Advanced Automata Theory 6
Automata on Infinite Sequences

Frank Stephan
Department of Computer Science
Department of Mathematics
National University of Singapore
fstephan@comp.nus.edu.sg
Types of Infinite Games

Survival Game: Anke and Boris move alternately in an infinite graph \((V, E)\) where Anke starts in node \(s \in V\). Anke wins play in game \((V, E, s)\) if the play is infinite and Boris wins if the play if it is finite and ends in a deadend where players cannot go on.

Update Game: Game \((V, E, s, W)\) played on finite graph \((V, E)\) with starting node \(s\) and special nodes \(W\). Anke wins play in game if the play is infinite and every node in \(W\) is visited infinitely often. Boris wins if there is a node \(w\) in \(W\) visited only finitely often.

Büchi Game: Game \((V, E, s, W)\) played on finite graph \((V, E)\) with starting node \(s\) and special nodes \(W\). Anke wins play in game if the play is infinite and some node in \(W\) is visited infinitely often. Boris wins if no node in \(W\) is visited infinitely often.
Boris has a memoryless winning strategy for the player.

<table>
<thead>
<tr>
<th>Nodes</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boris’ Moves</td>
<td>-</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>-</td>
<td>1</td>
<td>2</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>Half-moves remaining when Boris’ turn</td>
<td>-</td>
<td>3</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>5</td>
<td>-</td>
</tr>
<tr>
<td>Half-moves remaining when Anke’s turn</td>
<td>8</td>
<td>8</td>
<td>2</td>
<td>6</td>
<td>-</td>
<td>0</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
There is an algorithm which can check which player wins a survival game (when playing optimally).

Let $f(v, \text{Anke}) = f(v, \text{Boris}) = 0$ for nodes without successor.

Let $f(v, \text{Anke}) = n$ for least $n$ with $f(w, \text{Boris}) < n$ for all successors $w$ of $v$.

Let $f(v, \text{Boris}) = n$ for least $n$ with $f(w, \text{Anke}) < n$ for some successor $w$ of $v$.

Let $f(v, p) = \infty$ whenever $-f(v, p) \leq 2 \times |V|$.

If $f(s, \text{Anke}) = \infty$ then Anke has winning strategy by moving from each node $v$ with $f(v, \text{Anke}) = \infty$ to a successor $w$ with $f(w, \text{Boris}) = \infty$.

If $f(s, \text{Anke}) < \infty$ then Boris has winning strategy by moving from each node $v$ with $f(v, \text{Boris}) < \infty$ to a successor $w$ with $f(w, \text{Anke}) < f(v, \text{Boris})$. 

Repetition 3
Anke has a winning strategy for this update game but no memoryless one.
If $\text{maxval}$ is even and one can always go from $n$ to $n$ and to $\min\{\text{maxval}, n + 1\}$ and to 0 then Anke has a winning strategy for this parity game.
Sets of Infinite Sequences

Real Numbers have no finite representation; infinite words might a way to represent them.

\[ b_0b_1b_2 \ldots \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}^\omega \] can be used to represent the sum over all \( b_k \cdot 10^{-k-1} \) which permits to represent all real numbers between 0 and 1.

Wanted: Unique representation where only one of 0018999999999\ldots and 00190000000\ldots appears.

Goal: Make automaton which recognises those decimal strings which have some digit different from 9 infinitely often.
Büchi Automaton

Automaton \((Q, \Sigma, \delta, s, F)\) with \(Q, \Sigma, \delta, s, F\) being defined as a usual non-deterministic automaton but a different semantic for dealing with infinite words.

Given an infinite word \(b_0b_1b_2 \ldots \in \Sigma^\omega\), a run is a sequence \(q_0q_1q_2 \ldots \in Q^\omega\) of states such that \(q_0 = s\) and \((q_k, b_k, q_{k+1}) \in \delta\) for all \(k\). Let

\[ U = \{p \in Q : \exists \infty k [q_k = p] \} \]

be the set of infinitely often visited states on this run. The run is accepting iff \(U \cap F \neq \emptyset\). The Büchi automaton accepts an \(\omega\)-word iff it has an accepting run on this \(\omega\)-word, otherwise it rejects the \(\omega\)-word.
The following deterministic Büchi automaton accepts all \( \omega \)-words of reals between 0 and 1 which are not almost always 9.

Here a Büchi automaton is called deterministic iff for every \( p \in Q \) and \( a \in \Sigma \) there is at most one \( q \in Q \) with \( (p, a, q) \in \delta \); in this case one also writes \( \delta(p, a) = q \).

This automaton goes infinitely often through the accepting state \( t \) iff there is infinitely often one of the digits \( 0, 1, 2, 3, 4, 5, 6, 7, 8 \) and therefore the word is not of the form \( w9^\omega \).
Power of Non-Determinism

Let $L$ contain all $\omega$-words which from some point onwards have only 9s, so $L = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}^* \cdot 9^\omega$. Then some non-deterministic Büchi automaton recognises $L$. 

Advanced Automata Theory 6 Automata on Infinite Sequences – p. 10
Limitation of Determinism

No deterministic Büchi automaton recognises the language $L$ from the last slide.

So assume by way of contradiction that $(Q, \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}, \delta, s, F)$ would recognise $L$. Now one searches inductively for strings of the form $\sigma_0, \sigma_1, \ldots \in 09^*$ such that $\delta(s, \sigma_0\sigma_1\ldots\sigma_n) \in F$ for all $n$.

If all $\sigma_n$ can be found then $\sigma_0\sigma_1\ldots$ is a sequence which infinitely many 0 accepted by the Büchi automaton else there is an $n$ such that the sequence $\sigma_0\sigma_1\ldots\sigma_{n-1}09^\omega$ is not accepted by the Büchi automaton although it is in $L$. 
Exercises (Quiz)

If $L$ and $H$ are languages of $\omega$-words recognised by deterministic Büchi automata then also $L \cup H$ and $L \cap H$ are recognised by deterministic Büchi automata.

Make a non-deterministic Büchi automaton which recognises the language of all $\omega$-words in which exactly two digits from $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ occur infinitely often. Describe how the automaton is build; it is not necessary to make a complete diagram of its states.
Assume that \((Q_L, \Sigma, \delta_L, s_L, F_L)\) recognises \(L\) and \((Q_H, \Sigma, \delta_H, s_H, F_H)\) recognises \(H\).

Now let \(Q = Q_L \times Q_H \times \{10, 01, 11\}\) and for each \((q_L, q_H) \in Q_L \times Q_H\) and \(a \in \Sigma\) let
\[
\delta((q_L, q_H, r), a) = (\delta_L(q_L, a), \delta_H(q_H, a), r')\]
where \(r' = r\) if \(q_L \notin F_L\) and \(q_H \notin F_H\) and \(r' = F_L(q_L)F_H(q_H)\) if at least one of these bits is \(1\).

For the union, \((q_L, q_H, r) \in F\) if either \(q_L \in F_L\) or \(q_H \in F_H\). For the intersection, \((q_L, q_H, r) \in F\) iff \(q_L \in F_L\) and the second bit or \(r\) is \(1\) or \(q_H \in F_H\) and the first bit of \(r\) is \(1\).
Example of Intersection

$L$ is the set of all $\omega$-words containing infinitely many even digits.

$H$ is the set of all $\omega$-words containing infinitely often either 0 or 5.

Büchi Automaton for $L$ consists of two states $s_L$, $t_L$ where the automaton goes to $t_L$ iff it has just seen an even digit and to $s_L$ iff it has just seen an odd digit. $F_L = \{t_L\}$.

Büchi Automaton for $H$ consists of two states $s_H$, $t_H$ where the automaton goes to $t_H$ iff it has just seen 0 or 5 and to $s_H$ otherwise. $F_H = \{t_H\}$.

Intersection $L \cap H$ contains all $\omega$-words where infinitely many even digits and also infinitely many digits from 0, 5 appear in the $\omega$-word.
The Intersection Automaton

Product automaton has states from
\( \{s_L, t_L\} \times \{s_H, t_H\} \times \{01, 10, 11\} \).

How to determine the successor of \((q_L, q_H, r)\) on input \(a\).

Let \(r' = r\) for \((s_L, s_H, r)\), \(r' = 10\) for \((t_L, s_H, r)\), \(r' = 01\) for \((s_L, t_H, r)\) and \(r' = 11\) for \((t_L, t_H, r)\).

\[
\delta((q_L, q_H, r), 0) = (t_L, t_H, r'); \quad \delta((q_L, q_H, r), 5) = (s_L, t_H, r');
\]

\[
\delta((q_L, q_H, r), a) = (t_L, s_H, r') \quad \text{for } a \in \{2, 4, 6, 8\};
\]

\[
\delta((q_L, q_H, r), a) = (s_L, s_H, r') \quad \text{for } a \in \{1, 3, 7, 9\}.
\]

Starting state is \((s_L, s_H, 11)\).

\(F\) contains all nodes of form \((t_L, q_H, x1)\) and of form \((q_L, t_H, 1x)\). Here \(q_L, q_H\) are any states in the corresponding automata and \(x\) is any bit 0 or 1.
Consider the following automaton $B_{i,j}$ with $i \neq j$.

Now make a central starting node $s$ connected to cycles of three nodes $q_{i,j}, r_{i,j}, t_{i,j}$ for all pairs of distinct digits $i, j$ as in the example above. The nodes $t_{i,j}$ are the only accepting ones.
**Characterising Languages**

**Theorem** [Büchi 1960]
The following are equivalent for a language $L$ of $\omega$-words:

(a) $L$ is recognised by a non-deterministic Büchi automaton;

(b) $L = \bigcup_{m \in \{1, \ldots, n\}} A_m B^\omega_m$ for some $n$ and $2n$ regular languages $A_1, B_1, \ldots, A_n, B_n$.

Here $B^\omega_m$ is the concatenation of infinitely many non-empty strings from $B_m$. 
Direction One

Assume that non-deterministic Büchi automaton 
\((Q, \Sigma, \delta, s, F)\) recognises \(L\). Assume that 
\(F = \{p_1, p_2, \ldots, p_n\}\), let \(A_m\) the set of all words on which, 
from starting state \(s\) the automaton can end up at \(p_m\) and 
let \(B_m\) the set of all nonempty words on which the 
automaton can go from \(p_m\) to \(p_m\).

If an \(\omega\)-word is in \(L\) then the Büchi automaton has a run on 
it which goes infinitely often through one \(p_m\).
**Direction Two**

Assume that \( L = A_1 \cdot B_1^\omega \cup A_2 \cdot B_2^\omega \cup \ldots \cup A_m \cdot B_m^\omega \). Assume that each language \( A_m \) is recognised by the nfa \((N_{2m-1}, \Sigma, \delta_{2m-1}, s_{2m-1}, F_{2m-1})\) and each language \( B_m \) is recognised by the nfa \((N_{2m}, \Sigma, \delta_{2m}, s_{2m}, F_{2m})\).

Now let \( N_0 = \{s_0\} \cup N_1 \cup N_2 \cup \ldots \cup N_{2n} \) where all these sets are considered to be disjoint. The start symbol of the new automaton is \( s_0 \).

Furthermore, let \( \delta_0 \) be \( \delta_1 \cup \delta_2 \cup \ldots \cup \delta_{2n} \) plus the following transitions for each \( a \in \Sigma \): first, \((s_0, a, q)\) if there is an \( m \) such that \((s_{2m-1}, a, q) \in \delta_{2m-1}\); second, \((s_0, a, q)\) if there is an \( m \) such that \( \varepsilon \in A_m \) and \((s_{2m}, a, q) \in \delta_{2m}\); third, \((s_0, a, s_{2m})\) if \( a \in A_m \); fourth, \((q, a, s_{2m})\), if there are \( m \) and \( p \in F_{2m-1} \) with \( q \in N_{2m-1} \) and \((q, a, p) \in \delta_{2m-1}\); fifth, \((q, a, s_{2m})\), if there are \( m \) and \( p \in F_{2m} \) with \( q \in N_{2m} \) and \((q, a, p) \in \delta_{2m}\).
Continuation

Now \(\{s_2, s_4, s_6, \ldots, s_{2n}\}\) is the set \(F_0\) of the final states of the Büchi automaton \((N_0, \Sigma, \delta_0, s_0, F_0)\). That is, this Büchi automaton accepts an \(\omega\)-word iff there is a run of the automaton which goes infinitely often through a node of the form \(s_{2m}\); by the way how \(\delta_0\) is defined, this is equivalent to saying that the given \(\omega\)-word is in \(A_m \cdot B^\omega_m\).
A Muller automaton \((Q, \Sigma, \delta, s, G)\) consists of a set of states \(Q\), an alphabet \(\Sigma\), a transition relation \(\delta\), a starting state \(s\) and a set \(G\) of subsets of \(Q\). A run of the Muller automaton on an \(\omega\)-word \(b_0b_1b_2\ldots \in \Sigma^{\omega}\) is a sequence \(q_0q_1q_2\ldots\) with \(q_0 = s\) and \((q_k, b_k, q_{k+1}) \in \delta\) for all \(k\). A run of the Muller automaton is accepting iff the set \(U\) of infinitely often visited states satisfies \(U \in G\). The Muller automaton accepts the \(\omega\)-word \(b_0b_1b_2\ldots\) iff it has an accepting run on it.

A Muller automaton is deterministic iff the relation \(\delta\) is a function, that is, for each \(p \in Q\) and \(a \in \Sigma\) there is at most one \(q \in Q\) with \((p, a, q) \in \delta\).
Example

The language of all $\omega$-words of the form $w_9^\omega$ is recognised by the deterministic Muller automaton

$$(\{s, t\}, \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}, \delta, s, \{\{s\}\}),$$

where $\delta(s, a) = t$ for $a < 9$, $\delta(s, 9) = s$ and $\delta(t, a) = s$ for all $a$.

The following diagramme illustrates the Muller automaton:

$G = \{\{s\}\}$
Characterisations

**Theorem** [McNaughton 1966, Safra 1988]
The following conditions are equivalent for an $\omega$-language $L$.
(a) $L$ is recognised by a non-deterministic Büchi automaton;
(b) $L$ is recognised by a deterministic Muller automaton;
(c) $L$ is recognised by a non-deterministic Muller automaton.

**Application**
If a language $L$ is recognised by a non-deterministic Büchi automaton, so is its complement. There is an algorithm which constructs from a Büchi automaton for $L$ a Büchi automaton for its complement. The number of states grows exponentially.
Deterministic Automata

Exercise 6.6
Make a deterministic Büchi automaton which recognises the language \( L \) of all \( \omega \)-words containing all of the digits 0, 1, 2 infinitely often. Is it possible to recognise the complement of \( L \) with a deterministic Büchi automaton?

Exercise 6.8
Make a deterministic Muller automaton recognising the complement of the language \( L \) from Exercise 6.6. That is, the Muller automaton should accept all \( \omega \)-words in which one or two of the digits 0, 1, 2 occur only finitely often.
Exercise 6.11
Consider $L_h = \{b_0 b_1 b_2 \ldots \exists^\infty m [b_m = b_{m+h}]\}$ over alphabet $\{0, 1, 2\}$. Make a non-deterministic Büchi automaton and a deterministic Muller automaton to recognise $L_h$ and a non-deterministic Büchi automaton to recognise the complement of $L_h$. How many states do these automata have?
What to Learn for the Examination

Please revise Chapters 1-6. Learn the three lower levels of the Chomsky hierarchy: regular, context-free, context-sensitive. Which of these is closed under union, intersection, complement and concatenation? Learn how to modify the corresponding grammars and also how counter example could look like. Train making dfas, nfas, regular expressions and grammars for various sample languages. Revise finite and infinite games on finite graphs. Make some examples of automatic functions. Do the exercises of Section 6; you can hand them in early to get them checked by email or consultation in the case that you are not sure whether you got them right.

Read lecture notes on http://www.comp.nus.edu.sg/~fstephan/.