Lecture 12: Optimality of Safra's Determinization and the Kupferman-Vardi Complementation

I shall begin by presenting the "original" Safra's construction (see [2]). Let $A = (Q, \Sigma, \delta, s, F)$ be a nondeterministic Büchi automaton. We shall tranform this into an equivalent deterministic Rabin automaton A_s. The states of A_s are "Safra trees". A Safra tree is a tree with nodes drawn from the set $\{n_1, n_2, \ldots, \}$ (we shall restrict this to a finite set soon), equipped with a labelling function that labels nodes with subsets of Q and a colouring function that colours the nodes with either white or green. We shall write $\lambda(n)$ to denote the set labelling node n and $c(n)$ to denote the colour of the node n. A Safra tree satisfies the following properties:

1. If $\{n_1, n_2, \ldots n_k\}$ are the children of n then $\lambda(n)$ is a strict superset of $\bigcup_i \lambda(n_i)$.

2. If n_1 and n_2 are children of some n then $\lambda(n_1) \cap \lambda(n_2) = \emptyset$.

Observation 1: The depth of a Safra tree is bounded by $|Q|$. This follows from item 1 above.

Observation 2: The number of nodes in a Safra tree is bounded by $|Q|$. This can be seen as follows: pick any n in a Safra tree T. We claim that there is a q such that q appears in the sets labelling the nodes on the path from the root to n and no where else. In proof, pick any q that appears in $\lambda(n)$ but not in the labels of any of its children. Such a q must exist by item 1. (Observation 2 will allow us to restrict the set of nodes to the finite set $\{n_1, n_2, \ldots, n_{2|Q|+1}\}$.)

The start state of A_s is the tree with the single node n_1 labelled by the set $\{s\}$. It is coloured green if $s \in F$ and white otherwise. We now describe the transition function. Given T and a letter a we shall construct a Safra tree T' as follows:

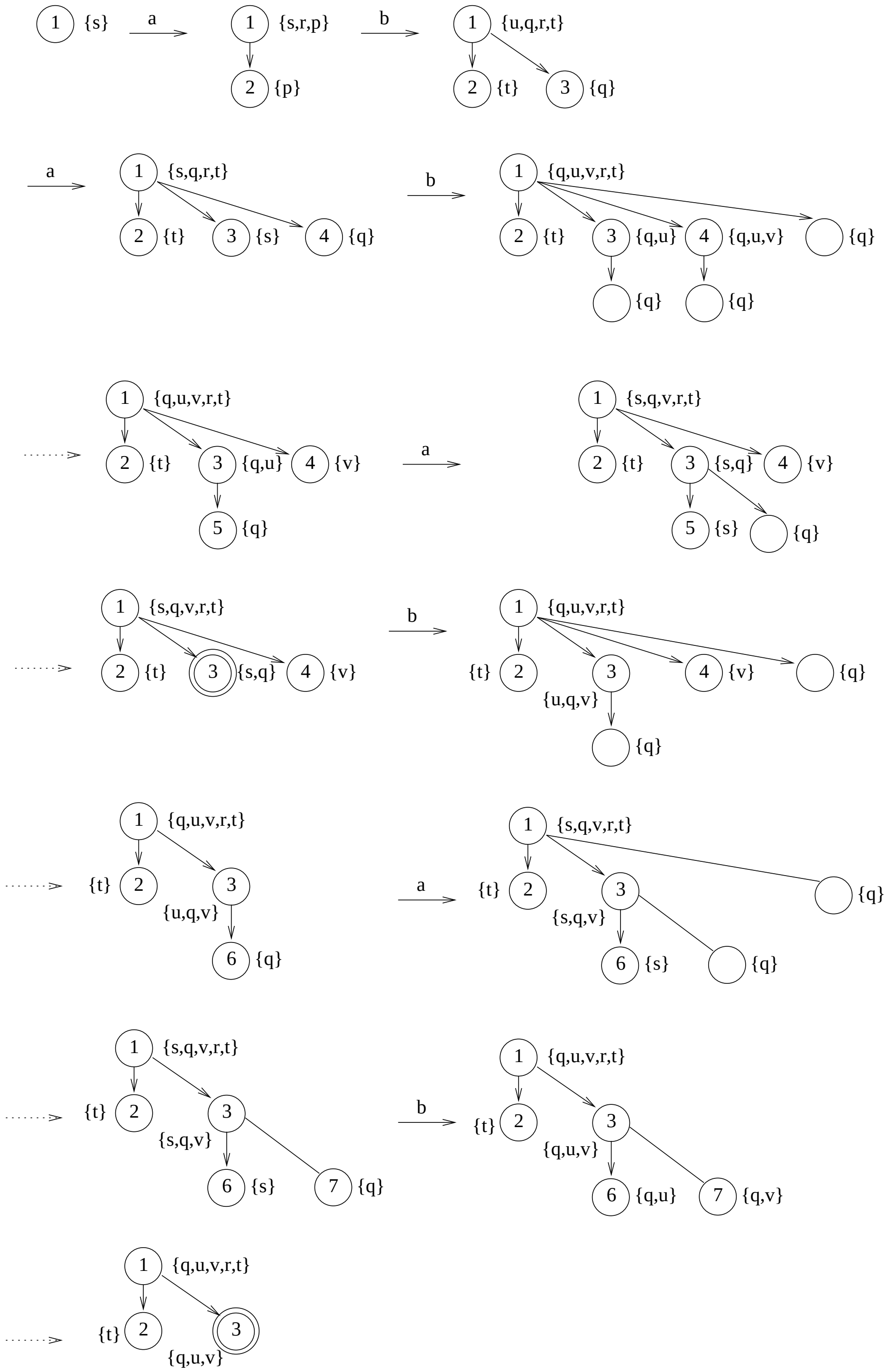
1. Every node in n in T is also a node of T'. The label of such a node is $\delta(\lambda(n), a)$ (Thus each node runs its own copy of the powerset automaton A_p instead of a copy of the marked powerset automaton A_m. As observed in the previous lecture, the second component of A_m is available in the state of the children of a node and so it suffices to simulate A_p at each node.)

2. For each $n \in T$, if $\delta(\lambda(n), a) \cap F$ is not empty then we create a new child for n in T' and label it with $\delta(\lambda(n), a) \cap F$. (Instead of forking one copy per accepting state, we simply start a single copy simultaneously from all of those states.)

3. If a state q appears along more than one path starting at the root delete it from all paths except the left most (with the understanding a child inserted earlier appears to the left of a child inserted later.) If in this process any node has the label $\emptyset$ then delete that node.

4. If n is any node with children $m_1, m_2, \ldots m_k$ with $\lambda(n) = \bigcup_i \lambda(m_k)$ then delete $m_1, \ldots m_k$ along with all their descendents and colour n with green. Label other nodes white.

The figure below describes a run of the automaton on page 6 of Lecture 11, on the input *ababab*. The effect of steps 1 and 2 are indicated via $\longrightarrow$ while the effect of steps 3 and 4 are indicated via dotted arrows.



Lemma 1 *A word $w = a_1a_2\ldots$ is accepted by A if and only if in the unique run of A_s on w, some node n is coloured green infinitely often.*

Proof: Let an accepting run of A on $a_1a_2\ldots$ be $q_1 \xrightarrow{a_1} q_2 \xrightarrow{a_2} q_3 \ldots$. Let $T_1 \xrightarrow{a_1} T_2 \xrightarrow{a_2} \ldots$ be the run of A_s.

First of all note that if n appears in T without children, $T \xrightarrow{x} T'$ such that the node n is not deleted along this run and further n is coloured green in T' then, $\lambda(n) \xrightarrow{x}_g \lambda'(n)$ (where $\lambda(n)$ is the label of n in T and $\lambda'(n)$ is the label of n in T'.) This follows from the definition of the transition relation of A_s. Also observe that, whenever a node is coloured green it has no children. From this it is quite easy to see that if some node n is coloured green infinitely often along the run of A_s on w and if the sets labelling n at these green coloured stages are $X_1, X_2, \ldots$ then $w = w_1w_2\ldots$ with $\{s\} \xrightarrow{w_1}_g X_1 \xrightarrow{w_2}_g X_2 \ldots$. Thus w is accepted by A.

For the converse, we associate a node $N_i \in T_i$ with the accepting run $q_1 \xrightarrow{a_1} q_2 \xrightarrow{a_2} \ldots$. It is the highest level node (i.e. farthest from the root) whose label carries the state q_i. We claim that $N_i = n$ for some n infinitely often. In proof note that if the associated node turns out to be the root (which appears in all the trees T_i) then there is nothing to prove. Otherwise, after some M_1, the associated node N_i is never the root for all $i \geq M_1$. Let m_1 be the node at level 2 visited after this. Beyond M_1, whenever the associated node is at level 2 it can only be m_1 or any other node at that level that appears to the left of m_1. Thus there are only finitely many choices. If it visits one such node infinitely often, we are done. Otherwise, there is is a point M_2 beyond which the associated node appears at level 2 or more. Continuing this argument and using the fact that the depth of the trees is bounded by $|Q|$ we conclude that there is a n that is hit infinitely often as the associated node.

Now we claim that this node n must be coloured green infinitely often. This follows from the fact that whenever q_i is an accepting state the associated node moves from n to a child of n. Since it returns back to n, there is a point where the children of n are deleted. At this point n must be coloured green.

Thus, the node n is coloured green infinitely often along the run of A_s on w. ■

Now, we can use the fact that the size of the tree is bounded by $|Q|$ to use a finite number of nodes (akin to the argument in Lecture 11, Lecture 8 and so on). We simply use a collection of $2|Q|+1$ nodes and ensure that whenever a node n is deleted at a step, it does not appear in the tree constructed (as all the new nodes added can be given other labels). As usual, we shall use a Rabin acceptance condition (F_i, I_i) one for each node i. F_i consists of all the trees where i does not appear and I_i consists of all the trees where i appears and is coloured green. It is easy to see that this finite state deterministic Rabin automaton accepts the same language as A_s. Henceforth we shall use A_s to refer to this automaton. What is the size of this automaton? Simply observe that there are at most n^{n-2} trees on n nodes and if we insist on ordered trees we get $n^{n-2} * n!$ and labelling involves picking functions from Q to nodes. Thus the total number of trees is of the order of $n^{n-2} * n! * n^{|Q|}$. But $n = 2 * |Q| + 1$. Thus, the number of trees is of the order of $2^{O(|Q| log(|Q|))}$.

Theorem 2 *(Safra) Any nondeterministic Büchi automaton with n states can be transformed into an equivalent deterministic Rabin automaton of size $O(2^{O(nlogn)})$ with $O(n)$ ac-*

cepting pairs. Any NBA can be complemented into a NBA with a size blowup of at most $O(2^{O(nlogn)})$.

For a proof of the second part use the following exercise.

Exercise: Show that every nondeterministic Streett automaton with n states and r accepting pairs can be tranformed into an equivalent nondeterministic Büchi automaton with $O(n * 2^r)$ states.
[**Hint:** The set of F_is hit infinitely often must be a subset of the set of E_is hit infinitely often. Keep track of the set of indices for which E_i has been hit and the set of indices for which F_i has been hit. If the latter is a subset of the former then reset both sets to the empty set and continue...]

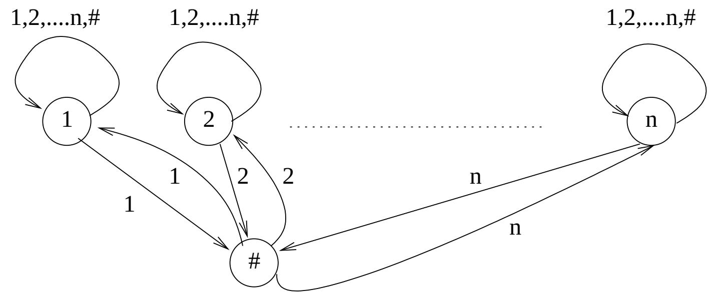
1 A lower bound

In this section we shall show that we cannot hope to improve Safra's construction. This result is due to M. Michel. Our presentation here follows that of C. Löding [1].

The idea is to construct a family of nondeterministic Büchi automata A_i of size n such that any NBA accepting $\overline{L(A_i)}$ must have at least $n!$ states. The automaton $A_n = (\{1, 2, \ldots, n, \#\}, \{1, 2, \ldots, n, \#\}, \delta_n, \{1, 2, \ldots n\}, \{\#\})$. (We have used a set of initial states instead of one. We can add one additional state and use that to replace this set by a single intitial state.)

$$\begin{array}{lcll} \delta(i,j) & = & \{i\} & \text{if } i \neq \# \text{ and } i \neq j \\ \delta(i,i) & = & \{i, \#\} & \text{if } i \neq j \\ \delta(\#, i) & = & \{i\} & \end{array}$$

Here is a pictorial presentation of A_n.



What is the language accepted by this automaton? Say we start at i. We could stay at i for some time reading say x_0, but eventually we have to move to $\#$ and this must be on reading i. The very next move takes us out of $\#$ to the state j where j is the next input letter. Thus, the input read up to the first trip through the accepting state would be of the form x_0ij for some j. Now, we could consume some input, say x_1 at j but then eventually we would have to make a trip through $\#$ and that would require us to read jk for some k

and we would end up at state k and so on. Thus the input read up to the first two trips through the accepting state looks like x_0ijx_1jk. Thus it is the pairs of adjacent letters in the given word that arrange for trips through the accepting set.

Claim 1: Suppose there is a sequence $i_1i_2\dots i_k$ of elements from $\{1, 2, \dots n\}$ such that the words i_1i_2, i_2i_3, ..., i_ki_1 all appear infinitely often in w then w is accepted by A_n.
Proof: Start at i_1. At the first i_1i_2 travel through # to reach i_2, then at the next i_2i_3 travel through # to i_3 and so on. This ensures that # is visited infinitely often. ■

Claim 2: If w is accepted by A_n then there is a sequence $i_1i_2\dots i_k$ of elements from $\{1, 2, \dots n\}$ such that the words i_1i_2, i_2i_3, ..., i_ki_1 appear infinitely often in w.
Proof: Let ρ be an accepting run on w. Suppose j_1 is a state from $\{1, 2, \dots, \#\}$ that appears infinitely often along this run. The run must go from j_1 to # infinitely often. Let j_2 be a state that is reached (in two steps) via # from j_1 infinitely often. Similarly, let j_3 be a state reached from j_2 via # infinitely often and so on. Since there are only finitely many possibilities, some $j_l = j_{l+k}$ for some l and k. Let $i_1, i_2 \dots i_k$ be the sequence $j_l, j_{l+1}, \dots j_{l+k-1}$.
■

Thus we have a characterization of the language accepted by this automaton: Let $IP(w) = \{(i,j) \mid 1 \leq i,j \leq n \text{ and the word } ij \text{ appears infinitely often in } w\}$. Then, w is accepted if and only if the directed graph on $\{1, 2, \dots, n\}$ with edge set $IP(w)$ has a cycle. This fact is used in the proof of the following theorem of Löding, that generalizes an earlier theorem of Michel.

Theorem 3 *(C. Löding) Any Streett automaton $A = (Q, \Sigma, \delta, s, ((E_1, F_1), (E_2, F_2) \dots (E_k, F_k)))$ accepting $\overline{L(A_n)}$ must have at least $n!$ states.*

Proof: Pick any two distinct permutations $i_1i_2\dots i_n$ and $j_1j_2\dots j_n$ of $\{1, 2, \dots, n\}$. The words $w_1 = (i_1i_2\dots i_n\#)^\omega$ and $w_2 = (j_1j_2\dots j_n\#)^\omega$ are not in $L(A_n)$ since $IP(w_1)$ and $IP(w_2)$ have no cycles. Let I_1 (I_2) be the set of states hit infinitely often along some accepting run of A on w_1 (w_2). We shall show that $I_1 \cap I_2$ is empty.

Suppose $q \in I_1 \cap I_2$. Then, following the run on w_1 we can construct a word $x_1 = ui_1i_2\dots i_nv$ such that $q \xrightarrow{x_1} q$ visiting exactly the set of states I_1. Similarly, following the run on the word w_2 we can construct a word $x_2 = u'j_1j_2\dots j_nv'$ such that $q \xrightarrow{x_2} q$ visiting exactly the set of states I_2. Thus there is a run $s \xrightarrow{y} q \xrightarrow{x_1} q \xrightarrow{x_2} q \xrightarrow{x_1} q \xrightarrow{x_2} q$ that visits precisely the set of states $I_1 \cup I_2$ infinitely often. This run is also an accepting run! This follows from the following exercise:

Exercise: Show that if the set X as well as the set Y satisfy the Streett condition $((E_1, F_1), \dots (E_k, F_k))$ then $X \cup Y$ also satisfies this Streett condition.

However, $IP(y(x_1x_2)^\omega)$ has a cycle: Let k be the least number so that $i_k \neq j_k$. Then j_k is some i_l for $l > k$. Similarly, $i_k = j_m$ for some $m > k$. Then $i_ki_{k+1}\dots i_lj_{k+1}\dots j_{m-1}$ forms a cycle. This contradicts the fact that $L(A) = \overline{L(A_n)}$. Thus, $I_1 \cap I_2$ is empty. Thus, the total number of states in A is at least $n!$. ■

Thus, Safra's construction gives an optimal way to transform a NBA into an equivalent Rabin automaton.

Corollary 4 *Any nondeterministic Büchi automaton accepting* $\overline{L(A_n)}$ *has at least* $n!$ *states.*

Proof: A NBA $(Q, \Sigma, \delta, s, F)$ is also the Street automaton $(Q, \Sigma, \delta, s, ((F, Q)))$. ■

Thus, Safras construction as well as the Kupferman-Vardi construction for complementing NBAs are optimal.

2 Transforming acceptance conditions

In this section we shall try to see how to translate automata of one kind into another. Let us recall what we know:

1. By Lemma 5 of Lecture 8, any nondeterministic Müller automaton of size n with m accepting sets can be transformed into a NBA of size $O(m.n)$.
2. Quite trivially, a NRA automaton can be transformed into a NMA with the same number of states. The number of accepting sets however is $O(2^n)$ where n is the number of states of the NRA. This construction transforms deterministic automata into deterministic automata.
3. The same as above for Streett automata.
4. NBAs can be transformed into DRA of size $2^{O(nlogn)}$ and $O(n)$ accepting pairs. This follows from Safra's construction.
5. NBAs can be transformed into DMA with $2^{O(nlogn)}$ states (and double exponential number of accepting sets). This follows from items 2 and 4.
6. By the exercise at the end of Safra's construction in this lecture, any nondeterministic Streett automaton of size n with r accepting pairs can be transformed into an equivalent NBA of size $O(n.2^r)$.
7. A NRA with n states and r accepting pairs can be transformed into an equivalent NBA of size $O(n.r)$.
   Proof: The Büchi automaton, guesses an i and checks whether the word is accepted via the accepting pair (E_i, F_i). This is done by "waiting" for the point from where no state from E_i appears and then verifying that this is indeed the case and also that some state from F_i is visited infinitely often. ■

But here are some transformations that are missing in the list above:

1. Transforming Müller automata into Streett/Rabin automata.

2. Transformations between deterministic Streett and Rabin automata. Note that the last two items in the above list allow transformations between NSA and NRA (since every Büchi automaton can be thought of as a Streett or as a Rabin automaton with one accepting pair). We can combine the $O(n.2^r)$ transformation from Streett to NBA with Safra's construction to get a $O(2^{O(n.2^r log(n.2^r))})$ translation from NSAs to DRAs. But notice that this is double-exponential in the number of accepting pairs. As we shall see, we can do better.

3 Transforming Müller automata into Streett/Rabin automata

This is a rather ingenius construction that will be used again and again. The original ideas behind this construction appear in the work of J.R.Büchi. Our presentation will follow that of Wolfgang Thomas [3].

The idea is to keep track of the history of the states visited. Let A be a Müller automaton with n states. Suppose a run of the NMA automaton on input $a_1 a_2 \ldots a_k \ldots$ is $\rho = q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \ldots \xrightarrow{a_k} q_k \ldots$. The history after the first k moves is $q_0 q_1 \ldots q_k$. Recall that the run is accepting if there is an $X \in \mathcal{F}$, such that $\mathit{inf}(\rho) = X$. This happens if

1. Every state in X is visited infinitely often.

2. There is a N such that for all $i \geq N$ the last $|X|$ distinct states in $q_0 q_1 \ldots q_i$ is the set X. The last k distinct states in $q_0 q_1 \ldots q_i$ is obtained by removing duplicate occurances of any state by deleting all but the right most copy of each state. For example the last 3 distinct states in $pqrsppqqrsssppp$ are $\{p, s, r\}$ as after deleting the duplicate entries we get $qrsp$. This permutation of (some subset of) Q obtained by deleting all but the right most occurance of each state in a state sequence σ is called the LAR (Last Appearance Record) of σ, written $\mathsf{LAR}(\sigma)$.

(Note that condition 2 alone is not enough. The last 3 distinct states of of $r(pq)^i$ is $\{p, q, r\}$ for all i. Thus condition 2 does not ensure that states in X are hit infinitely often.)

This suggests that we construct an automaton whose states are LARs. Our LARs will be permutations Q (that is, every state in Q will appear). Note that $\mathsf{LAR}(q_0 q_1 q_2 \ldots q_{i+1}) = \mathsf{LAR}(\mathsf{LAR}(q_0 q_1 \ldots q_i).q_{i+1})$. This leads us naturally to the following definition for the transition relation: Given a LAR $q_1 q_2 \ldots q_n$ and a letter a, we may move to a state $p_1 p_2 \ldots p_n$ if $p_1 p_2 \ldots p_n = \mathsf{LAR}(q_1 q_2 \ldots q_n p)$ where, $p \in \delta(q_n, a)$. Notice that $p_1 p_2 \ldots p_n = q_1 \ldots q_{i-1} q_{i+1} \ldots q_n p$ when $q_i = p$. The effect of such a transition is to delete p from the current list and append it at the right end. We shall say that *position i is hit* in a transition to mean that the ith state was moved to the right end by this transition. Thus, if $q_0 \xrightarrow{a_1} q_1 \xrightarrow{\cdots}$ is a run in Q then the LAR automaton simulates this run via the run $\mathsf{LAR}(q_0) \xrightarrow{a_1} \mathsf{LAR}(q_0 q_1) \xrightarrow{a_2} \mathsf{LAR}(q_0 q_1 q_2) \ldots$. (Well, we are fudging things here. Some of these are not really permutations of Q. Well, we pad it to the left to get a permutation. I leave it to the reader to figure out what to do.)

For a fixed $X \in \mathcal{F}$, if we use conditions 1 and 2 to describe acceptance in the LAR automaton then we get something that is almost a Rabin accepting pair. Condition 2 says that LARs in which the last $|X|$ entries include states from $Q \setminus X$ are visited only finitely often. Thus, condition 2 is of the form "E_i is hit finitely often". Condition 1 says that every state in X is visited infinitely often. Condition 1 is not quite equivalent to saying "F_i is hit infinitely often ". We shall fix that now. The trick is to record at each move the position from which a state was deleted (and moved to the right end).

Henceforth by an LAR we shall refer to a pair consisting of a permutation of the states of Q and an index i with $1 \leq i \leq n$. Given $(q_1 q_2 \ldots q_n, i)$ and a letter a, we allow a transition $(q_1 q_2 \ldots q_n, i) \xrightarrow{a} (p_1 p_2 \ldots p_n, j)$ if $q_n \xrightarrow{a} p$ and $q_j = p$ and $p_1 p_2 \ldots p_n = q_1 \ldots q_{j-1} q_{j+1} \ldots q_n p$. A run $q_0 \xrightarrow{a_1} q_1 \xrightarrow{q_2} \ldots$ is simulated by this automaton via a run of the form $(f_1, i_1)(f_2, i_2)\ldots$ where each f_i is a permutation of Q. We write $f_i(j)$ for the jth element of the permutation f_i. Then $f_i(n) = q_i$ for all $i \geq 1$ and further $f_i = \mathsf{LAR}(q_1 q_2 \ldots q_i)$.

For each $X_i \in \mathcal{F}$ we shall define a Rabin accepting pair (E_i, F_i) as follows: E_i is the set of states $(q_1 q_2 \ldots q_n, j)$ such that the last $|X_i|$ states of $q_1 q_2 \ldots q_n$ do not form the set X_i. Thus if E_i is hit finitely often along some run then there is an N such that for all $k \geq N$, the state reached along this run is of the form (f_k, j) where the last $|X_i|$ elements of f_k form the set X_i.

F_i consists of all tuples of the form $(f, n - |X_i| + 1)$. Suppose E_i is hit finitely often and F_i is hit infinitely often. Let $x \in X_i$. Pick any state (f_k, j) in the run with $k > N$ (where N is as defined in the previous paragraph). Therefore x appears somewhere among the positions $f_k(n - |X_i| + 1), f_k(n - |X_i| + 2) \ldots f_k(n)$. Notice that when the position j is hit in a transition, states that appear at positions $j + 1$ through n shift to the left by one position. The only way in which a state shifts right is when it is visited and in that case it is moved to the right end of the LAR. Since position $n - |X| + 1$ is hit infinitely often, at each such hit x is shifted to the left by one step. And if it is not visited then it will eventually reach the position $n - |X_i| + 1$. But then we are guaranteed to visit x as this position is hit sometime after this. Thus for all $k \geq N$ there is a $k' > k$ such that x is visited at step k'. Thus each state of X_i is visited infinitely often.

It is easy to check that the converse, that is if $q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \ldots$ accepts via X_i then the corresponding run $(f_0, i_0) \xrightarrow{a_1} (f_1, i_1) \ldots$ satisfies the Rabin condition (E_i, F_i). Thus, the LAR automaton accepts via the pair (E_i, F_i) if and only if the original automaton accepted via the set X_i. Thus we have transformed a Müller automaton into an equivalent Rabin automaton. Moreover this transformation sends deterministic automata to deterministic automata. This also allows us to translate deterministic Müller automata to deterministic Streett automata (simply complement, transform to Rabin and complement again). The number of states is $O(n!)$ and the number of accepting pairs is $O(m)$ where m is the number of sets in $\mathcal{F}$.

We shall now modify this construction to make the number of accepting pairs independent of m (and dependent only on n). At present, E_i ensures requirement 2 while F_i ensures requirement 1 (assuming E_i is hit finitely often). The idea is to redistribute this work.

Let E_i consist of all the pairs of the form (f, j) where $j <= n - |X_i|$. Let F_i consists of all the pairs of the form $(f, n - |X_i| + 1)$ where the last $|X_i|$ states in F_i are the states of X_i. Now, if E_i is hit finitely often along some run, then after some point the positions $1, 2, \ldots n - |X_i|$ are not hit. If further F_i is hit infinitely often then there must a later point at which the last $|X_i|$ states form the set X_i. From here on, since the hit never reaches positions $1, \ldots, n - |X_i|$, it is guaranteed that the last $|X_i|$ states will constitute the set X_i. More over, position $n-|X_i|+1$ is hit infinitely often. Thus, the new Rabin condition (E_i, F_i) also guarantees that the set of states of A hit infinitely often is X_i.

Now, if $|X_i| = |X_j|$ then $E_i = E_j$. Can we combine the sets F_i and F_j together? Yes, because $F_i \cup F_j$ is hit infinitely often if and only if one of F_i or F_j is hit infinitely often. Thus, we can replace (E_i, F_i) and (E_j, F_j) with $(E_i = E_j, F_i \cup F_j)$. Thus, we can obtain a Rabin automaton which has at the most n accepting pairs. Thus, every Müller automaton with n states can be transformed into a Rabin automaton with $O(n!)$ states and $O(n)$ accepting pairs.

Theorem 5 *Let $A = (Q, \Sigma, \delta, s, \mathcal{F})$ be a Müller automaton with n states. Then, there is a Rabin automaton A' accepting the same language as A with $O(n!)$ states and $O(n)$ accepting pairs.*

Proof: Let $Q' = \mathsf{Perm}(Q) \times \{1, 2, \ldots n\}$. Let $s' = (f, n)$ for some fixed permutation f with $f(n) = s$. The transition relation is given by

$$((q_1 q_2 \ldots q_n, i) \xrightarrow{a} (q_1 q_2 \ldots q_{j-1} q_{j+1} \ldots q_n q_j, j) \quad \Longleftrightarrow \quad (q_n, a, q_j) \in \delta$$

The accepting condition consists of a family of pairs $(E_1, F_1), (E_2, F_2) \ldots (E_n, F_n)$ where:

$$\begin{array}{rcl} E_i & = & \{(f, j) \mid j \leq i\} \\ F_i & = & \{(f, i) \mid \{f(i+1), f(i+2) \ldots, f(n)\} \in \mathcal{F}\} \end{array}$$

The correctness of this construction follows from the discussion above. ■

We shall not stop here, but modify this transformation to get a Rabin automaton with a very interesting structure.

3.1 Rabin Chain/Parity Automata

Observe that in any Rabin automaton we may replace the list of accepting pairs

$$(E_1, F_1), (E_2, F_2), \ldots, (E_k, F_k)$$

with

$$(E_1, E_1 \cup F_1), (E_2, E_2 \cup F_2) \ldots (E_k, E_k \cup F_k)$$

without changing the language accepted. In other words, we may always ensure that E is a subset of F in each accepting pair (E, F). Thus, we may replace the accepting pairs used in the proof of theorem 5 by the following:

$$\begin{array}{rcl} E_i & = & \{(f, j) \mid j \leq i\} \\ F_i & = & E_i \cup \{(f, i) \mid \{f(i+1), f(i+2) \ldots, f(n)\} \in \mathcal{F}\} \end{array}$$

Well, this gives us more — we also have $F_i \subseteq E_{i+1}$ for $1 \leq i < n$. Thus the family of Rabin accepting pairs actually form a chain $E_1 \subseteq F_1 \subseteq E_2 \subseteq F_2 \ldots \subseteq E_n \subseteq F_n$. A Rabin condition with this property is called a *Rabin Chain condition.* ,Thus, we have the following theorem:

Theorem 6 *Any Müller automaton can be transformed into an equivalent Rabin Chain automaton with $O(n!)$ states and n accepting pairs.*

Rabin chain automata are also Streett automata! Let A be a Rabin chain automaton with accepting pairs $(E_1, F_1), (E_2, F_2), \ldots, (E_k, F_k)$. Now consider the automaton A' obtained from A be replacing the list of accepting pairs by $(\emptyset, E_1), (F_1, E_2), \ldots, (F_{k-1}, E_k), (F_k, Q)$. A run is accepting in A if and only if it is rejecting in A'. This gives us the following results:

Theorem 7 *Deterministic Rabin chain automata can be complemented without any blow up in state space or accepting pairs.*

In proof, note that the automaton A' accepts the complement.

Theorem 8 *Any Müller automaton can be transformed into an equivalent Street (chain) automaton with $O(n!)$ states and $n + 1$ accepting pairs.*

In proof note, that A' as a Street automaton accepts the same set of words as A accepts as a Rabin automaton.

Parity Automata Here is a alternative way of looking at a Rabin chain condition $E_1 \subseteq F_1 \subseteq E_2 \subseteq F_2 \ldots \subseteq E_k \subseteq F_k$. We shall assume without loss of generality that $F_k = Q$. (Otherwise we may add an additional pair (Q, Q) without altering the language accepted and maintaining the chain property.) Thus we have a chain of $2k$ sets:

$$\begin{array}{ccccccccccccc} E_1 & \subseteq & F_1 & \subseteq & E_2 & \subseteq & F_2 & \subseteq & \ldots & \subseteq & E_k & \subseteq & F_k \\ 1 & & 2 & & 3 & & 4 & & \ldots & & 2k-1 & & 2k \end{array}$$

The index of the set E_i in this list is $2i-1$ while that of F_i is $2i$. With each $q \in Q$, we assign a number $c(q)$ where $c(q)$ is the index of the first set (from the left) in which it appears.

A run ρ is accepted under this Rabin (chain) condition if and only if there is some i such that E_i is hit finitely often (and hence $E_1, F_1, E_2, \ldots, F_{i-1}$ are hit finitely often) and F_i is hit infinitely often (and $E_{i+1}, F_{i+1} \ldots F_k$ are hit infinitely often). This is equivalent to saying that the minimum element of $c(\mathit{inf}(\rho))$ is an even number.

Definition 9 *A* parity automaton *is an automaton $(Q, \Sigma, \delta, s, c)$ where Q, Σ, δ and s are as before and c is a function from Q to the set of natural numbers. A run ρ of a parity automaton is accepting if the minimum element of $c(\mathit{inf}(\rho))$ is an even number.*

We just showed that every Rabin chain automaton can be transformed into an equivalent parity automaton with the same state space. The converse is also true. Let $(Q, \Sigma, \delta, s, c)$ be a parity automaton. Let $i_1 < i_2 < i_3 \ldots < i_k$ be the numbers that appear in the range of c. We may assume without loss of generality that $i_1, i_3, \ldots$ are odd numbers and $i_2, i_4, \ldots$ are even numbers. (Hint: Suppose i_2 is also an odd number, simply set $c(q) = i_1$ whenever $c(q) = i_2$.) Let $E_j = \bigcup_{k \leq 2j-1} c^{-1}(k)$ and $F_j = \bigcup_{k \leq 2j} c^{-1}(k)$.

Exercise: Show that the above construction yields a Rabin chain automaton equivalent to the parity automaton we started with.

To summarize we have shown the following:

Theorem 10 *Any Müller automaton A with n states can be transformed into an equivalent parity automaton with $O(n!)$ states that uses only $O(n)$ numbers to label the vertices. This translation preserves determinism.*

It is also quite evident that deterministic parity automata can be complemented trivially. Just add 1 to $c(q)$ for each q to invert the parity.

References

[1] Christof Löding: *Optimal Bounds for Transformations of ω-automata*, Proceedings of the International Conference on Foundations of Software Techonology and Theoretical Computer Science (FSTTCS) 1999, Springer Lecture Notes in Computer Science 1738, 1999.

[2] S. Safra: *On the complexity of ω-automata*, Proceedings of the 29th FOCS, 1988.

[3] Wolfgang Thomas: *Languages, automata, and logic* In the Handbook of Formal Languages, volume III, pages 389-455. Springer, New York, 1997.