Gymnasium Bäumlihof

Maturaarbeit

Theoretical Informatics: Formal languages and finite model thoery

A study of the connection of first order logic and context-sensitive languages

Written by:
Yaël Arn, 4A

Supervisor:
Aline Sprunger

Coreferent:
Bernhard Pfammatter

June 27, 2024, 4058 Basel

Contents

1 Introduction **3**

2 Descriptive Complexity **4**

2.1 Aims 4

2.2 Important Results 4

2.2.1 $\text{NSPACE}[s(n)] \subseteq \text{DSPACE}[s(n)^2]$ 4

2.2.2 $\text{FO(LFP)} = \text{P}$ 4

2.3 Open questions 4

2.3.1 $\text{P} \stackrel{?}{=} \text{NP}$ 4

2.3.2 $\text{NSPACE}[O(n)] \stackrel{?}{=} \text{DSPACE}[O(n)]$ 4

3 Formal Languages **5**

3.1 Definition 5

3.2 Chomsky Hierarchy 5

3.2.1 Regular Languages 5

3.2.2 Context-Free Languages 5

3.2.3 Context-Sensitive Languages 5

3.2.4 Recursive Languages 5

4 Personal Contribution **6**

5 Results **7**

6 Conclusion and Direction **8**

Index **10**

List of Figures **10**

Listings **11**

Bibliography **12**

A Mathematical Background **13**

A.1 Set Theory 13

A.2 First Order Logic 13

A.3 Second Order Logic 14

A.4 Turing Machines 15

Forword

1. Introduction

2. Descriptive Complexity

2.1 Aims

2.2 Important Results

2.2.1 $\mathbf{NSPACE}[s(n)] \subseteq \mathbf{DSPACE}[s(n)^2]$

2.2.2 $\mathbf{FO(LFP)} = \mathbf{P}$

2.3 Open questions

2.3.1 $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$

2.3.2 $\mathbf{NSPACE}[O(n)] \stackrel{?}{=} \mathbf{DSPACE}[O(n)]$

3. Formal Languages

3.1 Definition

3.2 Chomsky Hierarchy

3.2.1 Regular Languages

3.2.2 Context-Free Languages

3.2.3 Context-Sensitive Languages

3.2.4 Recursive Languages

4. Personal Contribution

5. Results

6. Conclusion and Direction

Thanks

List of Figures

Listings

Bibliography

[HR22] Malte Helmert and Gabriele Röger. *Lecture: Discrete Mathematics in Computer Science.* University of Basel. 2022. URL: `https://dmi.unibas.ch/en/studies/computer-science/courses-in-fall-semester-2022/lecture-discrete-mathematics-in-computer-science/` (visited on 06/26/2024).

[Imm99] Neil Immerman. *Descriptive Complexity.* Springer New York, 1999. ISBN: 9781461205395. DOI: `10.1007/978-1-4612-0539-5`.

[Rög23] Gabriele Röger. *Lecture: Theory of Computer Science.* Universität Basel. 2023. URL: `https://dmi.unibas.ch/de/studium/computer-science-informatik/lehrangebot-fs23/main-lecture-theory-of-computer-science-1/` (visited on 06/26/2024).

A. Mathematical Background

The definitions are taken from the lectures Discrete Mathematics in Computer Science [HR22] and Theory of Computer Science [Rög23] and also from the book Descriptive Complexity [Imm99].

A.1 Set Theory

Set An unordered collection of distinct elements, written with curly braces $\{\}$

Tuple An ordered collection of elements written with pointed braces $\langle\rangle$

Set operations There are multiple ways to form new sets from already existing sets:

- **Union** denoted as $\cup$, an element is in $A \cup B$ if and only if it is in A or B
- **Intersection** denoted as $\cap$, an element is in $A \cap B$ if and only if it is in A and B
- **Cartesian product** denoted as $\times$, $A \times B$ is the set of tuples with an element of A and an element of B
- **Cartesian power** A^k denotes the cartesian product of A with itself repeated k times

A.2 First Order Logic

We abbreviate first order logic as FO.

Variable A variable is an element that can have a value from a set.

Universe The set over which variables and constants can range

Relation A relation of arity k, $R(x_1, \dots, x_k)$ can be either true or false for any k-tuple of variables. In this work we always consider equality($=$), an ordering relation $\leq$, and $BIT(x, y)$, which means that the y^{th} bit of x is set in binary notation, to exist.

Vocabulary A tuple $\tau = \langle R_1^{a_1}, \dots, R_r^{a_r}, c_1, \dots, c_s\rangle$ of relations R_i with arity a_i and constants c_j (We omit functions as they can be simulated by a relation in our case)

Structure A tuple $\mathcal{A} = \langle |\mathcal{A}|, R_1^{\mathcal{A}}, \dots, R_r^{\mathcal{A}}, c_1^{\mathcal{A}}, \dots, c_s^{\mathcal{A}}\rangle$ where $|\mathcal{A}|$ is the universe, the constants are assigned a value from $|\mathcal{A}|$ and the truth of the relations have a truth value for each a_i-tuple from $|\mathcal{A}|^{a_i}$

First Order Formula A first order formula is inductively defined as follows:

Atoms Any formula of the form $R(x_1, \ldots, x_k)$ for some relation of arity k is called an atomic formula

conjuction If φ and ψ are formulas, $(\varphi \wedge \psi)$ is a formula

disjuction If φ and ψ are formulas, $(\varphi \vee \psi)$ is a formula

negation If φ is a formula, $\neg\varphi$ is a formula

Existencial Quantification If φ is a formula, $\exists x\varphi$ is a formula

Universal Quantification If φ is a formula, $\forall x\varphi$ is a formula

Semantics For any structure, we can assign a truth value to any formula (by assigning values from the universe to free variables if they exist in the formula). We say $\mathcal{A}$ satisfies ϕ (where ϕ is taken over the vocabulary of $\mathcal{A}$), denoted $\mathcal{A} \models \phi$ if and only if ϕ is true under the interpretation of the constant and relations of $\mathcal{A}$. This is inductively defined as follow:

Atoms For a formula ϕ of the form $R(x_1, \ldots, x_k)$, we have $\mathcal{A} \models \phi$ if and only if the interpretation of the relation maps $\langle x_1, \ldots, x_k \rangle$ to true

conjuction We have $\mathcal{A} \models (\varphi \wedge \psi)$ if and only if $\mathcal{A} \models \varphi$ and $\mathcal{A} \models \psi$

disjuction We have $\mathcal{A} \models (\varphi \vee \psi)$ if and only if $\mathcal{A} \models \varphi$ or $\mathcal{A} \models \psi$

negation We have $\mathcal{A} \models \neg\varphi$ if and only if $\mathcal{A} \not\models \varphi$

Existencial Quantification We have $\mathcal{A} \models \exists x\varphi$ if and only if there exists a $y \in |\mathcal{A}|$ such that $\mathcal{A} \models \varphi(y)$ (where $\varphi(y)$ denotes φ with any occurrence of x replaced with the element y)

Universal Quantification We have $\mathcal{A} \models \forall x\varphi$ if and only if for all $y \in |\mathcal{A}|$ we have $\mathcal{A} \models \varphi(y)$

A.3 Second Order Logic

In second order logic, we extend the capabilities of first order logic with the ability to quantify over relations. We thus also need to extend our definitions. We abreviate second order logic as SO.

SO variables A relation that is not given in the vocabulary and can be substituted with a specific interpretation

SO formula In addition to the inductive rules from the FO formulas, we can quantify over second order formulas

SO Existencial Quantification If φ is a formula, then $\exists V\varphi$ is a formula

SO Universal Quantification If φ is a formula, then $\forall V\varphi$ is a formula

SO Semantics Here we also need to extend the FO semantics

SO Existencial Quantification We have $\mathcal{A} \models \exists V\varphi$ if and only if there exists a relation U over $|\mathcal{A}|$ such that $\mathcal{A} \models \varphi(U)$ (where $\varphi(U)$ denotes φ with any occurrence of V replaced with U)

SO Universal Quantification We have $\mathcal{A} \models \forall V\varphi$ if and only if for all relations U over $|\mathcal{A}|$ we have $\mathcal{A} \models \varphi(U)$

A.4 Turing Machines

Turing machines are the most common model of computation. We abreviate Turing Machines as TM

Informal definition A turing machine is a automaton with a finite number of states and an infinite tape. Using a read/write head, which can read one symbol on the tape, modify one symbol on the tape and move left and right, a Turing Machine can compute functions

Formal definition Formally, a Turing machine is a 7-tuple $M = \langle Q, \sum, \Gamma, \delta, q_0, q_{accept}, q_{reject} \rangle$, where

Q is the set of states

$\sum$ is the set representing the symbols of which the input word on the tape can consist

Γ is the set of symbols which can be written or read on the tape

δ is the transition function, with $\delta : \Gamma \times Q \to \Gamma \times Q \times \{L, R\}$. So when a TM is in state n and reads a on the tape, δ tells us to which state we should transition, which symbol we should write and which direction we should move the read/write head

q_0 the start state

q_{accept} the accept state

q_{reject} the reject state

Turing computation At the beginning, the TM is in the start state, the input is written in a consecutive way on the tape and the read/write head is on the first character of the input word. In consecutive steps, the machine state then changes according to the transition function. If at some point the machine enters the accept or the reject state, the computation halts, and the TM is said to have accepted / rejected the input. In this work we will ignore the tape content after the computation and focus on decision problems.

Decidability If a TM halts on all inputs, we say that it decides a problem, as we can always be sure that the machine will accept or reject an input in finite time.

Nondetermenistic TM (NTM) We can extend the transition function δ to allow multiple transitions from a given state. If there exists any computational path which leads to an accept state, the NTM accepts. This is not analog to how real sequential computers work, but allows interesting results, and is as powerfull as a normal deterministic TM.

Church-Turing Thesis According to the Church-Turing Thesis, this formalism is equivalent to what any computer can compute.

Independence declaration (German)

Ich, Yaël Arn, 4A

bestätige mit meiner Unterschrift, dass die eingereichte Arbeit selbstständig und ohne unerlaubte Hilfe Dritter verfasst wurde. Die Auseinandersetzung mit dem Thema erfolgte ausschliesslich durch meine persönliche Arbeit und Recherche. Es wurden keine unerlaubten Hilfsmittel benutzt. Ich bestätige, dass ich sämtliche verwendeten Quellen sowie Informanten/-innen im Quellenverzeichnis bzw. an anderer dafür vorgesehener Stelle vollständig aufgeführt habe. Alle Zitate und Paraphrasen (indirekte Zitate) wurden gekennzeichnet und belegt. Sofern ich Informationen von einem KI-System wie bspw. ChatGPT verwendet habe, habe ich diese in meiner Maturaarbeit gemäss den Vorgaben im Leitfaden zur Maturaarbeit korrekt als solche gekennzeichnet, einschliesslich der Art und Weise, wie und mit welchen Fragen die KI verwendet wurde. Ich bestätige, dass das ausgedruckte Exemplar der Maturaarbeit identisch mit der digitalen Version ist. Ich bin mir bewusst, dass die ganze Arbeit oder Teile davon mittels geeigneter Software zur Erkennung von Plagiaten oder KI-Textstellen einer Kontrolle unterzogen werden können.

Ort & Datum　　　　　　Unterschrift