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

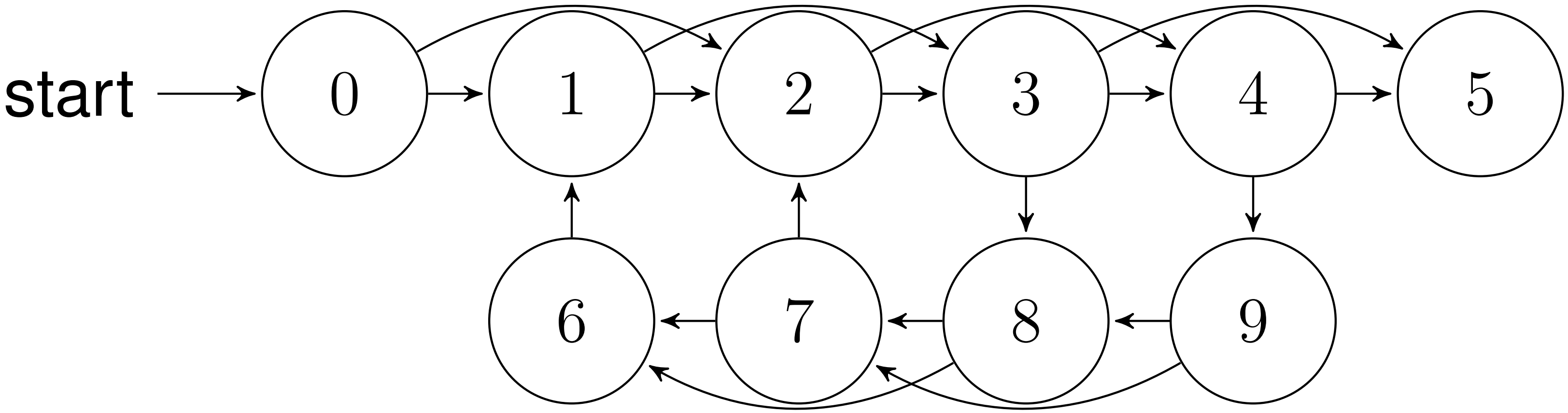
Types of Infinite Games

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.

Repetition 2



Boris has a memoryless winning strategy for the player.

Nodes	0	1	2	3	4	5	6	7	8	9
Boris' Moves	-	2	3	5	5	-	1	2	6	8
Half-moves remaining when Boris' turn	-	3	7	1	1	0	-	-	5	-
Half-moves remaining when Anke's turn	8	8	2	6	-	0	4	-	-	-

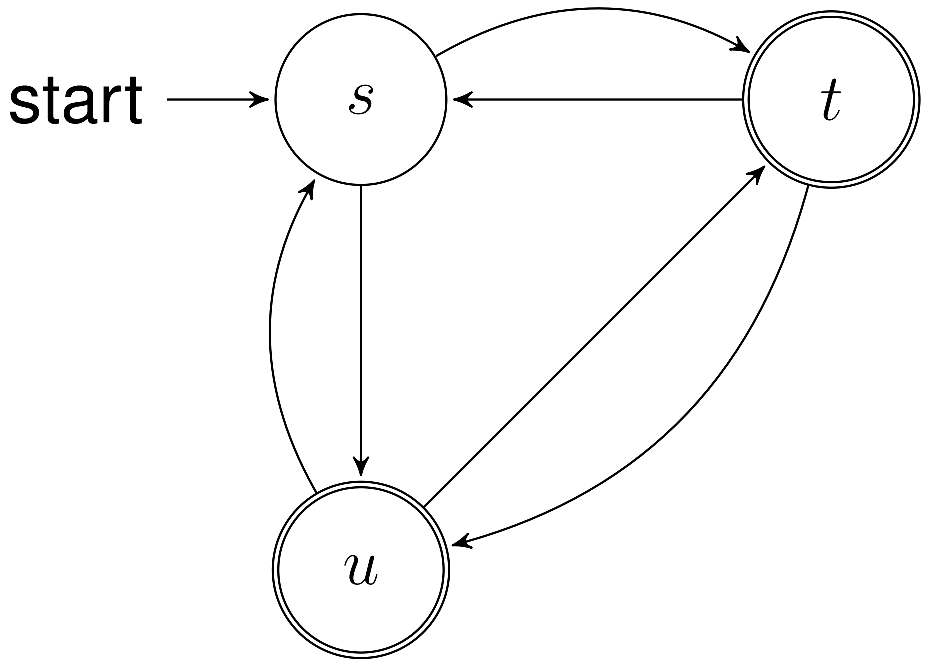
Repetition 3

There is an algorithm which can check which player wins a survival game (when playing optimally).

Let $\mathbf{f(v, Anke) = f(v, Boris) = 0}$ for nodes without successor.
Let $\mathbf{f(v, Anke) = n}$ for least $\mathbf{n}$ with $\mathbf{f(w, Boris) < n}$ for all successors $\mathbf{w}$ of $\mathbf{v}$.
Let $\mathbf{f(v, Boris) = n}$ for least $\mathbf{n}$ with $\mathbf{f(w, Anke) < n}$ for some successor $\mathbf{w}$ of $\mathbf{v}$.
Let $\mathbf{f(v, p) = \infty}$ whenever $\mathbf{\neg f(v, p) \leq 2 * |V|}$.

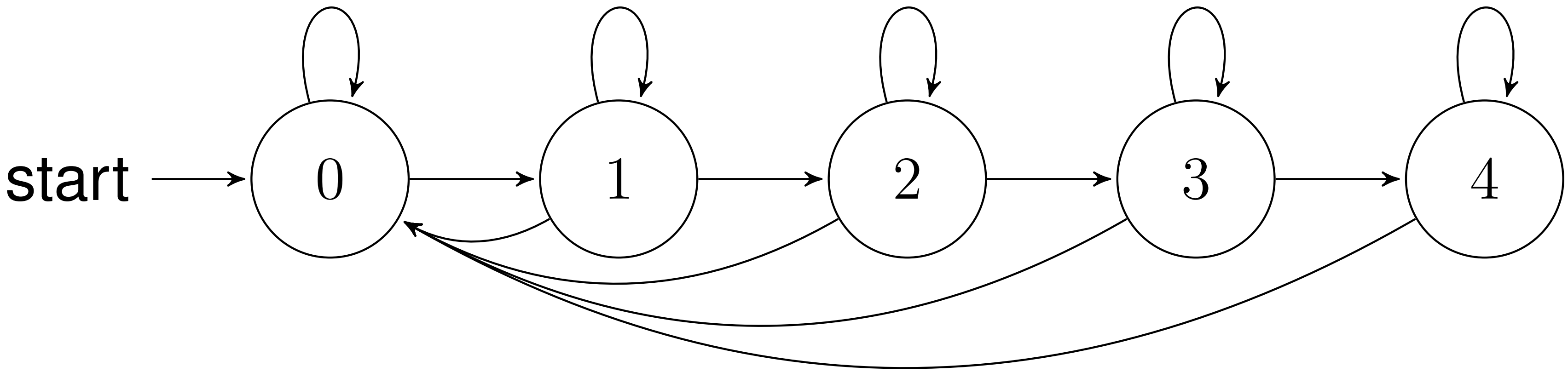
If $\mathbf{f(s, Anke) = \infty}$ then Anke has winning strategy by moving from each node $\mathbf{v}$ with $\mathbf{f(v, Anke) = \infty}$ to a successor $\mathbf{w}$ with $\mathbf{f(w, Boris) = \infty}$.
If $\mathbf{f(s, Anke) < \infty}$ then Boris has winning strategy by moving from each node $\mathbf{v}$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

Repetition 4



Anke has a winning strategy for this update game but no memoryless one.

Repetition 5



Node	0	1	2	3	4
Anke's Move	0	2	2	4	4

If $\mathbf{maxval}$ is even and one can always go from $\mathbf{n}$ to $\mathbf{n}$ and to $\mathbf{min\{maxval, n+1\}}$ and to $\mathbf{0}$ 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.

$\mathbf{b_0 b_1 b_2 \ldots \in \{0,1,2,3,4,5,6,7,8,9\}^\omega}$ can be used to represent the sum over all $\mathbf{b_k \cdot 10^{-k-1}}$ which permits to represent all real numbers between $\mathbf{0}$ and $\mathbf{1}$.

Wanted: Unique representation where only one of $\mathbf{0018999999999\ldots}$ and $\mathbf{00190000000\ldots}$ appears.

Goal: Make automaton which recognises those decimal strings which have some digit different from $\mathbf{9}$ infinitely often.

Büchi Automaton

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

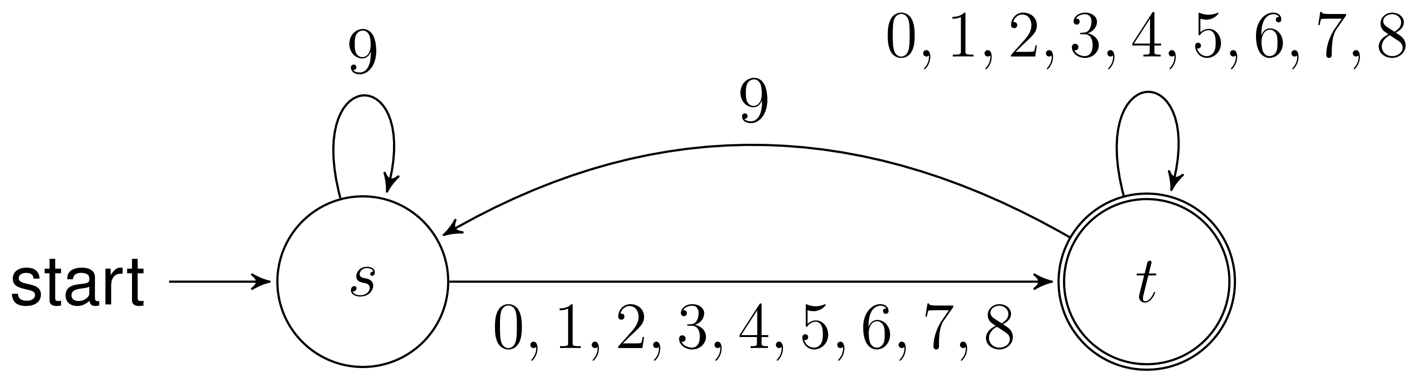
Given an infinite word $\mathbf{b_0 b_1 b_2} \ldots \in \mathbf{\Sigma}^\omega$, a run is a sequence $\mathbf{q_0 q_1 q_2} \ldots \in \mathbf{Q}^\omega$ of states such that $\mathbf{q_0} = \mathbf{s}$ and $(\mathbf{q_k}, \mathbf{b_k}, \mathbf{q_{k+1}}) \in \delta$ for all $\mathbf{k}$. Let

$$\mathbf{U} = \{\mathbf{p} \in \mathbf{Q} : \exists^\infty \mathbf{k}\, [\mathbf{q_k} = \mathbf{p}]\}$$

be the set of infinitely often visited states on this run. The run is accepting iff $\mathbf{U} \cap \mathbf{F} \neq \emptyset$. The Büchi automaton accepts an ω-word iff it has an accepting run on this ω-word, otherwise it rejects the ω-word.

Example of Automaton

The following deterministic Büchi automaton accepts all ω-words of reals between $\mathbf{0}$ and $\mathbf{1}$ which are not almost always $\mathbf{9}$.



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

This automaton goes infinitely often through the accepting state $\mathbf{t}$ iff there is infinitely often one of the digits $\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}, \mathbf{7}, \mathbf{8}$ and therefore the word is not of the form $\mathbf{w9}^\omega$.

Deterministic Büchi Automata

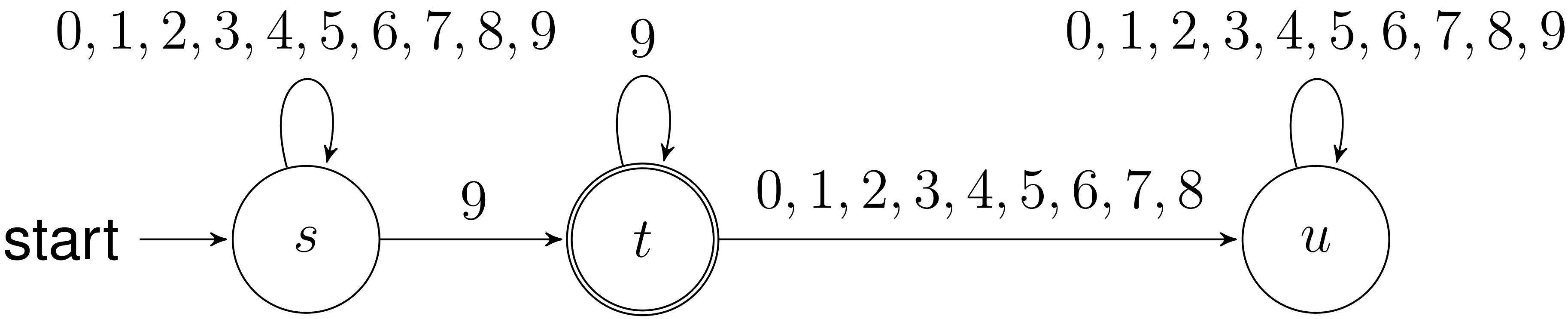
Exercise 6.2
Make a deterministic Büchi automaton which recognises the language $\mathbf{L}$ of all ω-words in $\{\mathbf{0},\mathbf{1},\mathbf{2}\}^\omega$ which contain all three digits infinitely often.

Exercise 6.3
Make a deterministic Büchi automaton which accepts an ω-word from $\{\mathbf{0},\mathbf{1},\mathbf{2}\}^\omega$ iff it contains at least two digits infinitely often.

Exercise 6.4
Make a deterministic Büchi automaton with three states which accepts all ω-words in which at least six of the usual ten digits occur infinitely often and which rejects all ω-words in which only one digit occurs infinitely often. There is no requirement what the automaton does on other ω-words.

Power of Non-Determinism

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



Limitation of Determinism

No deterministic Büchi automaton recognises the language $\mathbf{L}$ from the last slide.

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

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

Quiz

What problem arises at the usual product automaton construction for languages $\mathbf{L} \cup \mathbf{H}$ and $\mathbf{L} \cap \mathbf{H}$ of ω-words, given determistic Büchi automata recognising $\mathbf{L}$ and $\mathbf{H}$.

Let $\mathbf{U} = \{\mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_n}\}$ be a finite set of words. Is the set of all ω-words containing each of the above words infinitely often as a subword recognised by a deterministic Büchi automaton?

Product Automaton

Assume that $(\mathbf{Q_L}, \mathbf{\Sigma}, \delta_\mathbf{L}, \mathbf{s_L}, \mathbf{F_L})$ recognises $\mathbf{L}$ and $(\mathbf{Q_H}, \mathbf{\Sigma}, \delta_\mathbf{H}, \mathbf{s_H}, \mathbf{F_H})$ recognises $\mathbf{H}$.

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

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

The start state is $(\mathbf{s_L}, \mathbf{s_H}, \mathbf{11})$; note that the initial value of $\mathbf{r}$ is irrelevant.

Example of Intersection

$\mathbf{L}$ is the set of all ω-words containing infinitely many even digits.

$\mathbf{H}$ is the set of all ω-words containing infinitely often either $\mathbf{0}$ or $\mathbf{5}$.

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

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

Intersection $\mathbf{L \cap H}$ contains all ω-words where infinitely many even digits and also infinitely many digits from $\mathbf{0, 5}$ appear in the ω-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.

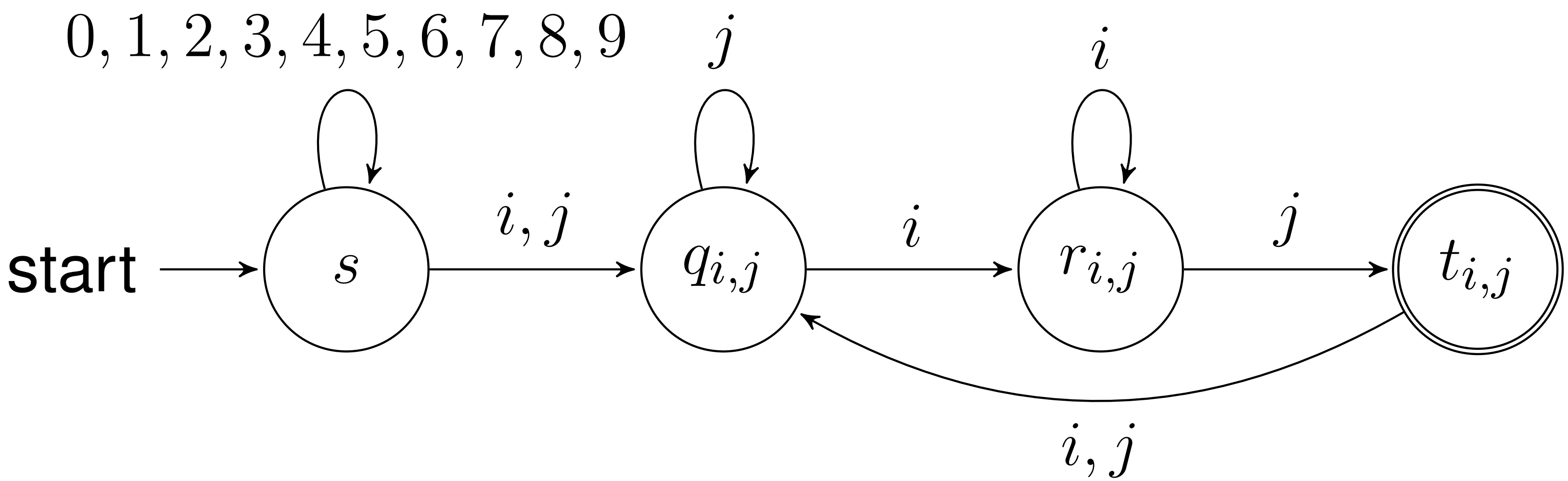
Let $r' = r$ for (s_L, s_H, r), $r' = 10$ for (t_L, s_H, r), $r' = 01$ for (s_L, t_H, r) and $r' = 11$ for (t_L, t_H, r).
$\delta((q_L, q_H, r), 0) = (t_L, t_H, r')$; $\delta((q_L, q_H, r), 5) = (s_L, t_H, r')$;
$\delta((q_L, q_H, r), a) = (t_L, s_H, r')$ for $a \in \{2, 4, 6, 8\}$;
$\delta((q_L, q_H, r), a) = (s_L, s_H, r')$ 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.

Exactly Two out of Ten Digits

Consider the following automaton $\mathbf{B_{i,j}}$ with $\mathbf{i \neq j}$.



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

Product Automata Exercises

Exercise 6.11: Construct deterministic Büchi automata for the language $\mathbf{L_{ab}}$ of all ω-word which do not contain the subword $\mathbf{ab}$ anywhere. Then construct the intersection automaton for $\mathbf{L_{01}} \cap \mathbf{L_{23}}$; the alphabet is $\{\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}\}$.

Exercise 6.12: Construct a deterministic Büchi automaton for the language $\mathbf{H_{ab}}$ of all ω-words which in which the subword $\mathbf{ab}$ occurs infinitely often. Then construct the intersection via a product automaton for $\mathbf{H_{01}} \cap \mathbf{H_{23}}$; the alphabet is $\{\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}\}$.

Exercise 6.13: Construct a deterministic Büchi automaton with four states for $\mathbf{H_{01}} \cup \mathbf{H_{23}}$ with $\mathbf{H_{ab}}$ as in Exercise 6.12. This automaton does not need to be of the form of a product automaton. The alphabet is $\{\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}\}$.

Characterising Languages

Theorem 6.14 [Büchi 1960]
The following are equvivalent for a language $\mathbf{L}$ of ω-words:
(a) $\mathbf{L}$ is recognised by a non-deterministic Büchi automaton;
(b) $\mathbf{L} = \cup_{\mathbf{m} \in \{1,\ldots,\mathbf{n}\}} \mathbf{A_m} \mathbf{B_m^\omega}$ for some $\mathbf{n}$ and $\mathbf{2n}$ regular languages $\mathbf{A_1}, \mathbf{B_1}, \ldots, \mathbf{A_n}, \mathbf{B_n}$.

Here $\mathbf{B_m^\omega}$ is the concatenation of infinitely many non-empty strings from $\mathbf{B_m}$.

Direction One

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

If an ω-word is in $\mathbf{L}$ then the Büchi automaton has a run on it which goes infinitely often through one $\mathbf{p_m}$.

Direction Two

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

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

Furthermore, let $\delta_\mathbf{0}$ be $\delta_\mathbf{1} \cup \delta_\mathbf{2} \cup \ldots \cup \delta_\mathbf{2n}$ plus the following transitions for each $\mathbf{a} \in \Sigma$: first, $(\mathbf{s_0}, \mathbf{a}, \mathbf{q})$ if there is an $\mathbf{m}$ such that $(\mathbf{s_{2m-1}}, \mathbf{a}, \mathbf{q}) \in \delta_{\mathbf{2m-1}}$; second, $(\mathbf{s_0}, \mathbf{a}, \mathbf{q})$ if there is an $\mathbf{m}$ such that $\varepsilon \in \mathbf{A_m}$ and $(\mathbf{s_{2m}}, \mathbf{a}, \mathbf{q}) \in \delta_{\mathbf{2m}}$; third, $(\mathbf{s_0}, \mathbf{a}, \mathbf{s_{2m}})$ if $\mathbf{a} \in \mathbf{A_m}$; fourth, $(\mathbf{q}, \mathbf{a}, \mathbf{s_{2m}})$, if there are $\mathbf{m}$ and $\mathbf{p} \in \mathbf{F_{2m-1}}$ with $\mathbf{q} \in \mathbf{N_{2m-1}}$ and $(\mathbf{q}, \mathbf{a}, \mathbf{p}) \in \delta_{\mathbf{2m-1}}$; fifth, $(\mathbf{q}, \mathbf{a}, \mathbf{s_{2m}})$, if there are $\mathbf{m}$ and $\mathbf{p} \in \mathbf{F_{2m}}$ with $\mathbf{q} \in \mathbf{N_{2m}}$ and $(\mathbf{q}, \mathbf{a}, \mathbf{p}) \in \delta_{\mathbf{2m}}$.

Continuation

Now $\{\mathbf{s_2}, \mathbf{s_4}, \mathbf{s_6}, \ldots, \mathbf{s_{2n}}\}$ is the set $\mathbf{F_0}$ of the final states of the Büchi automaton $(\mathbf{N_0}, \mathbf{\Sigma}, \delta_\mathbf{0}, \mathbf{s_0}, \mathbf{F_0})$. That is, this Büchi automaton accepts an ω-word iff there is a run of the automaton which goes infinitely often through a node of the form $\mathbf{s_{2m}}$; by the way how $\delta_\mathbf{0}$ is defined, this is equivalent to saying that the given ω-word is in $\mathbf{A_m} \cdot \mathbf{B_m^\omega}$.

6.15: Muller automata

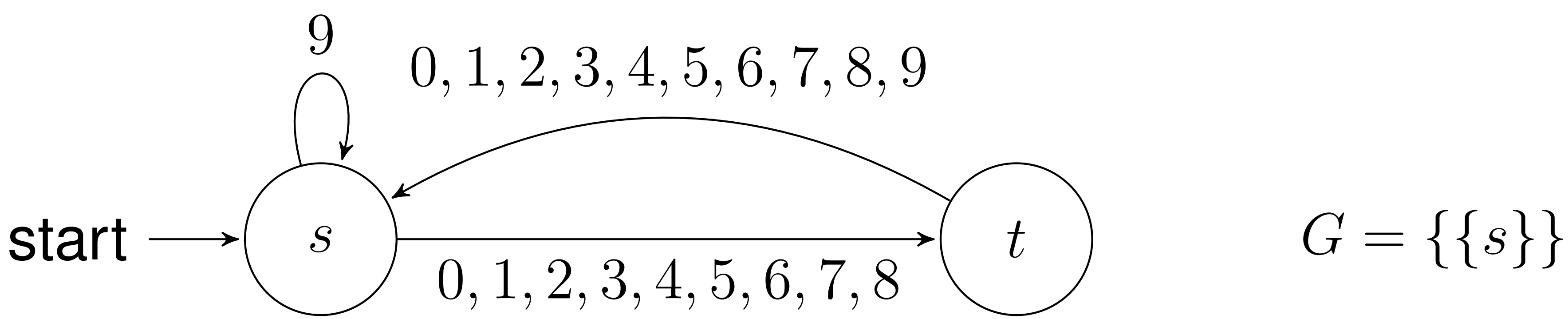
A Muller automaton $(\mathbf{Q}, \mathbf{\Sigma}, \delta, \mathbf{s}, \mathbf{G})$ consists of a set of states $\mathbf{Q}$, an alphabet $\mathbf{\Sigma}$, a transition relation δ, a starting state $\mathbf{s}$ and a set $\mathbf{G}$ of subsets of $\mathbf{Q}$. A run of the Muller automaton on an ω-word $\mathbf{b_0 b_1 b_2} \ldots \in \mathbf{\Sigma}^\omega$ is a sequence $\mathbf{q_0 q_1 q_2} \ldots$ with $\mathbf{q_0} = \mathbf{s}$ and $(\mathbf{q_k}, \mathbf{b_k}, \mathbf{q_{k+1}}) \in \delta$ for all $\mathbf{k}$. A run of the Muller automaton is accepting iff the set $\mathbf{U}$ of infinitely often visited states satisfies $\mathbf{U} \in \mathbf{G}$. The Muller automaton accepts the ω-word $\mathbf{b_0 b_1 b_2} \ldots$ iff it has an accepting run on it.

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

Example

The language of all ω-words of the form $\mathbf{w9}^\omega$ is recognised by the deterministic Muller automaton $(\{\mathbf{s},\mathbf{t}\},\{\mathbf{0},\mathbf{1},\mathbf{2},\mathbf{3},\mathbf{4},\mathbf{5},\mathbf{6},\mathbf{7},\mathbf{8},\mathbf{9}\},\delta,\mathbf{s},\{\{\mathbf{s}\}\})$, where $\delta(\mathbf{s},\mathbf{a})=\mathbf{t}$ for $\mathbf{a}<\mathbf{9}$, $\delta(\mathbf{s},\mathbf{9})=\mathbf{s}$ and $\delta(\mathbf{t},\mathbf{a})=\mathbf{s}$ for all $\mathbf{a}$.

The following diagramme illustrates the Muller automaton:



Deterministic Muller Automata

Exercise 6.16
Make a deterministic Muller automaton recognising the complement of the language $\mathbf{L}$ from Exercise 6.2. That is, the Muller automaton should accept all ω-words in which one or two of the digits $\mathbf{0}, \mathbf{1}, \mathbf{2}$ occur only finitely often.

Exercise 6.17
Make a deterministic Muller automaton with alphabet $\{\mathbf{0}, \mathbf{1}, \mathbf{2}\}$ which recognises the language of all ω-words for which there is an even number of $\mathbf{ab} \in \{\mathbf{01}, \mathbf{12}, \mathbf{20}\}$ which occurs infinitely often as a subword in the ω-word.

Exercise 6.18
Make a deterministic Muller automaton with alphabet $\{\mathbf{0}, \mathbf{1}, \mathbf{2}\}$ which recognises the language of all ω-words for which there are no $\mathbf{ab} \in \{\mathbf{21}, \mathbf{10}, \mathbf{02}\}$ which occur infinitely often as a subword in the ω-word.

Characterisations

Theorem 6.19 [McNaughton 1966, Safra 1988]
The following conditions are equivalent for an ω-language $\mathbb{L}$.
(a) $\mathbb{L}$ is recognised by a non-deterministic Büchi automaton;
(b) $\mathbb{L}$ is recognised by a deterministic Muller automaton;
(c) $\mathbb{L}$ is recognised by a non-deterministic Muller automaton.

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

Exponential Blow-Up

Exercise 6.21

Consider $\mathbf{L_h} = \{\mathbf{b_0 b_1 b_2} \ldots \exists^\infty \mathbf{m}\,[\mathbf{b_m} = \mathbf{b_{m+h}}]\}$ over alphabet $\{\mathbf{0}, \mathbf{1}, \mathbf{2}\}$. Make a non-deterministic Büchi automaton and a deterministic Muller automaton to recognise $\mathbf{L_h}$ and a non-deterministic Büchi automaton to recognise the complement of $\mathbf{L_h}$. How many states do these automata have?

Infinitely Often Occurring Symbols

Exercise 6.22
Let $\mathbf{h} = |\mathbf{\Sigma}|$ and let $\mathbf{L}$ be any language where for each ω-word α the membership of α in $\mathbf{L}$ only depends on the set of symbols which appears infinitely often in α. Show that there is a deterministic Muller automaton with $\mathbf{h}$ states recognising $\mathbf{L}$.

Exercise 6.23
Let $\mathbf{L}$ contain all ω-words $\alpha \in \mathbf{\Sigma}^\omega$ for which at least half of the symbols in $\mathbf{\Sigma}$ occurs infinitely often.
(a) Make a deterministic Büchi automaton recognising $\mathbf{L}$ with up to $\mathbf{2}^{|\mathbf{\Sigma}|-\mathbf{1}}$ states.
(b) Make a non-deterministic Büchi automaton recognising $\mathbf{L}$ with up to $|\mathbf{\Sigma}|^\mathbf{2}/\mathbf{2}+\mathbf{2}$ states.

Other Automata

Rabin and Streett automata are automata of the form $(\mathbf{Q}, \mathbf{\Sigma}, \delta, \mathbf{s}, \mathbf{\Omega})$ where $\mathbf{\Omega}$ is a set of pairs $(\mathbf{E}, \mathbf{F})$ of subsets of $\mathbf{Q}$ and a run on an ω-word $\mathbf{b_0 b_1 b_2} \ldots$ is a sequence $\mathbf{q_0 q_1 q_2} \ldots$ with $\mathbf{q_0} = \mathbf{s}$ and $(\mathbf{q_n}, \mathbf{b_n}, \mathbf{q_{n+1}}) \in \delta$ for all $\mathbf{n}$. For a Rabin automaton, a run is accepting iff the set $\mathbf{U} = \{\mathbf{p} \in \mathbf{Q} : \exists^\infty \mathbf{n}\, [\mathbf{p} = \mathbf{q_n}]\}$ of infinitely often visited nodes satisfies $\mathbf{U} \cap \mathbf{E} \neq \emptyset$ and $\mathbf{U} \cap \mathbf{F} = \emptyset$ for a pair $(\mathbf{E}, \mathbf{F}) \in \mathbf{\Omega}$; for a Streett automaton a run is accepting iff $\mathbf{U}$ satisfies $(\mathbf{U} \cap \mathbf{E} \neq \emptyset$ or $\mathbf{U} \cap \mathbf{F} = \emptyset)$ for all $(\mathbf{E}, \mathbf{F}) \in \mathbf{\Omega}$.

If an ω-language $\mathbf{L}$ is recognised by a complete deterministic Rabin automaton $(\mathbf{Q}, \mathbf{\Sigma}, \delta, \mathbf{s}, \mathbf{\Omega})$ then its complement is recognised by the deterministic Streett automaton $(\mathbf{Q}, \mathbf{\Sigma}, \delta, \mathbf{s}, \{(\mathbf{F}, \mathbf{E}) : (\mathbf{E}, \mathbf{F}) \in \mathbf{\Omega}\})$.

Example 6.25

Assume that an automaton with states $\mathbf{Q} = \{\mathbf{q_0}, \mathbf{q_1}, \ldots, \mathbf{q_9}\}$ on seeing digit $\mathbf{d}$ goes into state $\mathbf{q_d}$. Then the condition $\mathbf{\Omega}$ consisting of all pairs $(\mathbf{Q} - \{\mathbf{q_d}\}, \{\mathbf{q_d}\})$ produces an Rabin automaton which accepts iff some digit $\mathbf{d}$ appears only finitely often in a given ω-word.

Assume that an automaton with states $\mathbf{Q} = \{\mathbf{s}, \mathbf{q_0}, \mathbf{q_1}, \ldots, \mathbf{q_9}\}$ can go on any digit from $\mathbf{s}$ to any state; furthermore, in state $\mathbf{q_d}$, it stays on $\mathbf{d}$ in this state and returns on all other digits to $\mathbf{s}$. Let $\mathbf{E} = \{\mathbf{q_0}, \mathbf{q_1}, \ldots, \mathbf{q_9}\}$ and $\mathbf{F} = \{\mathbf{s}\}$ and $\mathbf{\Omega} = \{(\mathbf{E}, \mathbf{F})\}$. This Rabin automaton accepts all ω-words where exactly one digit occurs infinitely often. The corresponding Streett automaton uses $\mathbf{E} = \emptyset$ and $\mathbf{F} = \{\mathbf{s}\}$.

Exercises

Quiz

Give an algorithm to translate a Büchi automaton into a Streett automaton.

Exercise 6.27

Assume that for an ω-language $\mathbf{L_k}$ there is a Street automaton $(\mathbf{Q_k}, \mathbf{\Sigma}, \mathbf{s_k}, \delta_\mathbf{k}, \mathbf{\Omega_k})$. Prove that then there is a Streett automaton for $\mathbf{L_1} \cap \mathbf{L_2}$ with states $\mathbf{Q_1} \times \mathbf{Q_2}$, start state $(\mathbf{s_1}, \mathbf{s_2})$, transition relation $\delta_\mathbf{1} \times \delta_\mathbf{2}$ and an $\mathbf{\Omega}$ containing $|\mathbf{\Omega_1}| + |\mathbf{\Omega_2}|$ pairs. Explain how $\mathbf{\Omega}$ is constructed.

Alternating Büchi Automata

Definition. Anke and Boris decide on moves in afa $(\mathbf{Q}, \mathbf{\Sigma}, \delta, \mathbf{s}, \mathbf{F})$ while processing an ω-word $\mathbf{w}$. Three possibilities for pairs $(\mathbf{q}, \mathbf{a})$ of states $\mathbf{q}$ and symbols $\mathbf{a}$:

- $(\mathbf{q}, \mathbf{a}) \to \mathbf{r}$: Next state is $\mathbf{r}$;
- $(\mathbf{q}, \mathbf{a}) \to \mathbf{r} \vee \mathbf{p}$: Anke picks $\mathbf{r}$ or $\mathbf{p}$;
- $(\mathbf{q}, \mathbf{a}) \to \mathbf{r} \wedge \mathbf{p}$: Boris picks $\mathbf{r}$ or $\mathbf{p}$.

The afa accpets an ω-word $\mathbf{w}$ iff Anke has a winning strategy to ensure that the game always goes infinitely often through states from $\mathbf{F}$.

Example. $\mathbf{Q} = \{\mathbf{p}, \mathbf{q}, \mathbf{r}\}$; $\mathbf{\Sigma} = \{\mathbf{0}, \mathbf{1}\}$; language $(\{\mathbf{0}\}^* \cdot \{\mathbf{1}\})^\omega$.

state	type	$\mathbf{0}$	$\mathbf{1}$
$\mathbf{p}$	start, rejecting	$\mathbf{p} \wedge \mathbf{q} \wedge \mathbf{r}$	$\mathbf{q} \vee \mathbf{r}$
$\mathbf{q}$	accepting	$\mathbf{p} \wedge \mathbf{q} \wedge \mathbf{r}$	$\mathbf{p} \vee \mathbf{r}$
$\mathbf{r}$	accepting	$\mathbf{p} \wedge \mathbf{q} \wedge \mathbf{r}$	$\mathbf{p} \vee \mathbf{q}$

Büchi Automata and Games

Example 6.30: Assume a Büchi Game $(\mathbf{G}, \mathbf{E}, \mathbf{s}, \mathbf{W})$ is given. Now one can construct a Büchi AFA as follows: The set $\mathbf{G}$ is the state set, the alphabet is $\{\mathbf{Anke}, \mathbf{Boris}\}$. Assume that for a node $\mathbf{q}$ the outgoing edges to nodes $\mathbf{p_1}, \mathbf{p_2}, \ldots, \mathbf{p_k}$. Then the AFA has the following transitions:
$\delta(\mathbf{q}, \mathbf{Anke}) = \mathbf{p_1} \vee \mathbf{p_2} \vee \ldots \vee \mathbf{p_k}$;
$\delta(\mathbf{q}, \mathbf{Boris}) = \mathbf{p_1} \wedge \mathbf{p_2} \wedge \ldots \wedge \mathbf{p_k}$.
So the input tells which player selects the next move. The accepting states of the Büchi AFA are those in $\mathbf{W}$.

Now Anke has a winning strategy for the game $(\mathbf{G}, \mathbf{E}, \mathbf{s}, \mathbf{W})$ iff the Büchi AFA accepts $(\mathbf{Anke}\,\mathbf{Boris})^\omega$.

Exercise 6.31: Given a Büchi game $(\mathbf{G}, \mathbf{E}, \mathbf{s}, \mathbf{W})$, construct a deterministic Büchi automaton which reads plays (sequences of nodes visited by alternating moves of Anke and Boris) and which accepts iff all moves in the play are possible and Anke wins the play.

Exercises 6.32-6.33

Exercise 6.32: Given a Büchi game (G, E, s, W), a nondeterministic Büchi automaton is using the states $G \cup \{0\}$ with $W \cup \{0\}$ being accepting and the input is every second node of a play, so if the play is $s\, q_1\, q_2\, q_3\, q_4\, \ldots$ then Anke does the moves to $q_1, q_3, q_5, \ldots$ and Boris to $q_2, q_4, q_6, \ldots$; now the game is that Anke reads q_{2k} with $q_0 = s$ and if one cannot go from the current state q_{2k-1} to q_{2k} then Anke moves to 0 else Anke moves to a successor node of q_{2k} which is called q_{2k+1}. From 0, one can only move to 0 (independently of the input).

Show that Anke has a winning strategy for the Büchi game iff the so constructed nondeterministic finite automaton accepts all halfplays as described here.

Exercise 6.33: Construct the above Büchi automaton for the Büchi game with $G = \{1, 2, 3, 4\}$, $s = 1$, $W = \{2, 3\}$ and $E = \{(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 1), (3, 3), (3, 1), (4, 4)\}$.

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? (Context-sensitive languages are closed under complement, but the proof is not needed for the midterm.) 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. Learn the pumping lemmas.
Revise finite and infinite games on finite graphs and the automata for languages of ω-words.
Do the Selftests in the Lecture Notes at the end of Chapters 3 and 6.

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

Electronic Exam Instructions

You will find in the Luminus-Folder Mock Exam an old exam zip-file. Please load this and try to type solutions with pdf-commenting by text. Do not use sticky notes, as those sometimes get lost. Save the modified pdf-file and check whether your writing is still there. Store the file in the mock-submission folder of Luminus for getting it checked.

For midterm, prefix the file name with your student number. If student number is “A01234567X” the saved file should be “A01234567X-typed.pdf” (in the case that typed with a pdf-viewer) or “A01234567X-handwritten.pdf” (in the case that it is handwritten and scanned). Both options are fine.

Do the SoC Mock Exams and train the typing and scanning before the midterm exam. Exams use “encrypted questions” and “Luminus Files”. https://mysoc.nus.edu.sg/academic/e-exam-sop-for-students/