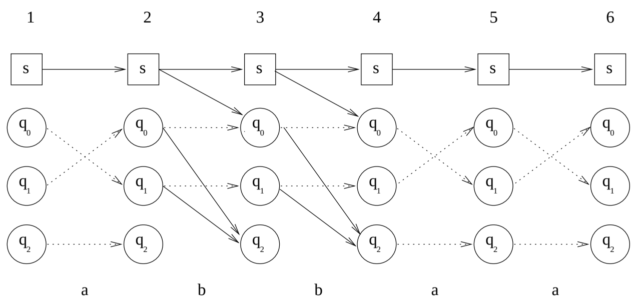
A *strategy* f for a player (automaton/pathfinder) is a function that "examines" the entire play so far and decides what move to make. (Formally a strategy for the automaton is a function that whose domain is the set of sequences of the form $q_1 \longrightarrow q_2 \longrightarrow q_3 \ldots q_i$ with $q_i \in Q_\exists$ and for such an input the function returns a p where $p \in \delta(q_i, a_i)$. Similarly, a strategy for the pathfinder is a function whose domain is the set of sequences of the form $q_1 \longrightarrow q_2 \longrightarrow \ldots q_i$ with $q_i \in Q_\forall$ and it must return a $p \in \delta(q_i, a_i)$.) We say that a play is *consistent with strategy* f for the automaton (pathfinder) if all the moves made by the automaton (pathfinder) along this play are as suggested by the strategy f. We say that f is a winning strategy for the automaton (pathfinder) if every play consistent with f is winning for the automaton (pathfinder).

Claim 1: If the automaton has an accepting run on the given input word, then he has a winning strategy in the game associated with that word.

Proof: The idea is to do the following: As the play proceeds we trace a corresponding path through the accepting run (which is a tree) and use that in defining the next move. The path starts at the root (which is labelled by the start state) and the starting configuration in the game consists of the token the token being placed at the start state in level 1.

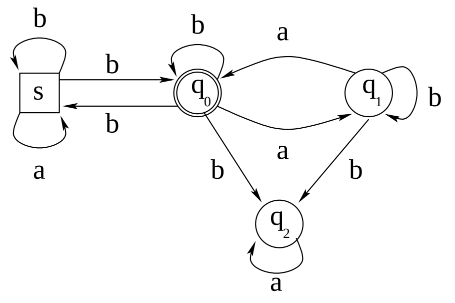
Inductively, assume that after i moves in the game the token is at a vertex labelled q in level $i+1$ and that corresponding path that we have traced in the accepting run ends a node at level $i+1$ labelled by the same state q. If q is a $Q_\exists$ state then there is unique child in the rune and suppose this is labelled p then our strategy recommends move $(q, i+1) \longrightarrow (p, i+2)$. If $q \in Q_\forall$, then it is the turn of the pathfinder to make a move. Suppose he moves the token to $(p, i+2)$ then we trace the edge from q at level $i+1$ to p at level $i+2$ in the accepting run (which must exist, since every state in $\delta(q, a_{i+1})$ appears as the label of a child of $(q, i+1)$.) Thus, following this strategy, after n rounds of moves the token will a node at level $n+1$ in the graph whose label is identical to that of the state (at level $n+1$) reached by the path traced in the accepting run. But every leaf node in the accepting run is labelled by a a state from F. Thus, this play ends in a node labelled by a state in F and is winning for the automaton. Since this argument works for all plays it follows that the strategy defined above is a winning strategy for the automaton. ■

A strategy f is said to be *positional* or *memoryless* if its value on $q_1 \longrightarrow q_2 \longrightarrow \ldots q_i$ depends only only q_i. That is, $f(q_1 \longrightarrow q_2 \longrightarrow \ldots q_{i-1} \longrightarrow q_i) = f(q'_1 \longrightarrow q'_2 \longrightarrow \ldots q'_{i-1} \longrightarrow q_i$ for any two runs ending at q_i. Thus, we may think of a positional winning strategy for the automaton (pathfinder) to be a function that sends each node that belongs to the automaton (pathfinder) to one of its neighbours. The following figure indicates a positional winning strategy (where the edges chosen by the strategy are marked by dots) for the automaton:

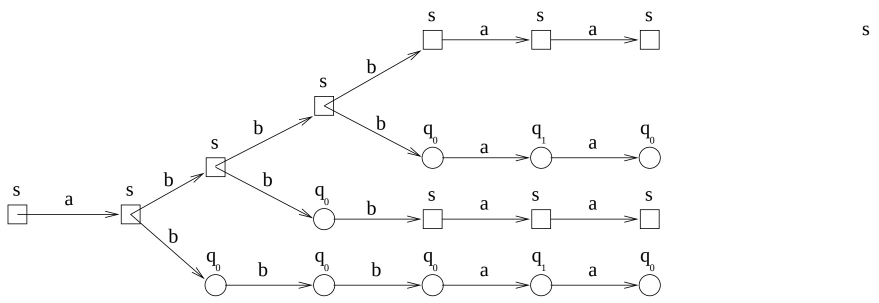


In the above game, an example of a non-positional winning strategy for the automaton is one that moves to q_0 on $s \longrightarrow s \longrightarrow q_0 \longrightarrow q_0$ and to q_2 on $s \longrightarrow s \longrightarrow s \longrightarrow q_0$. However such a strategy is not winning because, the pathfinder would play $s \longrightarrow s$ in the first round and $s \longrightarrow s$ in the second and $s \longrightarrow q_0$ in the third and now the automaton would use f to move to q_2 from where it cannot win the game.

Exercise: Consider the following automaton. Describe a non-positional and a positional winning strategy for the automaton in the game associated with the word *abbbaa*. Are there any non-positional strategies for the pathfinder in this game?



Will the strategy constructed in the proof of Claim 1 a positional strategy? Not always. It is possible that the accepting run may have the same state q appearing twice at the same level i and the state labelling the children of these two occurances may not be the same. If the paths from the root to these two occurances are $q_0 \longrightarrow q_1 \longrightarrow \ldots q_i$ and $q_0 \longrightarrow q_1' \overset{q\quad \prime}{\longrightarrow}_2 \longrightarrow \ldots \longrightarrow = q_i$ respectively, then the strategy induced by this run would play different moves on $q_0 \longrightarrow q_1 \longrightarrow q_2 \ldots q_i$ and $q_0 \longrightarrow q_1' \longrightarrow q_2' \ldots \longrightarrow q_i$ and thus it is not positional at q_i. As an illustration, consider the following accepting run of the automaton A_1 defined in the above exercise (on input *abbbaa*).



The strategy defined using this run yields q_0 on $s \longrightarrow s \longrightarrow s \longrightarrow q_0$ while it yields s on $s \longrightarrow s \longrightarrow q_0 \longrightarrow q_0$. Let us call an accepting run to be positional for if all occurances of a state $q \in Q_\exists$ in any fixed level i, the label of the unique child of each these occurances is the same.

It is quite easy to check that if we start with positional accepting run then, the strategy constructed in the proof of Claim 1 is a positional strategy for the automaton. Claim 1 together with this observation leads to the following lemma.

Lemma 1 *Let A be an alternating automaton and let w be a word. In the game associated with A and w, the automaton has a winning strategy whenver A accepts w. Moreover, if A has positional accepting run on w then the automaton has a positional winning strategy.*

What about the converse? Does the existence of a winning strategy for the automaton in the game $\mathcal{G}(A, w)$ guarantee that A accepts w? The answer turns to be yes with a rather simple proof.

Lemma 2 *Let A be an alternating automaton and let w be a word. A accepts w whenever the automaton has a winning strategy in the game $\mathcal{G}(A, w)$. Moreover, if the automaton has a positional winning strategy then there is a positional accepting run on w.*

Proof: Let f be a winning strategy for the automaton. We define the run level by level. At level 1, there is a single node, the root and it is labelled by s. Suppose we have inductively constructed the run upto level i. Pick any node at level i and suppose it is labelled by the state q. Consider the path $s \longrightarrow q_1 \longrightarrow q_2 \ldots q$ from the root to this node. if q is in $Q_\forall$, and $\delta(q, a_i) = \{p_1, p_2, \ldots p_k\}$, then add k children to this node and label them as p_1, p_2, ..., p_k. Otherwise, create a single child with label $f(s \longrightarrow q_1 \xrightarrow{q}_2 \ldots \longrightarrow q)$.

Notice that every path in this tree corresponds to a play of the game that is consistent with the strategy f for the automaton. Therefore, the state labelling the last node must be in F. Thus every leaf in this tree is labelled by a state in F and hence it is an accepting run. It is trivial to check that if f is a positional winning strategy then the constructed run is positional. ■

Thus, we have established a strong relationship between A accepting the word w and the existence of a winning strategy for the automaton in the game $\mathcal{G}(A, w)$.

Theorem 3 *The automaton wins the game $\mathcal{G}(A, w)$ if and only if A accepts w.*

2 Determinacy for the games $\mathcal{G}(A, w)$

In this section we shall prove that for any automaton A and word w, either the automaton or the pathfinder has winning strategy in the game $\mathcal{G}(A, w)$. As a matter of fact we prove:

Theorem 4 *Let A be an alternating automaton and w be a word. Either the automaton or the pathfinder has a positional winning strategy in the game $\mathcal{G}(A, w)$.*

Proof: We will consider every node in $\mathcal{G}(A, w)$ as a possible starting state for the game (not just the vertex with label s in the first level) and show that in each case either the automaton or the pathfinder has a positional winning strategy. We shall prove this starting at the nodes at level $n + 1$ and work our way back.

If the game begins at a node at level $n + 1$ then there are no moves to be made and if the label is in F then the automaton alwys wins the game and otherwise the pathfinder wins. The strategy is trivially positional as there are no choices available.

Suppose, we have determined for every vertex in levels $> j$ which player wins the game starting at the vertex and a positional winning strategy for that player. Consider a vertex v at level j. There are two cases to consider.

Case 1: Suppose v is labelled by a state $q \in Q_\exists$. If there is even one neighbour w of v in level j from where the automaton has a (positional) winning strategy to win the game then the automaton can also win the game beginning at v. He simply moves to w from v and then follows the (positional) strategy from w. Otherwise, the pathfinder wins the games beginning at each neighbour of v in level j. Suppose the neighbours are $v_1, v_2, \ldots v_k$. Then, the automaton has to make the first move and it will move it to some v_l. The pathfinder simply plays the (positional) strategy that is winning for him from v_l. Thus, either the automaton or the pathfinder has a (positional) winning strategy at v.

Case 2: Suppose v is labelled by a state $q \in Q_\forall$. Once again it is easy to see that if at least one neighbour of v in level j is winning for the pathfinder then the pathfinder has a (positional) winning strategy from v and otherwise the automaton has a (positional) winning strategy from v.

Thus, by induction, for each vertex v in $\mathcal{G}(A, w)$ either the automaton or the pathfinder has a positional winning strategy to win the game beginning at v. In particular, if the game begins at s either the automaton or the pathfinder has a positional winning strategy. ■

Notice that this result also implies that whenever the automaton (or the pathfinder) has a winning strategy it also has a positional winning strategy.

3 Back to Alternating Automata

Now that we have the positional determinacy for the games $\mathcal{G}(A, w)$ we can complement alternating automata and prove that their expressive power is the same as that of nondeterministic automata.

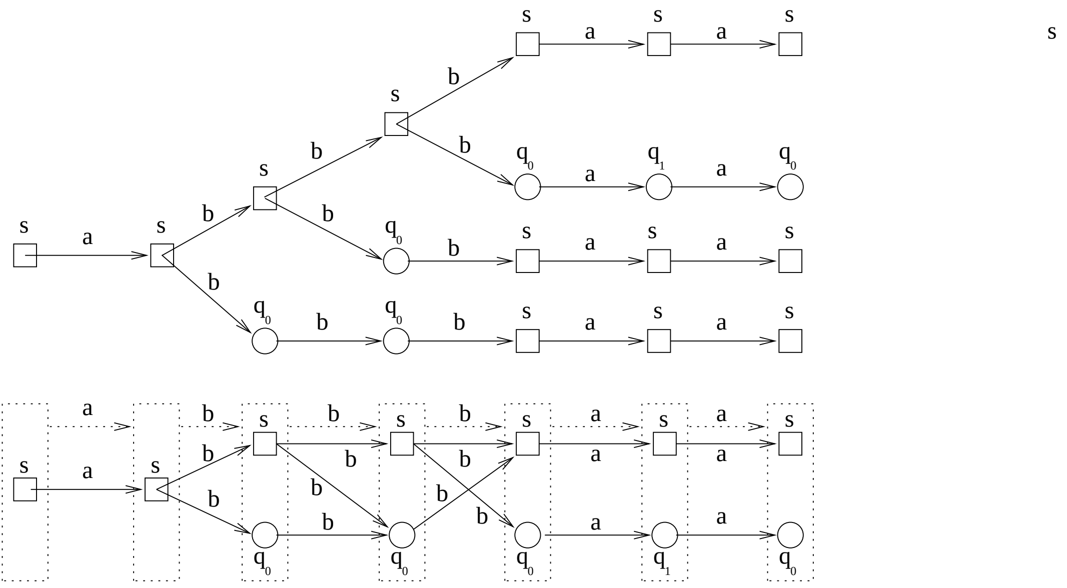
Right at the beginning of this lecture, we described a technique for complementing nondeterministic automata. In the language of alternating automata, it can be restated as follows: Given an automaton $A = (Q_\exists = Q, \emptyset, \delta, s, F)$, the automaton $(\emptyset, Q_\forall = Q, \delta, s, Q - F)$ accepts the complement of the language accepted by A. This is just a special case of the following theorem, which says that to complement an alternating automaton it suffices to exchange the $\forall$ and $\exists$ states and complement the accepting set.

Theorem 5 *Let $A = (Q_\exists, Q_\forall, \delta, s, F)$ be an alternating automata. Then $A^d = (Q_\forall, Q_\exists, \delta, s, Q - F)$ accepts the complement of $L(A)$.*

Proof: By Theorem 3 $w \notin L(A)$ if and only if the automaton does NOT have a winning strategy in the game $\mathcal{G}(A, w)$. By Theorem 4, the automaton does not have a winning strategy in the game $\mathcal{G}(A, w)$ if and only if the pathfinder has a winning strategy in the game $\mathcal{G}(A, w)$. Finally, since the game $\mathcal{G}(A^d, w)$ is nothing but the game $\mathcal{G}(A, w)$ where the roles of the automaton and the pathfinder have been interchanged and the winning condition has been complemented, it is trivial to see that the pathfinder has a winning strategy in $\mathcal{G}(A, w)$ if and only if the automaton has a winning strategy in $\mathcal{G}(A^d, w)$. And finally, by Theorem 3, this happens if and only if A^d accepts w. ■

And finally we turn out attention to showing that every alternating automaton accepts a regular language. The naive idea that suggests itself, is to somehow generate runs of the alternating automaton using a nondeterministic automaton. A run of an alternating automaton is a tree which expands one level at a time. So the natural idea would be take the set of all possible levels as the set of states of the NFA. However, there are infinitely many possible levels, since the tree may be of unbounded width. The alternating automaton described at the beginning of the section has $k+1$ leaves on reading the word b^k. However, note that k of these leaves are labelled by s.

In general, for any alternating automaton with m states, there are at the most m distinct states at any level. So can we keep the set of states appearing at a level rather than the level itself? The answer is yes, and here is where the existence of positional accepting runs comes to our rescue. A positional run can be represented by collapsing together duplicate vertices at each level so that we are left with a DAG with at the most m nodes at each level. For example, the here is a run of the automaton A_1 on *abbbaa* and its corresponding DAG representation (ignore the dotted boxes and arrows for the moment):



The NFA we construct, attempts to generate such a DAG and accepts a word only if it can generate a DAG corresponding to a positional accepting run. The dotted boxes in the above figure represent the states of the NFA in the run on *abbbaa* and the dotted arrows are the transitions in the run.

Theorem 6 *Let $A = (Q_\exists, Q_\forall, \Sigma, \delta, s, F)$ be an alternating automaton. Let A' be the nondeterministic finite automaton $(2^Q, \Sigma, \delta', \{s\}, 2^F \setminus \{\emptyset\})$ where*

$$\delta'(X,a) = \{X' \subseteq \delta(X,a) \mid \forall q \in X \cap Q_\forall.\ \delta(q,a) \subseteq X' \ \wedge\ \forall q \in X \cap Q_\exists.\ X' \cap \delta(q,a) \neq \emptyset\}.$$

Then A' accepts $L(A)$.

Proof: In one direction, it is easy to observe that if X_i is the list of states that appear at level i in a positional accepting run on $a_1 a_2 \ldots a_n$ then $X_1 \xrightarrow{a_1} X_2 \xrightarrow{a_2} X_3 \ldots \xrightarrow{a_n} X_n$ is an accepting run of A'.

For the other direction, note that if $X \xrightarrow{a} Y$ is a transition in A' then any positional accepting run on any w with X as the set of labels of the leaves can be extended to a positional run on wa with Y as the set of labels of the leaves. This completes the proof.

■

In particular the NFA corresponding to A^d for some A, essentially guesses a positional run for A^d on w or equivalently a positional winning strategy for the automaton in the game $\mathcal{G}(A^d, w)$ which in turn corresponds to a positional winning strategy for the pathfinder in the game $\mathcal{G}(A, w)$.

Thus we have generalized nondeterministic automata to alternating automata. We associate a game $\mathcal{G}(A, w)$ with every word w and accepting runs of A on w are in correspondence with winning strategies for the automaton in the game $\mathcal{G}(A, w)$. The connection between the game $\mathcal{G}(A, w)$ and accepting runs extends further. The determinacy of games allows us to translate non-existence of winning strategies for the automaton to existence of winning strategies for the pathfinder. Thus, we have achieved a *quantifier switch* (i.e. we have demonstrated that we can say "exists strategy (for the pathfinder) which is winning" instead of "every strategy (for the automaton) is not winning"). This leads us naturally to the complementation of AFAs. The complement automaton is simply the dual automaton the exchanges the two players and the winning condition. The existence of positional strategies allows us to turn AFAs in NFAs.

As we shall see in the coming lectures, this trick of connecting automata to games, establishing the determinacy of the game and using that to complement the automata is a recurring theme and a very powerful technique.

Notes: Alternation as a technique was introduced by Chandra and Stockmeyer. Alternating finite state automata have been used traditionally in the setting of infinite words and infinite trees. We choose to introduce them over finite words so that technical difficulties encountered at the infinite case do not obscure the ideas that make them such a powerful technique. It is true that all our proofs about AFAs would have been simpler proofs if we used a direct method rather than use the connection to games. However, our aim is to describe the ideas behind this connection and its use at this simple setting before studying the same in more complex settings. We have also used a rather restricted definition of alternating automata, once again for the same reason.

References