Advanced Automata Theory 12
Open Problems in Automata Theory

Frank Stephan

Department of Computer Science

Department of Mathematics

National University of Singapore

fstephan@comp.nus.edu.sg

Repetition: Learning Theory

Formal models of learning processes

Query Learning

Dialogue between learner (pupil) and teacher.
Learner asks queries using a specific query language.
Teacher provides answers to these queries.

Learning from Data

Learner reads more and more data about the concept to be learnt.
In response to the data, learner conjectures one or several hypotheses.
The last hypothesis conjectured must be correct.

Repetition: Angluin's DFA Learner

Query language – Learner can ask the following questions:

- Equivalence query (EQ): Does this DFA recognise the language $\mathbf{L}$ to be learnt?
  Teacher says "YES" or "NO"; when saying "NO" an $\mathbf{x}$ is provided on which the DFA does the opposite of what $\mathbf{L}$ does.
- Membership query (MQ): Is $\mathbf{x}$ a member of $\mathbf{L}$?
  Teacher says "YES" or "NO".

Learner could use a list of all DFAs and ask: does the first DFA recognise $\mathbf{L}$; does the second DFA recognise $\mathbf{L}$; ...?

This needs more than $\mathbf{2^n}$ steps in order to deal with all languages recognised by DFA's with $\mathbf{n}$ states.

Angluin's algorithm needs much less queries.

Repetition: Learning from Data

A text $\mathbf{T}$ for $\mathbf{L}$ is an infinite sequence $\mathbf{T(0)}, \mathbf{T(1)}, \ldots$ of strings and pause symbols $\#$ describing $\mathbf{L}$: $\mathbf{w} \in \mathbf{L}$ iff $\mathbf{w} \neq \#$ and $\mathbf{w} = \mathbf{T(n)}$ for some $\mathbf{n}$.

Learner $\mathbf{M}$ starts with initial memory $\mathbf{mem_0}$ and hypothesis $\mathbf{e_0}$. In the $\mathbf{n}$-th cycle, $\mathbf{M}$ updates $\mathbf{mem_n}$ and input $\mathbf{T(n)}$ to $\mathbf{mem_{n+1}}$ and $\mathbf{e_{n+1}}$; in short: $\mathbf{M}$ maps $\mathbf{(mem_n, T(n))}$ to $\mathbf{(mem_{n+1}, e_{n+1})}$. The values $\mathbf{e_1}, \mathbf{e_2}, \ldots$ depend on $\mathbf{T}$.

Learner $\mathbf{M}$ learns $\mathbf{L}$ iff, on every text $\mathbf{T}$ for $\mathbf{L}$, almost all hypotheses $\mathbf{e_n}$ of $\mathbf{M}$ are the same $\mathbf{e}$ describing $\mathbf{L}$.

Repetition: Learnability

Theorem [Gold 1967]
The class of all regular languages cannot be learnt from positive data.

Theorem [Angluin 1980]
An automatic family $\{\mathbf{L_e} : \mathbf{e} \in \mathbf{I}\}$ can be learnt from positive data iff there is for every $\mathbf{e} \in \mathbf{I}$ a finite subset $\mathbf{F_e} \subseteq \mathbf{L_e}$ such that there is no $\mathbf{d} \in \mathbf{I}$ with $\mathbf{F_e} \subseteq \mathbf{L_d} \subset \mathbf{L_e}$.

Example
The class of $\mathbf{L}_\varepsilon = \mathbf{\Sigma}^*$ and $\mathbf{L_x} = \{\mathbf{y} \in \mathbf{\Sigma}^* : |\mathbf{y}| < |\mathbf{x}|\}$ with $\mathbf{x} \in \mathbf{\Sigma}^+$ does not satisfy Angluin's tell-tale condition; hence this class is not learnable.

Repetition: Automatic Learners

Automatic learner given by $\mathbf{mem_0}, \mathbf{e_0}$ and
$\mathbf{uf} : \mathbf{conv}(\mathbf{mem_n}, \mathbf{x_n}) \mapsto \mathbf{conv}(\mathbf{mem_{n+1}}, \mathbf{e_{n+1}})$.
$\mathbf{mem_n}$ is a string in an arbitrary alphabet;
$\mathbf{e_n}$ is a hypothesis from $\mathbf{I}$ when learning $\{\mathbf{L_e} : \mathbf{e} \in \mathbf{I}\}$;
$\mathbf{uf}$ is an automatic update function of the learner.

Automatic learners can only remember few information about the past, as there is a constant $\mathbf{c}$ with $|\mathbf{mem_{n+1}}| \leq \max\{|\mathbf{mem_n}|, |\mathbf{x_n}|\} + \mathbf{c}$ for all $\mathbf{n}$.

Explicit memory bounds (with constant $\mathbf{k}$) when learning $\mathbf{L_d}$:

- Word-sized: $|\mathbf{mem_{n+1}}| \leq \max\{|\mathbf{x_0}|, |\mathbf{x_1}|, \ldots, |\mathbf{x_n}|\} + \mathbf{k}$.
- Hypothesis-sized: $|\mathbf{mem_{n+1}}| \leq |\mathbf{e_{n+1}}| + \mathbf{k}$.
- Target-sized: $|\mathbf{mem_{n+1}}| \leq |\mathbf{d}| + \mathbf{k}$.
- Constant: $|\mathbf{mem_{n+1}}| \leq \mathbf{k}$.

Repetition: Example 11.7

Assume $\mathbf{\Sigma = \{0, 1, 2\}}$ and
$\mathbf{I = \{conv(v, w) : v, w \in \Sigma^* \wedge v \leq_{lex} w\} \cup \{conv(3, 3)\}}$ with
$\mathbf{L_{conv(v,w)} = \{u \in \Sigma^* : v \leq_{lex} u \leq_{lex} w\}}$ for all
$\mathbf{conv(v, w) \in I}$; note that $\mathbf{L_{conv(3,3)} = \emptyset}$.

Data seen so far	Hypothesis	Conjectured language
—	conv(3,3)	$\emptyset$
#	conv(3,3)	$\emptyset$
# 00	conv(00,00)	$\{00\}$
# 00 0000	conv(00,0000)	$\{00, 000, 0000\}$
# 00 0000 1	conv(00,1)	$\{u : 00 \leq_{lex} u \leq_{lex} 1\}$
# 00 0000 1 0	conv(0,1)	$\{u : 0 \leq_{lex} u \leq_{lex} 1\}$
# 00 0000 1 0 112	conv(0,112)	$\{u : 0 \leq_{lex} u \leq_{lex} 112\}$
# 00 0000 1 0 112 #	conv(0,112)	$\{u : 0 \leq_{lex} u \leq_{lex} 112\}$

Repetition: Limitations

Theorem 11.10

Let $\mathbf{I} = \mathbf{\Sigma}^*$, $\mathbf{L}_\varepsilon = \mathbf{\Sigma}^+$ and $\mathbf{L_d} = \{\mathbf{w} \in \mathbf{\Sigma}^* : \mathbf{w} <_{\mathbf{ll}} \mathbf{d}\}$ for $\mathbf{d} \in \mathbf{\Sigma}^+$. The class $\{\mathbf{L_d} : \mathbf{d} \in \mathbf{I}\}$ can be learnt using a word-sized memory but not using an hypothesis-sized memory.

Word-sized Learner

Memory are length-lexicographically smallest and largest words seen so far.

If ε has not been seen, conjecture $\mathbf{\Sigma}^+$. If ε and $\mathbf{w}$ are minimal and maximal word seen so far then let $\mathbf{d}$ be the successor of $\mathbf{w}$ and conjecture $\mathbf{L_d}$.

Overview

Open Problems from Work in Singapore

These problems are not that famous but also not that difficult. There is some chance to get them out with some good ideas.

Famous Open Problems

Some of these problems have been open for many years or decades. They are difficult to get out. However, once, the undergraduate student Róbert Szelepcsényi from Slovakia (at that time ČSSR) solved one problem of this type by showing that context-sensitive classes are closed under complement; he and Immerman from the USA solved this problem independently.

Open Problems from Lecture 11

1. Does every automatic family which has an automatic learner also have a learner with word-sized memory?
2. Does every automatic family which has a learner with hypothesis-sized memory also have a learner with word-sized memory?
3. Does every automatic family which has a learner with hypothesis-sized memory also have an iterative learner?

Many-One and One-One Reducibilities

One says a set $\mathbf{A}$ is many-one reducible to $\mathbf{B}$ via $\mathbf{f}$ iff $\forall \mathbf{x}\,[\mathbf{A}(\mathbf{x}) = \mathbf{B}(\mathbf{f}(\mathbf{x}))]$. If $\mathbf{f}$ is one-one then $\mathbf{A}$ is one-one reducible to $\mathbf{B}$ via $\mathbf{f}$.

The set $\mathbf{A}$ is many-one / one-one reducible to $\mathbf{B}$ iff there is a function $\mathbf{f}$ witnessing this of a corresponding type (like being automatic or rational).

Incomparable sets: For alphabet $\{\mathbf{0}, \mathbf{1}\}$, the sets $\{\mathbf{0}\}^*$ and $\{\mathbf{0}\}^* \cdot \{\mathbf{1}\} \cdot \{\mathbf{0}, \mathbf{1}\}^*$ are incomparable with respect to automatic one-one reducibility.

Automatic Reducibility

Reducibilities without base set

Let $\mathbf{A} \leq_{\mathbf{au}} \mathbf{B}$ denote that there is an automatic function $\mathbf{f}$ such that $\forall \mathbf{x}, \mathbf{y} \in \mathbf{A} \, [\mathbf{x} \neq \mathbf{y} \Rightarrow \mathbf{f}(\mathbf{x}) \neq \mathbf{f}(\mathbf{y}) \wedge \mathbf{f}(\mathbf{x}) \in \mathbf{B}]$.

Similarly one writes $\mathbf{A} \leq_{\mathbf{tr}} \mathbf{B}$ for the corresponding definition where $\mathbf{f}$ is any function computed by a finite transducer.

Example

$\{\mathbf{0}\}^* \cup \{\mathbf{1}\}^* \leq_{\mathbf{tr}} \{\mathbf{2}\}^*$ and $\{\mathbf{0}\}^* \cup \{\mathbf{1}\}^* \not\leq_{\mathbf{au}} \{\mathbf{2}\}^*$.

A regular set $\mathbf{A}$ has size $\mathbf{\Theta}(\mathbf{n^k})$ iff there is $\mathbf{c} > \mathbf{0}$ with

$$\forall \mathbf{n} \, [\mathbf{n^k}/\mathbf{c} - \mathbf{c} \leq |\{\mathbf{x} \in \mathbf{A} : |\mathbf{x}| \leq \mathbf{n}\}| \leq \mathbf{n^k} \cdot \mathbf{c} + \mathbf{c}]$$

and if $\mathbf{A}$ is infinite and does not have size $\mathbf{\Theta}(\mathbf{n^k})$ for any $\mathbf{k}$ then $\mathbf{A}$ is exponential-sized.

Results of Tan Wai Yean

Theorem [Tan 2010]
Let $\mathbf{A}, \mathbf{B}$ be regular sets.

1. The sets $\mathbf{A}, \mathbf{B}$ are comparable for $\mathbf{tr}$-reducibility: $\mathbf{A} \leq_{\mathbf{tr}} \mathbf{B}$ or $\mathbf{B} \leq_{\mathbf{tr}} \mathbf{A}$. Furthermore, $\mathbf{A} \leq_{\mathbf{tr}} \mathbf{B}$ if one of the following conditions holds:
   - $\mathbf{A}, \mathbf{B}$ are both finite and $|\mathbf{A}| \leq |\mathbf{B}|$;
   - $\mathbf{A}$ is finite and $\mathbf{B}$ infinite;
   - $\mathbf{A}$ has size $\mathbf{\Theta(n^k)}$ and $\mathbf{B}$ has size $\mathbf{\Theta(n^h)}$ with $\mathbf{k \leq h}$;
   - $\mathbf{B}$ is exponential-sized.
2. If $\mathbf{A}$ is polynomial-sized or finite then $\mathbf{A} \leq_{\mathbf{au}} \mathbf{B}$ or $\mathbf{B} \leq_{\mathbf{au}} \mathbf{A}$. If $\mathbf{A}$ is of size $\mathbf{\Theta(n^k)}$, $\mathbf{B}$ is of size $\mathbf{\Theta(n^h)}$ and $\mathbf{k < h}$ then $\mathbf{A} \leq_{\mathbf{au}} \mathbf{B}$ and $\mathbf{B} \not\leq_{\mathbf{au}} \mathbf{A}$.

Exercise and Open Problem

Exercise 12.4

Make an automatic one-one function which maps the domain $\mathbf{A} = \mathbf{0}^*(\mathbf{1}^* \cup \mathbf{2}^*)$ to a subset of $\mathbf{B} = (\mathbf{0000})^*(\mathbf{1111})^*(\mathbf{2222})^*$, that is, show that $\mathbf{A} \leq_{\mathbf{au}} \mathbf{B}$.

Open Problem [Tan 2010]

Are there regular sets $\mathbf{A}, \mathbf{B}$ such that $\mathbf{A} \not\leq_{\mathbf{au}} \mathbf{B}$ and $\mathbf{B} \not\leq_{\mathbf{au}} \mathbf{A}$?

Remark

There are context-free sets $\mathbf{A}, \mathbf{B}$ with $\mathbf{A} \not\leq_{\mathbf{au}} \mathbf{B}$ and $\mathbf{B} \not\leq_{\mathbf{au}} \mathbf{A}$: $\mathbf{A} = \{\mathbf{0}\}^*$ and $\mathbf{B} = \{\mathbf{x2y} : \mathbf{x}, \mathbf{y} \in \{\mathbf{0}, \mathbf{1}\}^* \wedge |\mathbf{x}| = |\mathbf{y}|\}$.

The XX-Problem

For the following, consider the fixed one-one reduction $\mathbf{x} \mapsto \mathbf{xx}$ and $\mathbf{x} \mapsto \mathbf{x}(\mathbf{x}^{\mathbf{mi}})$.

Open Problem [Zhang 2013]

1. Given a regular language $\mathbf{A}$, is there a regular language $\mathbf{B}$ such that, for all $\mathbf{x}$, $\mathbf{A}(\mathbf{x}) = \mathbf{B}(\mathbf{xx})$?
2. Given a context-free language $\mathbf{A}$, is there a context-free language $\mathbf{B}$ such that, for all $\mathbf{x}$, $\mathbf{A}(\mathbf{x}) = \mathbf{B}(\mathbf{xx})$?

Zhang [2013] solved the reverse directions: If $\mathbf{B}$ is regular then the set $\mathbf{A} = \{\mathbf{x} : \mathbf{xx} \in \mathbf{B}\}$ is also regular. However, the set $\mathbf{B} = \{\mathbf{0^n 1^n 2^m 0^m 1^k 2^k} : \mathbf{n, m, k} \in \mathbb{N}\}$ is context-free while the corresponding set $\mathbf{A} = \{\mathbf{x} : \mathbf{xx} \in \mathbf{B}\} = \{\mathbf{0^n 1^n 2^n} : \mathbf{n} \in \mathbb{N}\}$ is properly context-sensitive.

The XM-Problem

Theorem [Fung 2014]
Assume that $\mathbf{A}$ is a regular set and let $\mathbf{B} = \{\mathbf{u} :$ there are an odd number of pairs $(\mathbf{y}, \mathbf{z})$ with $\mathbf{u} = \mathbf{yz}$, $\mathbf{y} \in \mathbf{A}$ and $\mathbf{z^{mi}} \in \mathbf{A}\}$.
Then $\mathbf{B}$ is regular and for all $\mathbf{x}$, $\mathbf{x} \in \mathbf{A} \Leftrightarrow \mathbf{x} \cdot \mathbf{x^{mi}} \in \mathbf{B}$.

Open Problem
Given a context-free set $\mathbf{A}$, is there a context-free set $\mathbf{B}$ such that for all $\mathbf{x}$, $\mathbf{x} \in \mathbf{A} \Leftrightarrow \mathbf{x} \cdot \mathbf{x^{mi}} \in \mathbf{B}$?

Ordered Groups

An ordered group $(\mathbf{G}, +, <, \mathbf{0})$ satisfies the group and order axioms and for all $\mathbf{x}, \mathbf{y}, \mathbf{z}$ that $\mathbf{x} < \mathbf{y}$ implies $\mathbf{x} + \mathbf{z} < \mathbf{y} + \mathbf{z}$ and $\mathbf{z} + \mathbf{x} < \mathbf{z} + \mathbf{y}$.

Theorem [Jain, Khoussainov, Stephan, Teng and Zou 2014] If an ordered group has a regular domain and fully automatic addition then the group is commutative.

Question [Khoussainov] Is there a fully automatic copy of $(\mathbb{Z}, +, \mathbf{0})$ such that the set $\{\mathbf{x} \in \mathbb{Z} : \mathbf{x} > \mathbf{0}\}$ is not regular?

Theorem [Jain, Khoussainov, Stephan, Teng and Zou 2014] There is a fully automatic copy of $(\{\mathbf{x} \cdot \mathbf{6^y} : \mathbf{x}, \mathbf{y} \in \mathbb{Z}\}, +, \mathbf{0})$ in which neither the order nor the set of positive numbers is automatic.

Synchronising words

Definition

Given a dfa with states $\mathbf{Q}$ and transition function δ, a word $\mathbf{w}$ is called synchronising iff $\delta(\mathbf{p}, \mathbf{w}) = \delta(\mathbf{q}, \mathbf{w})$ for all $\mathbf{p}, \mathbf{q} \in \mathbf{Q}$.

Example

The dfa which accepts all decimal numbers which are multiples of $\mathbf{3}$ does not have a synchronising word.

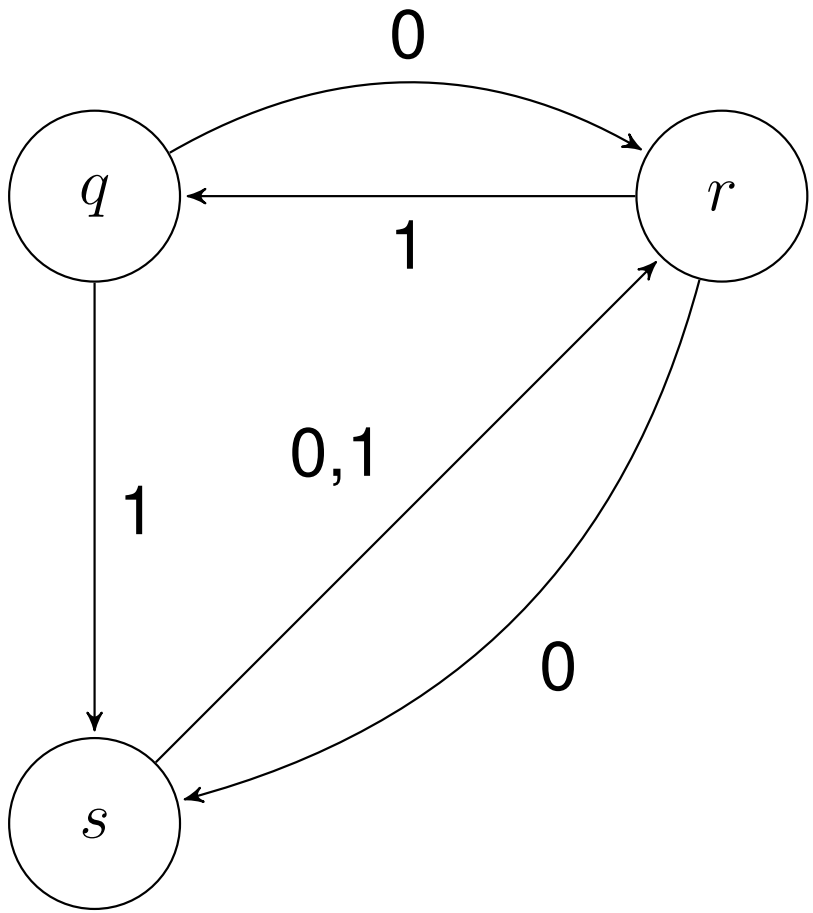
Example

The dfa of two states which accepts a decimal number iff it ends with $\mathbf{0}, \mathbf{2}, \mathbf{4}, \mathbf{6}, \mathbf{8}$ and rejects a decimal number iff it ends with $\mathbf{1}, \mathbf{3}, \mathbf{5}, \mathbf{7}, \mathbf{9}$ has a synchronising word.

Theorem [Černý 1964]

For each $\mathbf{n}$ there is a complete dfa with $\mathbf{n}$ states which has a synchronising word of length $(\mathbf{n}-\mathbf{1})^{\mathbf{2}}$ and no shorter ones.

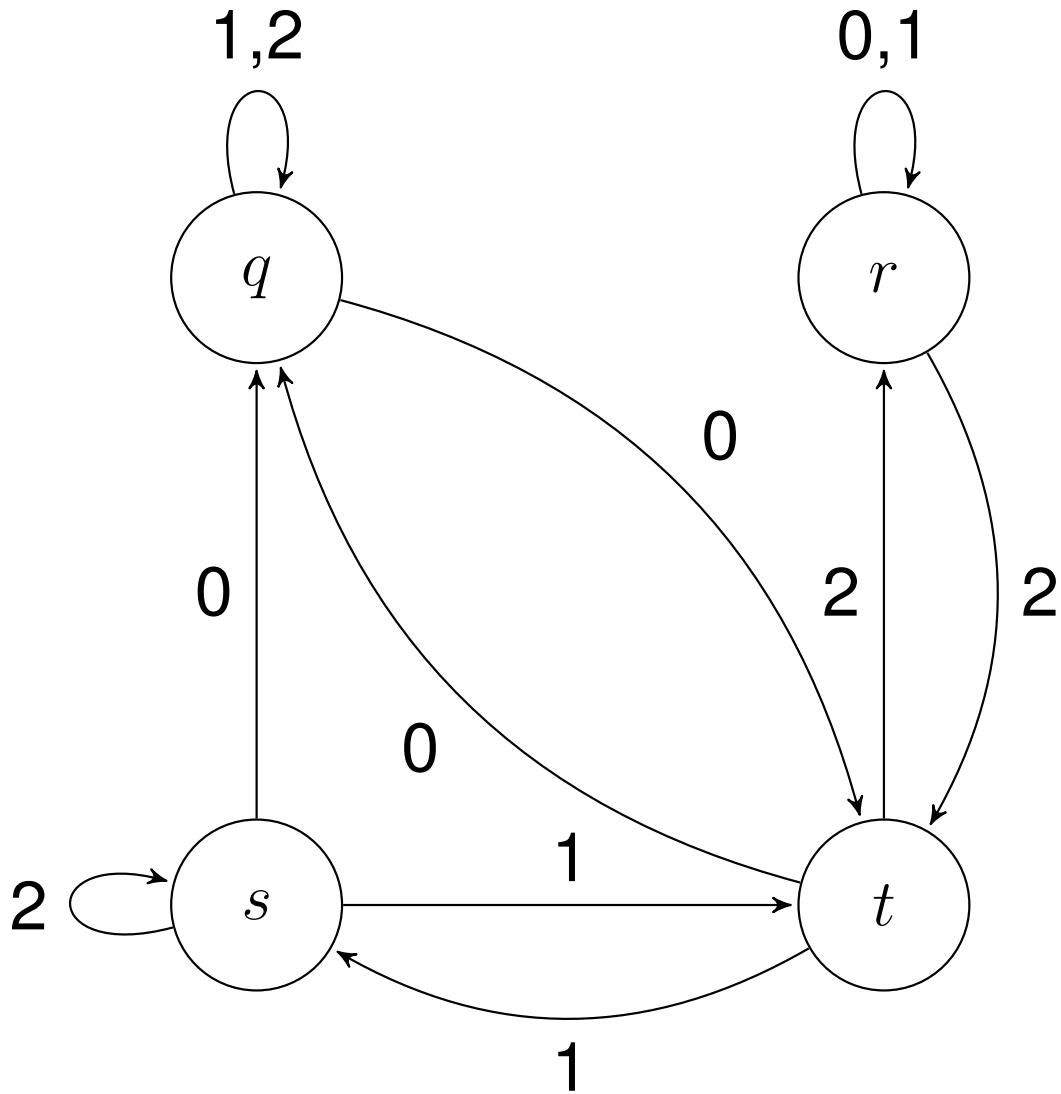
Example for 3 States, Length 4



The word $\mathbf{0110}$ is a synchronising word which sends all states to $\mathbf{r}$.

Example for 4 States, Length 9

Alphabet $\{0, 1, 2\}$ and synchronising word 012020120.



Upper Bounds

Černý's lower bound is quadratic, all known upper bounds are cubic or worse.

Theorem [Frankl 1982; Klyachko, Rystov and Spivac 1987; Pin 1983]. Assume a complete dfa has $\mathbf{n}$ states and has a synchronising word. Then it has a synchronising word not longer than $(\mathbf{n^3 - n})/\mathbf{6}$.

Example

If $\mathbf{n = 3}$ and the dfa has a synchronising word then there is a synchronising word of length up to $\mathbf{4}$.

Černý's Conjecture

If a dfa of $\mathbf{n}$ states has a synchronising word then it has one not longer than $\mathbf{(n-1)^2}$.

Exercise 12.13

Prove Černý's conjecture for $\mathbf{n = 4}$.

Star Height

Consider regular expressions formed by union, concatenation and Kleene star and count how often stars are nested:

- Let $\mathbf{S_0}$ contain all finite languages, note that $\mathbf{S_0}$ is closed under union and concatenation;
- For each $\mathbf{n}$, let $\mathbf{S_{n+1}}$ contain all languages which can be formed by taking unions and concatenations of languages of the form $\mathbf{L}$ or $\mathbf{L^*}$ with $\mathbf{L \in S_n}$.

The star-height of a regular language $\mathbf{L}$ is the minimal $\mathbf{n}$ with $\mathbf{L \in S_n}$.

Theorem

For each $\mathbf{n}$ there is a regular language of star height $\mathbf{n}$.

Examples

Here some examples for the lowest levels.

- The language $\mathbf{L_0} = \{0, 11, 222, 3333\}$ is finite and has star-height 0;
- The language $\mathbf{L_1} = \{00, 11\}^*$ has star-height 1;
- The language $\mathbf{L_2} = (\{00, 11\}^* \cdot \{22, 33\} \cdot \{00, 11\}^* \cdot \{22, 33\})^*$ has star-height 2.

Generalised Star-Height

Let $\mathbf{T_0}$ contain the all sets formed from $\mathbf{\Sigma^*}$ and all finite languages by iterated use of intersection, union, concatenation and set-difference. Let $\mathbf{T_{n+1}}$ be all sets fromed from languages $\mathbf{L}$ and $\mathbf{L^*}$ with $\mathbf{L \in T_n}$ by iterated use of intersection, union, concatenation and set-difference. The languages $\mathbf{L \in T_n}$ which are not in any $\mathbf{T_m}$ with $\mathbf{m < n}$ have generalised star-height $\mathbf{n}$.

Open Problem

For which levels $\mathbf{n > 1}$ do there exist regular languages of generalised star-height $\mathbf{n}$?

Examples

- The language $\{0,1\}^*$ has generalised star-height 0, as

$$\{0,1\}^* = \Sigma^* - \bigcup_{a \in \Sigma - \{0,1\}} \Sigma^* a \Sigma^*;$$

- L_2 from above has generalised star-height 1, as
  $L_2 = \{00,11,22,33\}^* \cap$
  $(\{0,1\}^* \cdot \{22,33\} \cdot \{0,1\}^* \cdot \{22,33\})^*$
  and so L_2 is the intersection of two languages of generalised star-height 1;
- $L_3 = \{w : w$ does not have a substring of the form $v\}$ for a fixed v is of generalised star-height 0 as $L_3 = \Sigma^* - \Sigma^* \cdot v \cdot \Sigma^*$;
- $L_4 = \{w : w$ has an even number of $0\}$ is of generalised star-height 1.

Quiz

Determine the generalised star-height of the following languages over the alphabet $\{0, 1, 2\}$ – it is zero or one:

1. $\{00, 11, 22\}^* \cdot \{000, 111, 222\}^*$;
2. $\{0, 1\}^* \cdot 2 \cdot \{0, 1\}^* \cdot 2 \cdot \{0, 1\}^*$;
3. $(\{0, 1\}^* \cdot 2 \cdot \{0, 1\}^* \cdot 2 \cdot \{0, 1\}^*)^*$;
4. $(\{0, 1\}^* \cdot 2 \cdot \{0, 1\}^*)^*$;
5. $(\{0, 1\}^+ \cdot 22)^*$;
6. $(\{0, 1\}^* \cdot 22)^*$;
7. $(((00)^+ \cdot 11)^+ \cdot 22)^+$.

Recognising Languages

Deciding Membership

Let $\mathbf{L}$ be a language and let $\mathbf{n} = |\mathbf{w}|$ be the length of an input word $\mathbf{w}$. For various types of $\mathbf{L}$ there are methods to decide whether $\mathbf{w} \in \mathbf{L}$.

If $\mathbf{L}$ is regular then a dfa can check in linear time whether a word $\mathbf{w} \in \mathbf{L}$.

If $\mathbf{L}$ is context-free then one can check in time $\mathbf{O(n^3)}$ whether $\mathbf{w} \in \mathbf{L}$; this algorithm of Cocke, Kasami and Younger has been improved to $\mathbf{O(n^{2.38})}$ by Coppersmith and Winograd.

If $\mathbf{L}$ is context-sensitive then one can check in time $\mathbf{O(c^n)}$ for some $\mathbf{c}$ whether $\mathbf{w} \in \mathbf{L}$. Furthermore, there are non-deterministic algorithms running in space $\mathbf{O(n)}$ and deterministic algorithms running in space $\mathbf{O(n^2)}$.

Questions

Open Problem

What is the best time complexity to decide the membership of a context-free language?

Open Problem

Can the membership in a given context-sensitive language be decided in deterministic linear space?

This is related to the overall problem whether a non-determinisitc algorithm using space $\mathbf{s(n)}$ can be made into a deterministic algorithm using space $\mathbf{O(s(n))}$; the best bound known is $\mathbf{O(s^2(n))}$.

Open Problem

Given an automatic group $\mathbf{G}$ with a finite generator set $\mathbf{F}$, let $\mathbf{L} = \{\mathbf{w} \in \mathbf{F}^* : \mathbf{w}$ and ε represent the same member of $\mathbf{G}\}$. Is $\mathbf{L}$ solvable in logarithmic space?

Isomorphism Problems

In certain cases one does not ask how difficult a problem is but only whether it has at all an algorithmic solution. One can, for example, for automatic well-orderings $(\mathbf{A}, <)$ and $(\mathbf{B}, <)$ decide with an algorithm whether they are isomorphic; similarly for automatic linear orders. One cannot do this for automatic equivalence relations.

Open Problem

- Assume that $(\mathbf{A}, \mathbf{Succ_A}, \mathbf{P_A})$ and $(\mathbf{B}, \mathbf{Succ_B}, \mathbf{P_B})$ are automatic structures such that $(\mathbf{A}, \mathbf{Succ_A})$ and $(\mathbf{B}, \mathbf{Succ_B})$ are isomorphic to the natural numbers with successor and that $\mathbf{P_A}$ and $\mathbf{P_B}$ are regular predicates (subsets) on $\mathbf{A}$ and $\mathbf{B}$. Can one decide whether $(\mathbf{A}, \mathbf{Succ_A}, \mathbf{P_A})$ and $(\mathbf{B}, \mathbf{Succ_B}, \mathbf{P_B})$ are isomorphic?
- Can one decide whether the commutative fully automatic groups $(\mathbf{A}, +)$ and $(\mathbf{B}, +)$ are isomorphic?

Complexity of Deciding Games

One studies the complexity of deciding games (or other problems) in dependence of parameters. For parity games, the parameters taken here are the number $\mathbf{n}$ of nodes and the number $\mathbf{m}$ of values.

- McNaughton 1993: $\mathbf{O((kn)^{m+1})}$ for some $\mathbf{k}$.
- Browne, Clarke, Jah, Long and Marrero 1997: $\mathbf{O(n^2 \cdot (2n/m)^{(m+3)/2})}$.
- Jurdinski, Patterson and Zwick 2006/2008: $\mathbf{n^{k \cdot \sqrt{n}}}$ for some $\mathbf{k}$.
- Schewe 2007/2016: $\mathbf{n^2 \cdot (k \cdot n \cdot m^{-2})^{m/3}}$ for some $\mathbf{k}$.
- Calude, Jain, Khoussainov, Li and Stephan 2016: $\mathbf{O((m/\log(n))^4 \cdot n^{3.45+\log(m/\log(n)+3)})}$.

Open Problem: Can parity games be decided in polynomial time?

Winning Statistics 1

Modify parity game to parity game with winning statistics. Winning statistics are vectors $\mathbf{b_0\, b_1\, \ldots, b_k}$ with $\mathbf{k = \lceil \log(n) + 2 \rceil}$ with the following meaning: $\mathbf{b_i > 0}$ stands for the observation of $\mathbf{2^i}$ nodes in the play so far such that between any of these nodes the highest value was of the player's parity and $\mathbf{b_i}$ is the largest value observed with starting the end of the sequence. If $\mathbf{b_i, b_j > 0}$ and $\mathbf{j < i}$ then the sequence for $\mathbf{b_j}$ can only start after the $\mathbf{2^i}$ nodes of the previous sequence and also the node with value $\mathbf{b_i}$ have been observed.

Winning statistics indicate whether a player will eventually win in the case that the winner plays a memoryless winning strategy.

Winning Statistics 2

Initialisation: All $\mathbf{b_i}$ of a winning statistics are $\mathbf{0}$.

Update rule: For each new node with value $\mathbf{b}$, choose the largest $\mathbf{i}$ which can do one of the following:

- $\mathbf{b}$ and $\mathbf{b_0, b_1, \ldots, b_{i-1}}$ have the players parity and $\mathbf{b_i}$ does not;
- $\mathbf{0 < b_i < b}$.

If found, let $\mathbf{b_i = b}$ and all $\mathbf{b_j = 0}$ for all $\mathbf{j < i}$.

Theorem

If Anke plays a memoryless winning strategy her winning statistics will eventually have $\mathbf{b_k > 0}$ what will indicate that she has a winning strategy while Boris' winning statistics will never indicate a win. Similarly for Boris and his winning statistics.

Reduction to Survival Games

Game graphs of survival game consist of all nodes $(\mathbf{v}, \mathbf{p}, \mathbf{w})$ with node $\mathbf{v}$ of parity game, player $\mathbf{p}$ to move and current winning statistics $\mathbf{w}$ of Boris. Move from $(\mathbf{v}, \mathbf{p}, \mathbf{w})$ to $(\mathbf{v}', \mathbf{p}', \mathbf{w}')$ is possible iff there is edge from $\mathbf{v}$ to $\mathbf{v}'$ in parity game, $\mathbf{p} \neq \mathbf{p}'$ and on move with value of $\mathbf{v}'$, winning statistics $\mathbf{w}$ of Boris are updated to $\mathbf{w}'$ and $\mathbf{w}$ is not already won for Boris. Anke loses if game gets stuck.

There is an algorithm to determine the winner of a survival game which needs time linear in number of edges of the survival game, so linear in time $\mathbf{n^2 \cdot n^{\log(m)+4}}$ where $\mathbf{n^{\log(m)+4}}$ is an upper bound on the number of winning statistics for Boris. This upper bound is based on the fact that Winning statistics consist of $\lceil \log(\mathbf{n}) + \mathbf{3} \rceil$ numbers with $\lceil \log(\mathbf{m} + \mathbf{1}) \rceil$ bits.

More careful counting gives slightly better bound.

Exercises 12.25-12.28

Let $\mathbf{A} = \{0\}^* \cdot \{1\}^*$, $\mathbf{B} = \{00\}^* \cdot \{11\}^* \cdot \{22\}^*$; $\mathbf{C} = \{00, 11, 22\}^*$; $\mathbf{D} = \{0, 1\}^* - \{1\}^*$.

Exercise 12.25

How are the above sets $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ ordered by $\leq_{au}$? Provide the reductions where they exist.

Exercise 12.26

How are the above sets $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ ordered by $\leq_{tr}$? Provide the reductions where they exist.

Exercise 12.27

Find for $\mathbf{A}, \mathbf{B}, \mathbf{C}$ regular sets $\mathbf{A}', \mathbf{B}', \mathbf{C}'$ such that for all $\mathbf{x}$, $\mathbf{x} \in \mathbf{A} \Leftrightarrow \mathbf{xx} \in \mathbf{A}'$ and $\mathbf{x} \in \mathbf{B} \Leftrightarrow \mathbf{xx} \in \mathbf{B}'$ and $\mathbf{x} \in \mathbf{C} \Leftrightarrow \mathbf{xx} \in \mathbf{C}'$.

Exercise 12.28

Provide a regular set $\mathbf{E}$ such that $\mathbf{E}'$ exists as in 12.27, but $\mathbf{E}'$ is neither $\mathbf{E}$ nor $\mathbf{E} \cdot \mathbf{E}$.

Exercises 12.29-12.31

Determine for the following languages $\mathbf{F}, \mathbf{G}, \mathbf{H}$ the minimal complete dfa and either determine its smallest synchronising word or show that there is no synchronising word.

Exercise 12.29

Do the above for $\mathbf{F} = (\{\mathbf{0}\}^* \cdot \{\mathbf{1}\} \cdot \{\mathbf{0}\}^* \cdot \{\mathbf{1}\})^*$.

Exercise 12.30

Do the above for $\mathbf{G} = (\{\mathbf{0}\}^+ \cdot \{\mathbf{1}\} \cdot \{\mathbf{0}\}^+ \cdot \{\mathbf{1}\})^*$.

Exercise 12.31

Do the above for $\mathbf{H} = (\{\mathbf{0}\}^+ \cdot \{\mathbf{1}\} \cdot \{\mathbf{0}\}^+ \cdot \{\mathbf{1}\})^* \cup (\{\mathbf{0},\mathbf{1}\}^* \cdot \{\mathbf{11}\} \cdot \{\mathbf{0},\mathbf{1}\}^* \cdot \{\mathbf{00}\}) \cup (\{\mathbf{1}\} \cdot \{\mathbf{0},\mathbf{1}\}^* \cdot \{\mathbf{00}\})$.

Final Examination

- Final examination counts 60 marks; midterm counts 30 marks and homework counts 10 marks.
- The final examination covers all 12 lectures.
- There are 10 questions with six marks each equally distributed over the material.
- Some material in the lecture notes was not covered or only sketched during lecture hours; this is additional add-on for those who are interested.
- When revising the material, look also at the exercises in order to test whether you understood the material and also check out the self-tests.
- There will be no questions covering topics which are not in the lecture notes.
- The final examination is a closed book assessment.