Advanced Automata Theory 4
Games

Frank Stephan

Department of Computer Science

Department of Mathematics

National University of Singapore

fstephan@comp.nus.edu.sg

Repetition 1

Given two dfas recognising $\mathbf{L}$ and $\mathbf{H}$, one can form new dfas recognising $\mathbf{L} \cap \mathbf{H}$, $\mathbf{L} \cup \mathbf{H}$, $\mathbf{L} - \mathbf{H}$, $(\mathbf{L} - \mathbf{H}) \cup (\mathbf{H} - \mathbf{L})$, $\mathbf{L}^*$ and $\mathbf{L} \cdot \mathbf{H}$. Furthermore, there are dfas recognising any finite set.

In various cases (Kleene star, concatenation), one first constructs an nfa recognising the correspdonding language and then transforms it into a dfa recognising the same language using Büchi's power set construction.

Hence, one can prove by induction over the size of regular expressions, that every language defined by a regular expression is also recognised by a dfa.

Repetition 2

If $\mathbf{L}$ and $\mathbf{H}$ are context-free, so are $\mathbf{L} \cdot \mathbf{H}$, $\mathbf{L} \cup \mathbf{H}$ and $\mathbf{L}^*$.

There are context-free languages $\mathbf{L}$ and $\mathbf{H}$ such that $\mathbf{L} - \mathbf{H}$ and $\mathbf{L} \cap \mathbf{H}$ are both not context-free.

Example: $\mathbf{L} = \{\mathbf{0^n 1^n 2^m : n, m} \in \mathbb{N}\}$ and $\mathbf{H} = \{\mathbf{0^n 1^m 2^k : n, m, k} \in \mathbb{N} \wedge \mathbf{m} \leq \mathbf{k}\}$.

$\mathbf{L} \cap \mathbf{H} = \{\mathbf{0^n 1^n 2^m : n} \leq \mathbf{m}\}$ and $\mathbf{L} - \mathbf{H} = \{\mathbf{0^n 1^n 2^m : n} > \mathbf{m}\}$. Both are not context-free by the Pumping Lemma.

$\mathbf{L} \cap \mathbf{H}$: Take $\mathbf{n}$ large enough and pump $\mathbf{0^n 1^n 2^n}$. At most two types of digits are pumped. If the pumped parts contain $\mathbf{2}$ then omitting the pumped parts produces a word outside $\mathbf{L} \cap \mathbf{H}$; if the pumped parts do not contain a $\mathbf{2}$, then inserting the pumped parts twice makes the number of $\mathbf{0}$ and $\mathbf{1}$ to be larger than the number of $\mathbf{2}$.

$\mathbf{L} - \mathbf{H}$: Take $\mathbf{n}$ large enough and pump $\mathbf{0^n 1^n 2^{n-1}}$.

Repetition 3

Theorem.
If $\mathbf{L}$ is context-free and $\mathbf{H}$ is regular then $\mathbf{L} \cap \mathbf{H}$ is context-free.

Construction.
Let $(\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}, \mathbf{S})$ be a context-free grammar generating $\mathbf{L}$ with every rule being either $\mathbf{A} \rightarrow \mathbf{w}$ or $\mathbf{A} \rightarrow \mathbf{BC}$ with $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbf{N}$ and $\mathbf{w} \in \mathbf{\Sigma}^*$.

Let $(\mathbf{Q}, \mathbf{\Sigma}, \delta, \mathbf{s}, \mathbf{F})$ be a dfa recognising $\mathbf{H}$.

Let $\mathbf{S}' \notin \mathbf{Q} \times \mathbf{N} \times \mathbf{Q}$ and make the following new grammar $(\mathbf{Q} \times \mathbf{N} \times \mathbf{Q} \cup \{\mathbf{S}'\}, \mathbf{\Sigma}, \mathbf{R}, \mathbf{S}')$ with rules $\mathbf{R}$:
$\mathbf{S}' \rightarrow (\mathbf{s}, \mathbf{S}, \mathbf{q})$ for all $\mathbf{q} \in \mathbf{F}$;
$(\mathbf{p}, \mathbf{A}, \mathbf{q}) \rightarrow (\mathbf{p}, \mathbf{B}, \mathbf{r})(\mathbf{r}, \mathbf{C}, \mathbf{q})$ for all rules $\mathbf{A} \rightarrow \mathbf{BC}$ in $\mathbf{P}$ and all $\mathbf{p}, \mathbf{q}, \mathbf{r} \in \mathbf{Q}$;
$(\mathbf{p}, \mathbf{A}, \mathbf{q}) \rightarrow \mathbf{w}$ for all rules $\mathbf{A} \rightarrow \mathbf{w}$ in $\mathbf{P}$ with $\delta(\mathbf{p}, \mathbf{w}) = \mathbf{q}$.

Repetition 4

Context-sensitive languages are closed under union, concatenation, Kleene star and intersection.

Construction for intersection most complicated (among these).

Used method: Overlayed non-terminal characters with upper and lower half. Word in upper half follows derivation of first language, word in lower half follows derivation of second language. Need to use spaces and have rules for space management.

At the end, there is a word $\mathbf{v} \in \Sigma^*$ coded into the upper half and a word $\mathbf{w} \in \Sigma^*$ coded into the lower half. Can be terminalised only if $\mathbf{v} = \mathbf{w}$.

Closure under complement delayed to Lecture 7.

Games

Here games are two-player games.
Anke versus Boris.
Anke starts to play and then Boris and Anke move alternately.

Game in Graph.
“Board of the game” is a finite graph $(\mathbf{G}, \mathbf{E})$.
Players move a marker around in the graph.
The player who moves the marker into the target wins.

Although many games are not defined that way, they can be represented as a game moving a marker on a graph.

Digit Game

Make a number to $\mathbf{0}$.

Starting with a decimal number, say $\mathbf{257}$.

Each player replaces one digit by a smaller one.

The player who reaches $\mathbf{0}$ wins.

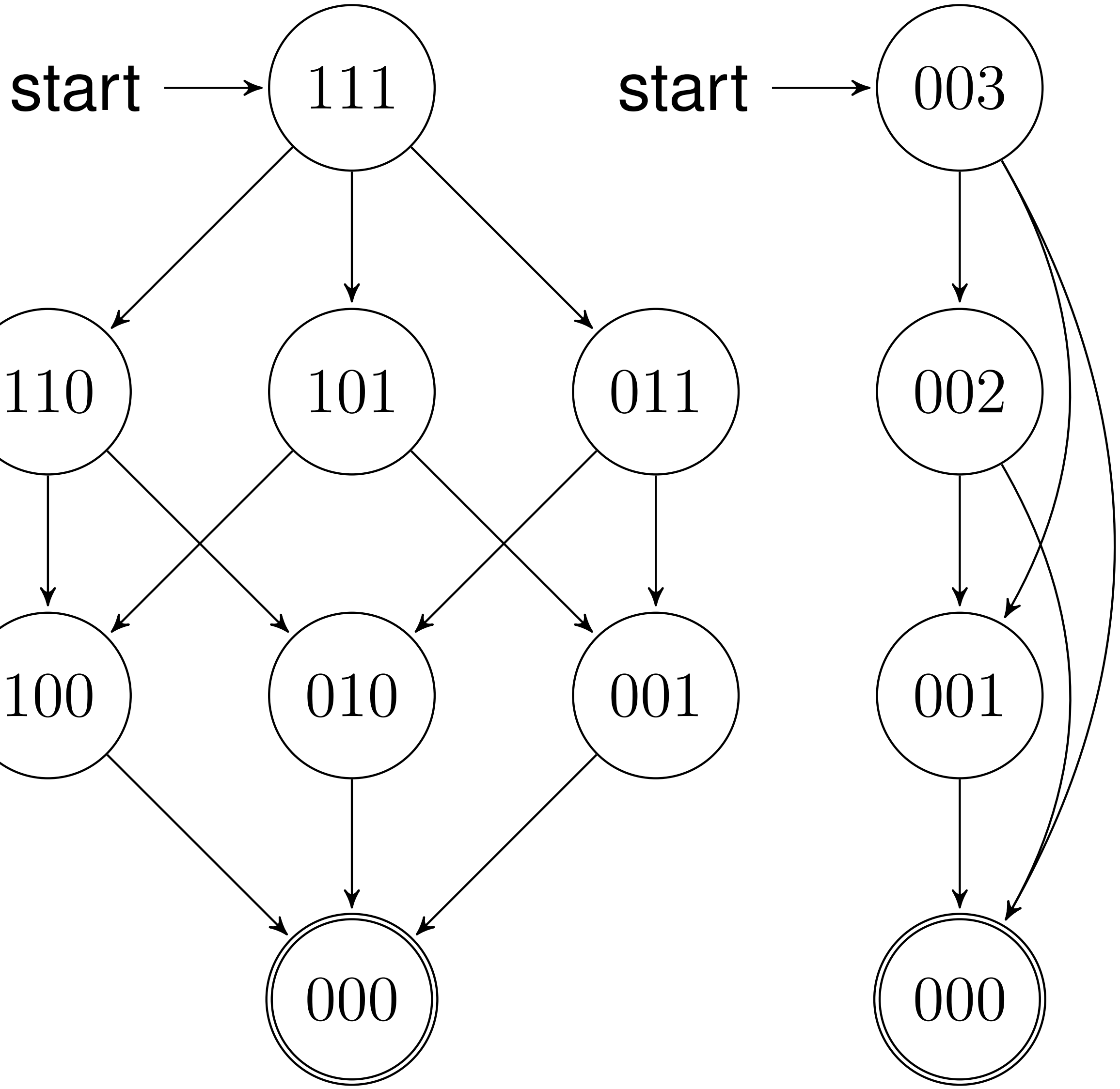
Sample Plays

Anke: $\mathbf{257} \rightarrow \mathbf{252}$
Boris: $\mathbf{252} \rightarrow \mathbf{222}$
Anke: $\mathbf{222} \rightarrow \mathbf{221}$
Boris: $\mathbf{221} \rightarrow \mathbf{211}$
Anke: $\mathbf{211} \rightarrow \mathbf{011}$
Boris: $\mathbf{011} \rightarrow \mathbf{001}$
Anke: $\mathbf{001} \rightarrow \mathbf{000}$.

Anke: $\mathbf{257} \rightarrow \mathbf{157}$.
Boris: $\mathbf{157} \rightarrow \mathbf{154}$.
Anke: $\mathbf{154} \rightarrow \mathbf{114}$.
Boris: $\mathbf{114} \rightarrow \mathbf{110}$.
Anke: $\mathbf{110} \rightarrow \mathbf{100}$.
Boris: $\mathbf{100} \rightarrow \mathbf{000}$.

Digit Game as Graph

Game graph when starting at two sample positions.



Winning Positions and Strategies

A winning strategy is an algorithm or table which tells Anke in each position how to move (in dependence of the prior moves which occurred in the game) such that Anke will eventually win.

A node $\mathbf{v}$ is a winning position for Anke iff there is a winning strategy which tells Anke how to win, provided that the game starts from the node $\mathbf{v}$.

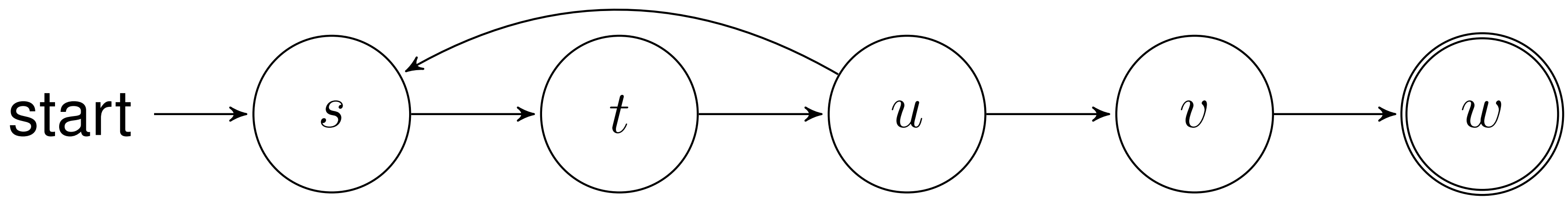
Similarly one defines winning strategies and positions for Boris.

Example 4.3: Assume Anke starts in the following nodes. Then $\mathbf{001}, \mathbf{012}, \mathbf{111}$ are winning positions and $\mathbf{011}, \mathbf{213}, \mathbf{257}$ are losing positions.

Quiz: Assume that it is Anke's turn in the following positions. Which are for Anke winning and which are losing positions: $\mathbf{125}, \mathbf{323}, \mathbf{246}, \mathbf{555}$?

Example 4.5

Consider the following game:



Assume that it is Anke's move. If she is in $\mathbf{v}$ then she can win, so $\mathbf{v}$ is a winning position.

What about $\mathbf{u}$? If she moves from $\mathbf{u}$ to $\mathbf{v}$, she loses and Boris wins. So every player reaching $\mathbf{u}$ returns to $\mathbf{s}$.

Positions which are neither winning nor losing positions are called draw positions. Here $\mathbf{s}, \mathbf{t}, \mathbf{u}$ are draw positions and when both players play optimally, then the game runs forever.

Certain board games have specific rules like that a draw is reached if a position is visited three times in order to abort infinite sequences of moves.

Theorem 4.6: Deciding Games

Theorem. There is an algorithm which determines which player has a winning strategy. The algorithm runs in time polynomial in the size of the graph.

Proof. Let $\mathbf{Q}$ be the set of all nodes and $\mathbf{T}$ be the set of target nodes. The games starts in some node in $\mathbf{Q} - \mathbf{T}$.

1. Let $\mathbf{T_0 = T}$ and $\mathbf{S_0} = \emptyset$ and $\mathbf{n = 0}$.
2. Let $\mathbf{S_{n+1} = S_n \cup \{q \in Q - (T_n \cup S_n)}$: one can go in one step from $\mathbf{q}$ to a node in $\mathbf{T_n\}}$.
3. Let $\mathbf{T_{n+1} = T_n \cup \{q \in Q - (T_n \cup S_{n+1})}$: if one goes from $\mathbf{q}$ one step then one ends up in $\mathbf{S_{n+1}\}}$.
4. If $\mathbf{S_{n+1} \neq S_n}$ or $\mathbf{T_{n+1} \neq T_n}$ then let $\mathbf{n = n + 1}$ and goto 2.
5. $\mathbf{S_n}$ are winning positions, $\mathbf{T_n - T}$ are losing positions and $\mathbf{Q - (T_n \cup S_n)}$ are draw positions.

Comments

“Winning position” and “Losing position” refers to the player whose turn is to move. Assume that it is Anke’s turn to move.

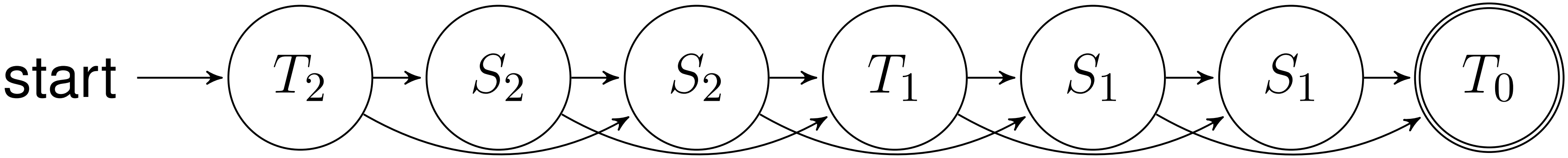
$\mathbf{S_n}$ is the set of positions such that Anke can win within $\mathbf{n}$ rounds when she starts to play now. $\mathbf{T_n}$ are those positions $\mathbf{v}$ where Anke loses within $\mathbf{n}$ rounds when she starts to move now or where the game is already in the target (so that the last player Boris moving it there has won).

$\mathbf{S_{n+1}}$ is the set of nodes from which Anke can move into a node in $\mathbf{T_n}$; $\mathbf{T_{n+1}}$ is the set of nodes where either Anke cannot move or the game is terminated or any move ends up in a node in $\mathbf{S_{n+1}}$ so that the opposing player wins within $\mathbf{n+1}$ moves.

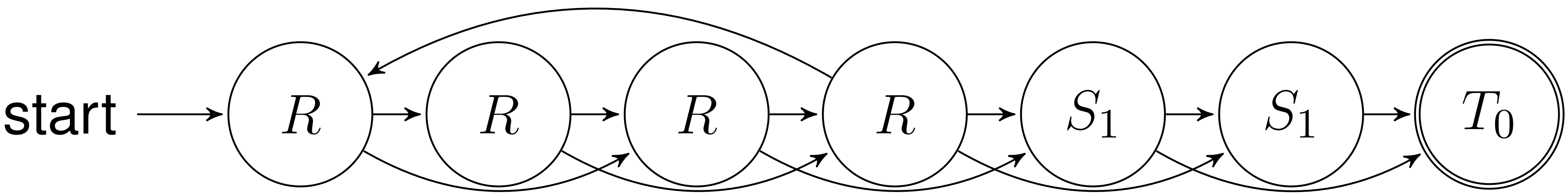
If a player is in a draw position, then the player can move such that the game remains in a draw position.

Example for Theorem 4.6

Winning and Losing Positions for Easy Game.



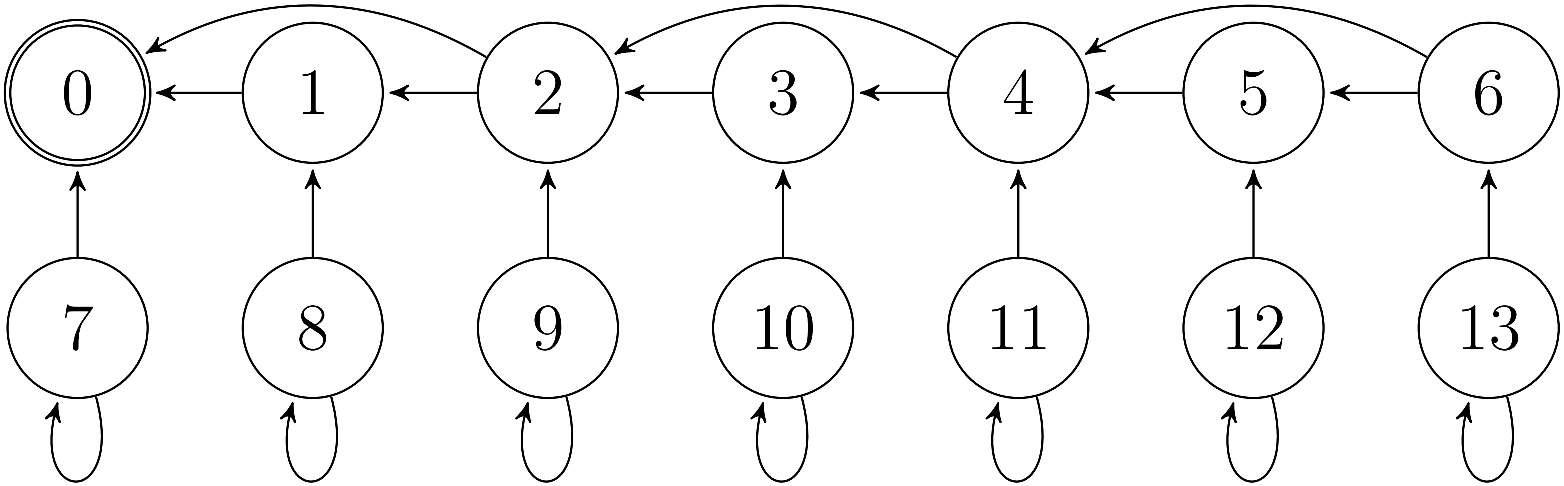
So the above game is a losing game for Anke and a winning game for Boris.



Here the players will always move inside the set $\mathbf{R}$ of nodes and not move to the nodes of $\mathbf{S_1}$ as then the opponent wins.

Exercise 4.7

Consider a graph with node-set $\mathbf{Q = \{0, 1, 2, \ldots, 13\}}$*, target* $\mathbf{T = \{0\}}$ *and the following edges between the nodes.*



Determine which of the nodes $\mathbf{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13}$ *are winning-positions, losing-positions and draw-positions for player Anke.*

Example 4.8: Tic Tac Toe

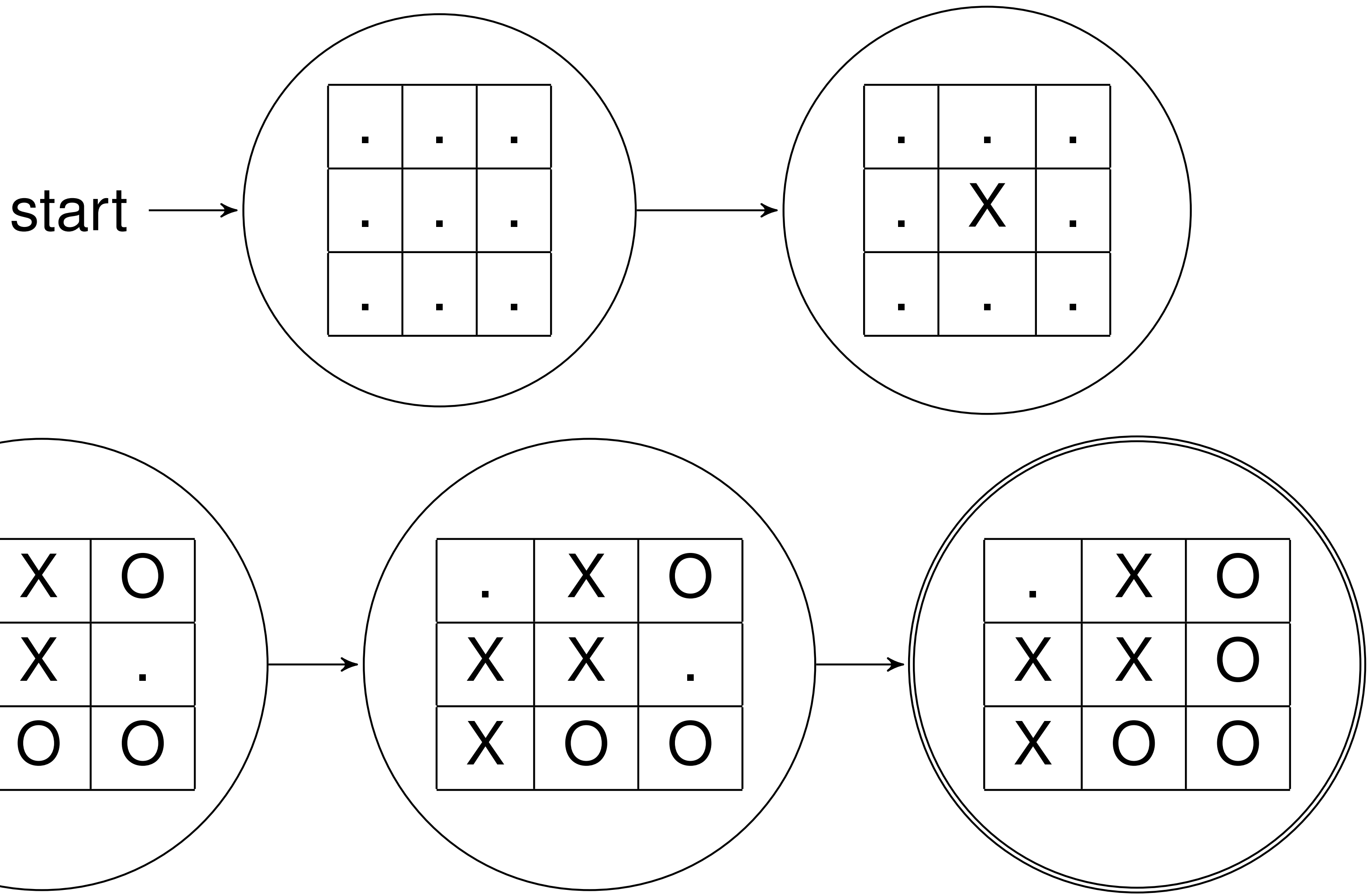
Code each possible $3 * 3$ board as a node of the graph.

Each possible board has n markers X placed by Anke and m markers O placed by Boris such that $0 \leq m \leq n \leq m+1 \leq 5$.

The empty board is the start position and the players move alternately by placing X and O into empty fields. Anke wins if there are three X in a row, column or diagonal; Boris wins if there are three O in a row, column or diagonal. T is the set of all nodes where a player wins or the board is full.

Players move alternately and the following invariants hold: If the game is not in a target node and there are m of X and O each then Anke can move and places an X in an empty field; if the game is not in a target node and there are $m+1$ of X and m of O then Boris can move and places an O in an empty field.

Game Graph



Comment

Tic Tac Toe is a bit different from previous games as the game can get stuck without a winner to be declared.

Such positions are also considered as draw positions.

If such dead ends exist, the algorithm to decide which nodes are winning and losing has to be adjusted.

Furthermore, Tic Tac Toe is a moderate game, as there are only $3^9 = 19683$ many board positions of which many cannot be reached by alternate moves (like 6 times X and 3 times O or having a row full of X plus a row full of O).

Computers can play this game optimally and even compute a table which gives for each possible board the optimal move in the case that this situation arises; in the case that there are several moves of the same quality, the computer might chose by random one of them for having variations when playing the game repeatedly.

Deciding Games

For strategic games with two alternately moving players without random aspects, there are three possibilities (plus unknown). Here what is known for famous games.

- The first player has a winning strategy: Connect Four, Hex (on $n * n$ board), 15$*$15 Gomoku (no opening rules).
- The second player has a winning strategy: 4$*$4 Othello, 6$*$6 Othello.
- Both players have a draw strategy: Draughts (Checkers), Nine Men's Morris, Tic Tac Toe.
- Unknown: Chess, Go, 19$*$19 Gomoku, 8$*$8 Othello (conjecture: draw).

`http://en.wikipedia.org/wiki/Solved_game`

Not in this list

Games involving random aspects (cards, dices, ...) do not have perfect strategies. The reason is that a move which is good with high probability might turn out to be bad if some unlikely random event happens. Nevertheless, computers might be better than humans in playing these games.

Multiplayer games usually do not have winning strategies as at 3 players, 2 might collaborate to avoid that the third player wins (although they should not do it).

Therefore the above analysis was for 2-player games without random aspects. If there is just a random starting point (in the graph), but no other random event, one can determine for each possible starting point which player has a winning strategy when starting from there.

Games might still be unsolved due to the high complexity which an algorithmic solution of the game would need.

Exercise 4.10

Let $\mathbf{Divandinc_{n,m}}$ be given by the graph with domain $\{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}\}$, starting state $\mathbf{m} \in \{\mathbf{2}, \ldots, \mathbf{n}\}$ and target state $\mathbf{1}$. Furthermore, each player can move from $\mathbf{k} \in \{\mathbf{2}, \ldots, \mathbf{n}\}$ to $\ell \in \{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}\}$ iff either $\ell = \mathbf{k} + \mathbf{1}$ or $\ell = \mathbf{k}/\mathbf{p}$ for some prime number $\mathbf{p}$.

(a) Show that every position is either a winning position for Anke or for Boris. In particular, whenever the game goes through an infinite sequence of moves then some player leaves out a possibility to win.

(b) Show that if $\mathbf{m} \leq \mathbf{n} \leq \mathbf{n'}$ and $\mathbf{n}$ is a prime number, then the player who can win $\mathbf{Divandinc_{n,m}}$ can also win $\mathbf{Divandinc_{n',m}}$.

(c) Find values $\mathbf{m}, \mathbf{n}, \mathbf{n'}$ with $\mathbf{m} < \mathbf{n} < \mathbf{n'}$ where Anke has a winning strategy for $\mathbf{Divandinc_{n,m}}$ and Boris for $\mathbf{Divandinc_{n',m}}$.

Variants of Graph Games

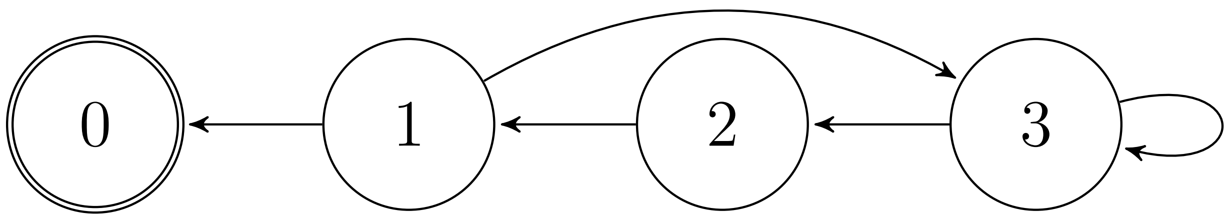
One can vary the setting of graph games:

- The set of nodes is partitioned into sets $\mathbf{A}, \mathbf{B}$ such that every node is in exactly one of these sets and player Anke moves iff the marker is in $\mathbf{A}$ and player Boris moves iff the marker is in $\mathbf{B}$;
- There are three disjoint sets of nodes $\mathbf{T_A}, \mathbf{T_B}, \mathbf{T_D}$ of target nodes; Anke wins when the game ends up in $\mathbf{T_A}$, Boris wins if the game ends up in $\mathbf{T_B}$, the game is draw when ending up in $\mathbf{T_D}$. A node is in one of these three sets iff it has no outgoing edges.

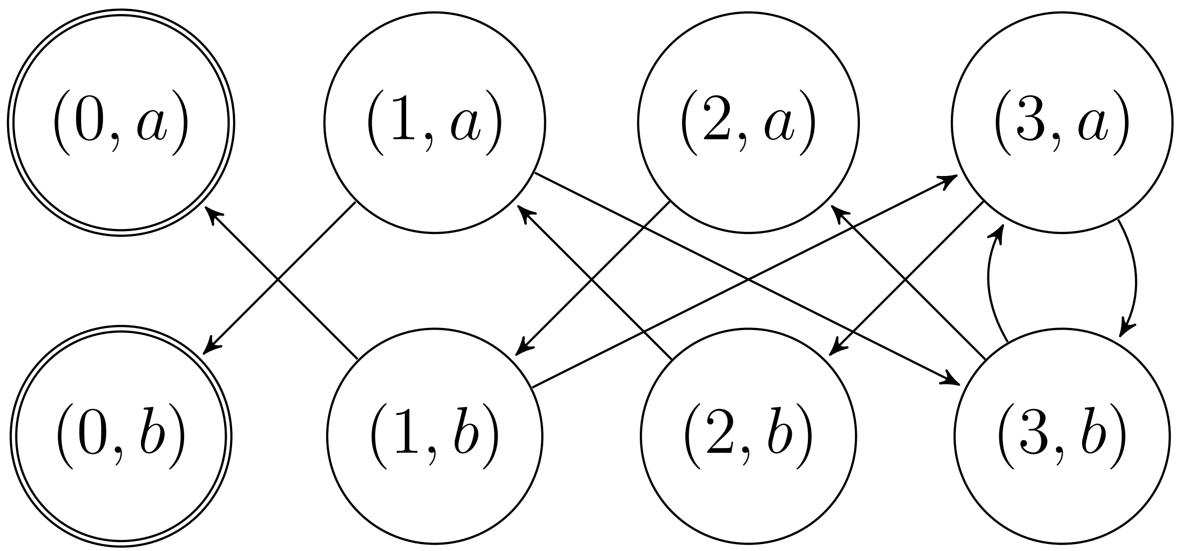
Tic Tac Toe can be made to satisfy the above constraints.

Example 4.12

Assume that a game with states $\mathbf{Q}$ and target set $\mathbf{T}$ is given. Now consider a new game with nodes $\mathbf{Q} \times \{\mathbf{a}, \mathbf{b}\}$ and edges $(\mathbf{p}, \mathbf{a}) \to (\mathbf{q}, \mathbf{b})$ and $(\mathbf{p}, \mathbf{b}) \to (\mathbf{q}, \mathbf{a})$ whenever $\mathbf{p} \to \mathbf{q}$ in the old game, $\mathbf{T_A} = \mathbf{T} \times \{\mathbf{b}\}$, $\mathbf{T_B} = \mathbf{T} \times \{\mathbf{a}\}$. The game



with $\mathbf{T} = \{\mathbf{0}\}$ is translated into the below one with $\mathbf{A} = \mathbf{Q} \times \{\mathbf{a}\}$, $\mathbf{B} = \mathbf{Q} \times \{\mathbf{b}\}$, $\mathbf{T_A} = \{(\mathbf{0}, \mathbf{b})\}$, $\mathbf{T_B} = \{(\mathbf{0}, \mathbf{a})\}$.



Exercise 4.13

Design a game with $\mathbf{A}, \mathbf{B}$ being disjoint nodes of Anke and Boris and the edges chosen such that

- the players move alternately;
- the sets $\mathbf{T_A}, \mathbf{T_B}$ of the winning nodes are disjoint;
- every node outside $\mathbf{T_A} \cup \mathbf{T_B}$ has outgoing edges, that is, $\mathbf{T_D} = \emptyset$;
- the so designed game is not an image of a symmetric game in the way it was done in the previous example.

Which properties of the game can be used to enforce that?

Exercise 4.14

The following game satisfies the second constraint from Remark 4.11 and has an infinite game graph.

Assume that $\mathbf{Q} = \mathbb{N}$, $\mathbf{x+4, x+3, x+2, x+1 \rightarrow x}$ for all $\mathbf{x} \in \mathbb{N}$ with the exception that nodes in $\mathbf{T_A}$ and $\mathbf{T_B}$ have no outgoing edges where $\mathbf{T_A = \{0, 6, 9\}}$ and $\mathbf{T_B = \{5, 7, 12, 17\}}$.

If the play of the game reaches a node in $\mathbf{T_A}$ then Anke wins and if it reaches a node in $\mathbf{T_B}$ then Boris wins. Note that if the game starts in nodes from $\mathbf{T_A}$ or $\mathbf{T_B}$ then it is a win for Anke or Boris in $\mathbf{0}$ moves, respectively.

Determine for both players (Anke and Boris) which are the winning positions for them. Are there any draw positions?

Alternating Automata

Definition. Anke and Boris decide on moves in nfa while processing a word $\mathbf{w}$. Three possibilities for pairs $(\mathbf{q}, \mathbf{a})$ of states $\mathbf{q}$ and symbols $\mathbf{a}$:

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

The afa accpets a word $\mathbf{w}$ iff Anke has a winning strategy.

Example. States $\{\mathbf{p}, \mathbf{q}, \mathbf{r}\}$; alphabet $\{\mathbf{0}, \mathbf{1}\}$; language $\{\mathbf{0}, \mathbf{1}\}^* \cdot \mathbf{1}$.

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	$\mathbf{p} \wedge \mathbf{q} \wedge \mathbf{r}$	$\mathbf{p} \vee \mathbf{q}$

Determinising Afas

Given afa with state-set $\mathbf{Q}$ and accepting states $\mathbf{F}$.

Alternating to Non-Deterministic Automaton

States: All non-empty $\mathbf{P} \subseteq \mathbf{Q}$; $\mathbf{P}$ is accepting iff $\mathbf{P} \subseteq \mathbf{F}$. Non-deterministic transitions: $\mathbf{P} \subseteq \mathbf{Q}$ to $\mathbf{R}$ on $\mathbf{a}$ by looking at all $\mathbf{p} \in \mathbf{P}$ and check the type of transition. If $\mathbf{p} \to \mathbf{q} \vee \mathbf{r}$ choose one of the successors and put it into $\mathbf{R}$ else put all successors into $\mathbf{R}$.

Alternating to Deterministic Automaton

States are non-empty sets of non-empty subsets of $\mathbf{Q}$. For each set $\mathbf{M}$ of subsets of $\mathbf{Q}$ and each state $\mathbf{A}$, put for all $\mathbf{P} \in \mathbf{A}$ all the $\mathbf{R}$ into the successor-state $\mathbf{B}$ which can be chosen by above nfa. However, if there are $\mathbf{R}, \mathbf{R}' \in \mathbf{B}$ with $\mathbf{R} \subset \mathbf{R}'$ then one can remove $\mathbf{R}'$ from $\mathbf{B}$. State $\mathbf{A}$ is accepting iff there is a $\mathbf{P} \subseteq \mathbf{F}$ contained in $\mathbf{A}$.

Double Exponential Growth

Alphabet $\{0, 1, \ldots, n\}$, states $\{s, q, p_1, \ldots, p_n, r_1, \ldots, r_n\}$, set $\{p_1, \ldots, p_n\}$ of accepting states.

state	0	i	$j \notin \{0, i\}$
s	$s \vee q$	s	s
q	$p_1 \wedge \ldots \wedge p_n$	q	q
p_i	p_i	r_i	p_i
r_i	r_i	p_i	r_i

The language recognised by the afa contains all words of the form $x0y0z$ where $x, y, z \in \{0, 1, \ldots, n\}^*$ and z contains each non-zero digit an even number of times.

Afa has $2n + 2$ states, dfa has $2^{2^n} + 1$ states.

Exercise 4.19. Show that an afa with two states can be converted into a dfa with four states; this bound is optimal.

Product of Automata

Theorem 4.20
If there are $\mathbf{n}$ dfas $(\mathbf{Q_i}, \mathbf{\Sigma}, \delta_\mathbf{i}, \mathbf{s_i}, \mathbf{F_i})$ with $\mathbf{m}$ states each recognising $\mathbf{L_1}, \ldots, \mathbf{L_n}$, respectively, then there is an afa recognising $\mathbf{L_1} \cap \ldots \cap \mathbf{L_n}$ with $\mathbf{1} + \mathbf{mn}$ states.

Wlog the $\mathbf{Q_i}$ are pairwise disjoint and let $\mathbf{s} \notin \bigcup_\mathbf{i} \mathbf{Q_i}$ and $\mathbf{Q} = \{\mathbf{s}\} \cup \bigcup_\mathbf{i} \mathbf{Q_i}$.

On $\mathbf{a}$ let $\mathbf{s} \to \delta_\mathbf{1}(\mathbf{s_1}, \mathbf{a}) \wedge \ldots \wedge \delta_\mathbf{n}(\mathbf{s_n}, \mathbf{a})$; furthermore, for all $\mathbf{Q_i}$ and $\mathbf{q_i} \in \mathbf{Q_i}$, on $\mathbf{a}$ let $\mathbf{q_i} \to \delta_\mathbf{i}(\mathbf{q_i}, \mathbf{a})$.

The state $\mathbf{s}$ is accepting iff $\varepsilon \in \mathbf{L_1} \cap \ldots \cap \mathbf{L_n}$ and $\mathbf{q_i} \in \mathbf{Q_i}$ is accepting iff $\mathbf{q_i} \in \mathbf{Q_i}$.

Example 4.21

Let $\mathbf{L_i}$ contain the word with an even number of digit $\mathbf{i}$ and $\mathbf{\Sigma = \{0, 1, \ldots, n\}}$, $\mathbf{n = 3}$. Now $\mathbf{Q_i = \{s_i, t_i\}}$ and $\mathbf{F_i = \{s_i\}}$.
If $\mathbf{i = j}$ then $\delta_\mathbf{i}(\mathbf{s_i, j}) = \mathbf{t_i}, \delta_\mathbf{i}(\mathbf{t_i, j}) = \mathbf{s_i}$
else $\delta_\mathbf{i}(\mathbf{s_i, j}) = \mathbf{s_i}, \delta_\mathbf{i}(\mathbf{t_i, j}) = \mathbf{t_i}$.

Now $\mathbf{Q = \{s, s_1, s_2, s_3, t_1, t_2, t_3}$, on $\mathbf{0}$, $\mathbf{s \to s_1 \wedge s_2 \wedge s_3}$, on $\mathbf{1}$, $\mathbf{s \to t_1 \wedge s_2 \wedge s_3}$, on $\mathbf{2}$, $\mathbf{s \to s_1 \wedge t_2 \wedge s_3}$, on $\mathbf{3}$, $\mathbf{s \to s_1 \wedge s_2 \wedge t_3}$. On $\mathbf{j}$, $\mathbf{s_i} \to \delta_\mathbf{i}(\mathbf{s_i, j})$ and $\mathbf{t_i} \to \delta_\mathbf{i}(\mathbf{t_i, j})$. The states $\mathbf{s, s_1, s_2, s_3}$ are accepting.

Word $\mathbf{2021}$: On $\mathbf{2}$: $\mathbf{s \to s_1 \wedge t_2 \wedge s_3}$;
On $\mathbf{0}$: $\mathbf{s_1 \wedge t_2 \wedge s_3 \to s_1 \wedge t_2 \wedge s_3}$;
On $\mathbf{2}$: $\mathbf{s_1 \wedge t_2 \wedge s_3 \to s_1 \wedge s_2 \wedge s_3}$;
On $\mathbf{1}$: $\mathbf{s_1 \wedge s_2 \wedge s_3 \to t_1 \wedge s_2 \wedge s_3}$.
The last state rejects because the conjunction contains a rejecting state.

Intersection of nfas

Exercise 4.22

If there are $\mathbf{n}$ nfas $(\mathbf{Q_i}, \boldsymbol{\Sigma}, \delta_\mathbf{i}, \mathbf{s_i}, \mathbf{F_i})$ with $\mathbf{m}$ states each recognising $\mathbf{L_1}, \ldots, \mathbf{L_n}$, respectively, show that there is an afa recognising $\mathbf{L_1} \cap \ldots \cap \mathbf{L_n}$ with $\mathbf{1} + (\mathbf{m} + |\boldsymbol{\Sigma}|) \cdot \mathbf{n}$ states.

In particular, for $\mathbf{n} = \mathbf{2}$ and $\boldsymbol{\Sigma} = \{\mathbf{0}, \mathbf{1}, \mathbf{2}\}$, construct explicitly nfas and the product afa where $\mathbf{L_1}$ is the language of all words where the last letter has already appeared before and $\mathbf{L_2}$ is the language of all words where at least one letter appears an odd number of times.

The proof can be done by adapting the one of Theorem 4.20 and use nfas in place of dfas. The main adjustment is that in the first step one goes to new, conjunctively connected states which have to memorise the character just seen. From then on, all rules are disjunctive and not deterministic as in Theorem 4.20.

Exercise 4.23

Assume that a game has an infinite board $\mathbb{N}$ and starts with three numbers $\mathbf{a}, \mathbf{b}, \mathbf{c}$ such that $\mathbf{a} < \mathbf{b} < \mathbf{c}$; the initial value is $\mathbf{a} = \mathbf{12}, \mathbf{b} = \mathbf{13}, \mathbf{c} = \mathbf{14}$. Possible moves are to increment one of the numbers by $\mathbf{1}$, as long as the condition on the order of the numbers is not violated. The game ends with a winner, when $\mathbf{c}$ becomes the double of $\mathbf{a}$. Anke starts to move. Is this game a winning game for Anke, a winning game for Boris or a draw game. Provide the winning strategy of the respective player or the draw strategy for both players.

Exercises 4.24-4.26

4.24: A game has fields $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$ and two markers which initially stand on 0 and 6. The players move alternately one of the markers by adding, modulo 12, either 1 or 2 to its position. When a player makes a move such that both markers stand on the same field, the game ends and the player wins. Is this game a winning game for Anke, a winning game for Boris or a draw game. Provide the winning strategy of the respective player or the draw strategy for both players.

4.25: Find a regular language L and a number n such that both the best dfa and nfa have n states but some afa needs less states.

4.26: Construct an afa for the language of all decimal numbers which are not divisible by any 1-digit prime number.

Exercises 4.27-4.30

Consider games on decimal numbers $a_n a_{n-1} \ldots a_1 a_0$. Players Anke and Boris move alternately. Determine for the below games which player wins from the following start situations: 300, 288, 1111, 1024. The player who reaches 0 wins. Let x denote the current number when the move is to be made, for each nonzero x, some move have to be made.

4.27: The player can replace x by $x - y$ where y is odd and $y \leq x$.

4.28: The player can reduce one non-zero digit by 1.

4.29: The player can replace one digit by a digit which is one or two or three smaller.

4.30: The player can reduce one nonzero digit a_m by 1 and change (optional) one digit a_k with $k < m$ to an arbitrary value from $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$.

Exercises 4.32-4.34

Consider the following game on binary numbers. The player make the number smaller by either changing a 1 to a 0 or by interchanging a 1 with a more behind 0. The player making the number to 0 wins. In the next two exercises, determine which of the given binary numbers are winning for Anke, who moves first. Sketch the winning strategies for the winner to win the games.

Exercise 4.32

$\mathbf{1010}$, $\mathbf{10101010}$, $\mathbf{10000}$, $\mathbf{100001}$, $\mathbf{1111}$.

Exercise 4.33

$\mathbf{1110}$, $\mathbf{111100}$, $\mathbf{110011001100}$, $\mathbf{101111}$, $\mathbf{101010}$, $\mathbf{10101000}$.

Exercise 4.34

Provide a regular infinite set of binary numbers such that each of them is a winning position for Boris (the player who moves second). Prove that this set works.