LOGICS SEMINAR LECTURE

GUY LANDO

ABSTRACT. In this note I will present the material of "ω Automata", chapter 1 in Automata, Logic and infinite games, edited by Gradel, Thomas and Wilke, LNCS 2500.

(1) For an ω-automaton $\mathcal{A}$, a run of $\mathcal{A}$ on an ω-word $\alpha = \alpha_1\alpha_2\ldots$ is an infinite state sequence $\varrho = \varrho(0)\varrho(1)\cdots \in Q^\omega$, in which the ω-word passes through in the ω-automaton, which means that the following conditions hold:
   (a) $\varrho(0) = q_I$
   (b) $\varrho(i) \in \delta(q_{i-1}, a_i)$ for $i \geq 1$ if $\mathcal{A}$ is nondeterministic, and $\varrho(i) = \delta(q_{i-1}, a_i)$ for $i \geq 1$ if $\mathcal{A}$ is deterministic.

(2) We will denote ω-runs by ϱ, σ.

(3) $Inf(\varrho)$ the set of symbols occurring infinite number of times in the run ϱ

(4) An ω-automaton is a Buchi automaton if the acceptance component is $Acc := F \subset Q$ and the Buchi acceptance condition is used which states that an ω-word α is accepted if there exists a run ϱ on $\mathcal{A}$, at least one state in F is visited infinitely many times which means that: $Inf(\varrho) \cap F \neq \emptyset$.

(5) For a Buchi automaton $\mathcal{A} = (Q, \Sigma, \delta, q_I, F)$, we can look at it as a finite automaton on finite words and if we start with state p and finish with state q such that our new $F' = \{q\}$ is the acceptance states set, then we denote the regular language generated by this automaton by $W(p, q)$.

(6) Notice that by definition α is accepted in $\mathcal{A}$ if and only if some state q in F occurs infinitely many times, so we can look at the part of α until first occurrence of q and the rest of α which must go back from q to q infinitely many times. So in conclusion we get that: $\alpha \in W(q_I, q) * W(q, q)^\omega$. From this we conclude Theorem 1.5 in the book which states that the Buchi recognizable ω-languages are the ω-languages of the form:

$$L = \bigcup_{i=1}^{k} U_i V_i^\omega, k \in \omega, U_i, V_i \in REG, i = 1, \ldots, k$$

Date: March 4, 2016.

This family of -languages is also called the ω-Kleene closure of the class of regular languages.

(7) We can conclude that each nonempty Buchi recognizable ω-language contains an ultimately periodic word.

(8) Thus we can decide the emptyness problem for Buchi automatons because a Buchi automaton $\mathcal{A}$ has an empty language if and only if it does not have an eventually periodic accepted word if and only if none of the reachable accepted states has a path to itself (a loop). So we decide the problem by finding all reachable accepted states and checking for each one if there is a path from it to itself.

(9) An ω-non-deterministic automaton is a Muller automaton if the acceptance component is $Acc := \mathcal{F} \subset 2^Q$ and the Muller acceptance condition is used which states that an ω-word α is accepted if there exists a run ϱ on $\mathcal{A}$, such that $Inf(\varrho) \in \mathcal{F}$ which means that the set of states occurring infinitely many times is part of $\mathcal{F}$, which also means that the set of states in which the run moves from some position onwards is a set in $\mathcal{F}$.

(10) We now prove the opposite direction. Given a Muller automaton $\mathcal{A} = (Q, \Sigma, \delta, q_I, \mathcal{F})$, we build a Buchi automaton.

We first guess a group $G \in \mathcal{F}$ and for each group G we add copies of G for each subset of G so that we can create a memory which remembers which states of G we visited. Notice that if a word is accepted then there is some G in $\mathcal{F}$ for which the run of the word in the automaton is running inside G from some place.

Our new automaton will simulate the transitions of $\mathcal{A}$ on original states Q but also each time there is a transition towards a state in G, we will also add a transition towards the copy of G corresponding to the empty set and from there the transitions will move only towards other copies of G according to which states of G are visited. In case that after moving to a copy of G, the original run in $\mathcal{A}$ goes back to some state in $Q \setminus G$ then the non-deterministic run in the copies of G dies. Otherwise, the accepting states are the states corresponding to G in the empty set copy of G related to G and they will be visited in the loop only after all states of G were visited by the word in the original automaton and thus if those states will be visited infinitely many times it means that G is on the one hand contained in the set of states which are visited infinitely many times and since there is no transitions from the copies to states of $Q \setminus G$ it also means that G is exactly the set of states which are visited infinitely many times as desired. If $|Q| = m, |\mathcal{F}| = m$ then you can notice we added $m * n * 2^m$ states.

(11) This concludes theorem 1.10 in the book which says that translation from Buchi non deterministic of size n can be done explicitly to n states Muller automaton, and translation from Muller n states and

$|\mathcal{F}| = m$ can be done explicitly to Buchi of at most $n * m * 2^m$ states automaton.

(12) The transformation stated above transforms nondeterministic Buchi automata into nondeterministic Muller automata and conversely. For a given deterministic Buchi automaton the translation yields a deterministic Muller automaton. On the other hand, a deterministic Muller automaton is converted into a nondeterminsitic Buchi automaton. As we shall see later, this nondeterminism cannot in general be avoided.

(13) An ω-non-deterministic automaton is a Rabin automaton if the acceptance component is $Acc := \Omega \subset 2^Q \times 2^Q$ and the Rabin acceptance condition, also called Pairs condition, is used which states that an ω-word α is accepted if there exists a run ϱ on $\mathcal{A}$, such that $\exists(E, F) \in \Omega, Inf(\varrho) \cap E = \emptyset \wedge Inf(\varrho) \cap F \neq \emptyset$ which means that there is some accepting component pair such that the run visits the negative set of states finitely many times while visiting some state of the positive set of states infinitely many times.

(14) An ω-non-deterministic automaton is a Steett automaton if the acceptance component is $Acc := \Omega \subset 2^Q \times 2^Q$ and the Steett acceptance condition, also called complemented Pair condition or fairness condition, is used which states that an ω-word α is accepted if there exists a run ϱ on $\mathcal{A}$, such that $\forall(E, F) \in \Omega, Inf(\varrho) \cap E \neq \emptyset \vee Inf(\varrho) \cap F = \emptyset$ which means that for each accepting component pair if the run visits some state of the right set of states infinitely many times then it must visit some state of the left set of states infinitely many times.

(15) Notice that the condition for Streett is the negation of the Rabin condition and thus a Rabin automaton looked at as a Streett automaton with Streett condition will accept the complement language and vise versa.

(16) Example: automate with states q_I, p and arrow from p to itself with a, from q_I to p and back and q_I to q_I with a, b. To get the language $\{a, b\}^* a^\omega$ with Buchi we define $F = \{p\}$, with Muller we define $\mathcal{F} = \{\{p\}\}$, with Rabin we define $\Omega = \{\{q_I\}, \{p\}\}$, with Streett we define $\Omega = \{\{\}, \{q_I\}\}$.

(17) Notice that any Rabin automaton can be transfered to Muller automaton automaton by building acceptance component by taking all sets such that for some pair the set intersects the right set of the pair and does not intersect the left side of the pair. For Streett automaton take sets which for each pair, do not intersect the right set of the pair or if intersects it then must also interesect the left set of the pair.

(18) Any Buchi automaton can be transformed to Rabin automaton with acceptance component $\Omega = \{\{\}, F\}$. Any Buchi automaton can be transformed to Streett automaton with acceptance component $\Omega = \{F, Q\}$.

(19) As a conclusion we get that any Muller automaton can be transferred to Rabin or street automaton by first transferring it to Buchi and then to Rabin or Steett.
(20) An ω-non-deterministic automaton is a parity automaton if the acceptance component is a coloring $Acc := c : Q \to \{0, \dots, k\}$ and the parity acceptance condition is used which states that an ω-word α is accepted if there exists a run ϱ on $\mathcal{A}$, such that $min\{c(q) : q \in Inf(\varrho)\}$ is even, which means that the minimal color of a state which occurs infinitely many times in the run is even.
(21) Example: automate with states q_I, p and arrow from p to itself with a, from q_I to p and back and q_I to q_I with a, b. To get the language $\{a, b\}^* a^\omega$ with Buchi we define $c(q_I) = 1, c(p) = 2$
(22) For a Rabin automaton with acceptance component $\Omega = \{(E_i, F_i) : 1 \leq i \leq k\}$ such that $E_1 \subset F_1 \subset E_2 \subset F_2 \subset \dots$ we can define a coloring $c(E_i) = 2 * i - 1, c(F_i \setminus E_i) = 2 * i$ and we get an equivalent paring automaton.
(23) Notice also that a Rabin automaton with acceptance component $\Omega = \{(\emptyset, F)\}$ has the chain condition stated previously and thus it is equivalent to a paring automaton. Any no-deterministic Muller, Rabin, Steett automaton can be transferred to a non deterministic Buchi automaton constructed before and then transferred to a Rabin automaton with acceptance component $\Omega = \{(\emptyset, F)\}$ which can be transferred as stated, to a parity automaton.
(24) From previous statements we can now conclude theorem 1.19 in the book which states that: Nondeterministic Buchi automata, Muller automata, Rabin automata, Streett automata, and parity automata are all equivalent in expressive power, i.e. they recognize the same ω-languages. We call the class of those languages the class ω-KC(REG),i.e. the ω-Kleene closure of the class of regular languages. The ω-languages in this class are commonly referred to as the regular ω-languages, denoted by ω-REG
(25) Two questions rise: 1)Is there a deterministic automatons class which accepts exactly the ω-REG languages? because we like deterministic automatons more. 2)Is the ω-REG class closed under complement? Both questions have positive answers. The complement problem can be dealt with by determinization and thus we now deal with the deterministic versions of the automatons we dealt with so far. In chapter 3 in the book it is proved that the Muller deterministic automatons accept exactly the ω-REG class languages. And also we will see that the deterministic Muller, Rabin, Parity, Steett automatons are equivalent expressively, while the Buchi deterministic automaton is weaker.
(26) Here is a proof that Buchi deterministic automaton is weaker by proving that some non-deterministic Buchi automaton can't be converted to a deterministic one. Look at the non-deterministic Buchi

automaton which has the language $\{a,b\}^*a^\omega$ (state q has arrow with a,b to itself and a,b (or just a) to state p, state p has only arrow a to itself, F is the state p). Then there is this non-deterministic Buchi automaton for the language but there is no deterministic Buchi automaton for the language because if there was one $\mathcal{A}$ then lets look at the word a^ω. It is accepted by the language and thus by the automaton and thus there is some accepting state q_1 of the automaton which is entered by the run of a^ω on $\mathcal{A}$ at some position n_1 which means $a^{n_1} * a^\omega$ is accepted by $\mathcal{A}$ and a^{n_1} runs into q_1. Now, by the definition of the language $a^{n_1} * b * a^\omega$ is also accepted by the automaton and thus there is some accepting state q_2 of $\mathcal{A}$ and position n_2 such that $a^{n_1} * b * a^{n_2} a^\omega$ is the accepted word and $a^{n_1} * b * a^{n_2}$ runs into q_2. Now, the important part is that since $\mathcal{A}$ is deterministic it means that there is an accepting run which moves though both q_1 and q_2. Now by definition of the language $a^{n_1} * b * a^{n_2} * b * a^\omega$ is also accepted and so on. By this construction we get a word with infinitely many b which visites states from F infinitely many times (and by pidgeon hole principle there is a state in F visited infinitely many times since F is finite and the run is infinite) and thus this word is accepted by the automaton which is a contradiction to the language of the automaton which does not accept words with infinitely many b.

(27) Simulate deterministic Muller by deterministic Rabin: (record the past is the technique used previously where you fill visited sets without remembering the order), here we will use LAR (last appearance record) which is a permutation on Q together with a pointer @ and it allows to also know the which states were visited last. Given Muller automaton with usual notations, we construct a new Rabin automaton where the states are the collection of permutations on $Q \cup \{@\}$. If $|Q| = n$, we name the states of Q $q, \ldots, k$ where 1 is the initial state and start from state $(@, k, k-1, \ldots, 1)$ (probably it doesn't really matter where we start from as long as the initial state of original automaton is at the right). If state m moves to state s in original automaton and currently the rightmost state in the currently visited permutation is m then we can transition to the permutation where s moves to the rightmost position and the pointer @ moves to the original position of s and all states which were on the right of s shift one position left in the permutation.

Notice (first statement) that by this construction we get that states which are visited finitely often will eventually be placed on the left part of the permutation, from the left of the pointer, so the states on the right of the pointer infinitely many times are states which are visited infinitely many times in the original automaton. Also notice (second statement) that as long as the transitions in the original automaton are between states which are at the right most k places in the permutation, then the pointer will never go before the first k+1 places and all the changes in the permutation will occur in the last k+1 places and other states will be untouched. This means that if there are k states in the original automaton which are visited infinitely often by a run then first of all from the first statement it means that from some move onwards the pointer will be always in the right most k+1 places at most (maybe less places), and from the last statement we get that it will visit infinitely many times the k+1 place (or otherwise the size of infinitely many times visited states in original automaton would be smaller than k).

Thus we get as conclusion Lemma 1.21 in the book and if we define the accepting pairs of the Rabin automaton to be pairs for each $1 \leq i \leq n$ such that the i pair consists of negative set of permutations where the pointer only appears in the left most i places and the positive set is the permutations where the pointer only appears in the left most i places (this is a trick to turn this to easily convertible to parity automaton) or appears exactly on the i+1 place from the left and the states on the right of the pointer together create a set which appears in the original automaton accepting component.

(28) Another, easier proof (not from the book): If $\mathcal{F} = \{F_1, \ldots, F_m\}$ then $Q' = Q \times 2^{F_1} \times \cdots \times 2^{F_k}$ and the idea is to simulate original automaton on Q while saving in memory (represented by the new states) which states of the accepting sets are visited and when some accepting set visited in full then empty it and start filling it again. The accepting pairs are pairs where the positive pair is one in which there is some accept set F_i is filled in full in memory and the additional state of Q is also inside this accepting state, the negative set is when the additional state of Q is not in the F_i of the positive set (each pair deals exclusively with some unique i). The positive condition makes sure that the states visited infinitely many times in the original automaton include the states of F_i and the negative condition makes sure that starting from some place no other state except the states of F_i is visited and thus F_i is exactly all the states visited infinitely many times and since F_i is in $\mathcal{F}$ it means that the word is also accepted in the original automaton. Easy to see the other direction.

(29) Lower bounds lemma 1 proof: assume the union run IS accepting, then there is some pair of the automaton which have its positive set intersect the Inf union and the negative set has no intersection with the Inf union. Thus one of the Inf's in the union intersects the positive set while none of the Inf's intersect the negative set and thus the Inf which intersects the positive set allows the run to conform to the rabin condition and thus be accepting in contradiction. Same proof for the second lemma from duality between rabin and streett conditions.

(30) Non-determinitstic Buchi to Deterministic Rabin notes:

(a) An accepting word of $\mathcal{A}_n$ must visit q_0 infinitely many time and by the pidgeon hole principle we get that there eventually is some loop of unique states moved through to visit q_0 and this loop together with the fact that the automaton is non-deterministic, proved the characterization of the languages of $\mathcal{A}_n$.

(b) The symbols encoding is done because we work with automatons over finite language and we want to use the infinite family of automatons over the same finite language. The encoding requires to add states corresponding to the encoding, yet this can be done by adding at most O(n) states by adding n-1 states instead of the arrow from q_n to q_0 and using those states to parse the other arrows to q_0 and similarly handle the other direction of the arrows from q_0.

(c) The statement can be proven for complement language using Streett automaton because it is the complementary automaton of the Rabin automaton.

(d) The hash symbol appearance fixes the location of the run in the sense that without the hash, we can find a non-deterministic run that could be at this stage either in q_0 or not in q_0 but if we met a hash then we are surely not in q_0.

(e) Any permutation of 1,..,n can visit q_0 in $\mathcal{A}_n$ at most once and it must return from q_0 to a different state than the one it came from to q_0. Thus in the run of the word $i_1i_2...i_n\#i_1i_2...i_n\#$, if the first permutation before the first hash return from q_0 to state i_{j_1} then if the second permutation visits q_0 then it must visit it from state i_{j_1} and thus if it returns to state i_{j_2} then $j_2 > j_1$ and from this, since the states set is finite, we can conclude that the words $(i_1i_2...i_n\#)^\omega$, $(j_1j_2...j_n\#)^\omega$ are not accepted by $\mathcal{A}_n$.

(f) If some Streett automaton accepts the complement language then it accepts $\alpha = (i_1i_2...i_n\#)^\omega$, $\beta = (j_1j_2...j_n\#)^\omega$ (we choose different permutations) and thus there accepting runs $\varrho_\alpha, \varrho_\beta$. If we show that $Inf(\varrho_\alpha) \cap Inf(\varrho_\beta) = \emptyset$ then since there are $n!$ permutations, we can conclude as desired that there are atleast n! states (since the Inf sets are not empty and coprime).
By contradiction, if there is some $q \in Inf(\varrho_\alpha) \cap Inf(\varrho_\beta)$ then q is visited infinitely often by ϱ_α which means that there is some word v moving over states in ϱ_α from q back to q and word w having same property for ϱ_β. Suppose that u is some word over the run ϱ_α or ϱ_β which brings one of the runs from initial state to q.
Look now at the word $\gamma = u(vw)^\omega$. By the definition of γ we know that $Inf(\varrho_\gamma) = Inf(\varrho_\alpha) \cup Inf(\varrho_\beta)$ and thus by lemma states at (29) we get that ϱ_γ is an accepting run since $\varrho_\alpha, \varrho_\beta$

are accepting runs. We showed that γ is in the complement language.
Notice, once again that the characterization of the language of $\mathcal{A}_n$ (in (a)) does not care of where the pairs, creating the loop, appear. The only thing that matters is that those pairs will appear in a loop somewhere and it does not matter if other words appear between them.
Now look at the word $\gamma = u(vw)^\omega$ and specifically at $(vw)^\omega$. This is a loop so the only thing needed in order for $\mathcal{A}_n$ to accept it is for vw to contain one loop of pairs. The only condition we needed on v, w was that they would be paths in the runs from q to q so we can choose to take them such that $|v|, |w| > 2n + 2$ which means from the definition of α and β that $i_1 i_2 ... i_n$ is a substring of v and $j_1, j_2 ..._n$ is a substring of w. Since α and β were constructed using different permutations it means that starting from some position, $i_1 i_2 ... i_n$ and $j_1, j_2 ..._n$ are different which means that there is some $m >= 1$ such that for all $s < m$ it holds that $i_s = j_s$ while $i_m \neq j_m$. But those are two permutations consisting of same elements thus there must be some $k', l' > m$ such that $i_m = j'_k, j_m = i'_l$ and this proves that the word vw consists of the pairs loop: $i_m, i_{m+1}, ..., i_{l'-1}, i_{l'}, j_m, j_{m+1}, ..., j_{k'-1}, j_{k'}$ satisfying the charactarization of the language of $\mathcal{A}_n$ and thus $\gamma = u(vw)^\omega$ is in $\mathcal{A}_n$, but we already proved it is in the complement of $\mathcal{A}_n$ and this gives the desired contradiction showing that $Inf(\varrho_\alpha) \cap Inf(\varrho_\beta) = \emptyset$ as desired.

(31) Determinitstic Street to Deterministic Rabin notes:
   (a) Define the set of deterministic Street automatons: $\mathcal{A}_n$ such that $\mathcal{A}_n$ has 4 states for each i¡=n: 1,-1,i,-i with transitions from i to -j when receiving j and to j from -i when receiving j, with initial state -1. Accepting pairs are with negative set i and positive set -i.
   (b) For a word α denote $even(\alpha) = Inf(\alpha(0)\alpha(2)\alpha(4)...), odd(\alpha) = \alpha(1)\alpha(3)\alpha(5)....$
   (c) Notice that on odd states in the run we are in a state named by a positive number while in even states in the run we are in a state named by a negative number and thus we can conclude that a word is accepted if and only if $odd(\alpha) \subseteq even(\alpha)$. From this it follows that any move by a word of even length from the initial position, will not change the accepting language of the automaton.
   (d) To get a finite language, encode the symbols similarly to the encoding in previous proof.
   (e) Proof by induction on n that any deterministic rabin automaton accepting the language $L(\mathcal{A}_n)$ has atleast $n!$ states.

(f) For n=2, any automaton recognizing a proper non empty set of words needs 2 states thus statement holds.

(g) Assume for n-1 and lets prove for n. Let $\mathcal{A} = (Q, \Sigma_n, q_0, \delta, \Omega)$ be a Rabin deterministic automaton accepting $L(\mathcal{A}_n)$. Let Q_{even} be the states that can be reached from q_0 by reading a prefix of even length.

For every $i \in \{1, ..., n\}$ and every $q \in Q_{even} \setminus \{i, -i\}$ define deterministic Rabin automaton $\mathcal{A}_i^q$ over $\Sigma_n \setminus \{i\}$ by removing the two states $i, -i$ and all arrows which had $i, -i$ and setting the initial state to q. Notice that this did not affect the other states and up to states name changing we created an automaton isomorphic to $\mathcal{A}_{n-1}$ and the initial state change does not affect the language as previously explained so by induction assumption the created automaton $\mathcal{A}_i^q$ has $(n-1)!$ states.

Since the initial state can be changed by moving with a word of even length, we can take a strongly connected componet of $\mathcal{A}_i^q$ which doesn't have any other strongly connected component reachable from it, and move by an even word into this strongly connected component and then the induction hypothesis holds for this component because since we start in it and it doesn't reach any other component it means that the language of this component is $L(\mathcal{A}_{n-1})$ and thus this component has $(n-1)!$ states. So we proved that $\mathcal{A}_i^q$ has a strongly connected component with $(n-1)!$ states.

The final step of the proof will be constructing a word α_i for each i with a run such that they have coprime Inf sets of size atleast (n-1)! which proves the statement because $|Q| \geq \Sigma_{i=1}^{n} |Inf(\varrho_{\alpha_i}| \geq n * (n-1)! = n!$.

For $i \in \{1, \ldots, n\}$, we construct the word α_i as follows. First take a $u_0 \in (\Sigma_n \setminus \{i\})^*$ such that u_0 has even length and contains every letter from $\Sigma_n \setminus \{i\}$ on an even and on an odd position. Also $\mathcal{A}_i^{q_0}$ should visit at least (n-1)! states while reading u_0 and this is possible since we shown that $\mathcal{A}_i^{q_0}$ has a strongly connected component of size at least (n-1)!. Let q_1 be the state reached by $\mathcal{A}_i^{q_0}$ after having read the word $u_0 ij$, where j is different from i. Then we choose a word $u_1 \in (\Sigma_n \setminus \{i\})^*$ with the same properties as u_0, using $\mathcal{A}_i^{q_1}$ instead of $\mathcal{A}_i^{q_0}$. This means u_1 has even length, contains every letter from $\Sigma_n \setminus \{i\}$ on an even and on an odd position and $\mathcal{A}_i^{q_1}$ visits at least (n-1)! states while reading u_1.

Repeating this gets us a word $\alpha_i = u_0 ij u_1 ij u_2 ij....$ From the construction it follows that $even(\alpha_i) = \{1, ..., n\} \setminus \{i\}$ and $odd(\alpha_i) = \{1, ..., n\}$ and therefore by characterization of $L(\mathcal{A}_n)$ we have that $\alpha_i \notin L(\mathcal{A}_n)$. We also get by the construction that the specified run of α_i in $\mathcal{A}$ has $|Inf(\alpha_i)| \geq (n-1)!$.

We are only left to show that the Inf for different α_i are coprime. By contradiction, if there is some value in the Inf intersection of Inf of α_i, α_j then by using the intersection state we can create loops going through all the states such that the even set and odd set of the new word will be $\{1, .., n\}$, which means by cheractarization of $L(\mathcal{A}_n)$ that this word will be accepted, while the Inf of this word uses the Inf's of the two original words and thus is their union and this is a contradiction because in a Rabin automaton by previous lemma a word which has an Inf set which is a Union of Infs of rejecting words should reject.
The proof doesn't appear in chapter 1 in the book. It can be found on page 48 in `http://www.automata.rwth-aachen.de/~loeding/diploma_loeding.pdf`

(32) Simulate Staiger and Wagner condition automaton by Buchi automaton (similarly to how the Muller was simulated) by remembering which states were visited and once all of F was visited, can turn into accepting hole state. The memory requires exponential blowup of states ($Q' = Q \cup 2^Q$).
(33) Weak condition 1 can be simulated by Buchi automaton without an exponential blowup by turning the set F into accepting holes.
(34) Weak condition 1' can be simulated by Buchi automaton without an exponential blowup by turning the set $Q \setminus F$ into non-accepting holes and making the states of F accepting states.
(35) Even the strongest Staiger and Wagner weak acceptance condition can't create automatons equivalent to some Buchi automaton. For example there is a Buchi automaton accepting only words with infinitely many b's where the language is a,b (states q, p with arrow from q to p with b on arrow and arrow p to q with a on arrow and arrow from q to q with a on it and from p to p with b on it, F contains p). If there was a weak condition automaton accepting this language, we will show contradiction. Suppose it has n states. Then it would accept $(a^{n+1}b)^\omega$ so by the weak condition there is some k when after $(a^{n+1}b)^k$, the run already visited all the states it will ever visit. Since there is a total of n states then by pidgeon hole principle the run assumes a loop in the run on the following a^{n+1} and thus we can have a run on $(a^{n+1}b)^k * a^\omega$ which covers same states as the original run on $(a^{n+1}b)^\omega$ and thus it would be accepted by the weak condition which is a contradiction since it does not have infinitely many b's.

Raymond and Beverly Sackler School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel
E-mail address: `guylando@post.tau.ac.il`