Introduction to Descriptive Complexity Theory

Matteo Cavallaro

Università degli Studi di Udine

date

Descriptive complexity theory

- a branch of **computational complexity theory** and **finite model theory**
- characterizes *complexity classes* by the type of *logic* needed to express the languages in them.

Main results by Fagin, Immerman, Vardi, and others.

Definition 9.5

Let $\mathcal{K}$ be a complexity class, $\mathcal{L}$ a logic, and $\mathcal{C}$ a class of finite structures. $\mathcal{L}$ captures $\mathcal{K}$ on $\mathcal{C}$ if

- $\left(\mathfrak{A} \overset{?}{\models} \varphi\right) \in \mathcal{K}$, with φ sentence in $\mathcal{L}$, $\mathfrak{A} \in \mathcal{C}$,
- for any $\mathcal{P} \subseteq \mathcal{C}$ s.t. $\left(\mathfrak{A} \overset{?}{\in} \mathcal{P}\right) \in \mathcal{K}$, exists $\varphi_{\mathcal{P}}$ sentence in $\mathcal{L}$ s.t.

$$\mathfrak{A} \models \varphi_{\mathcal{P}} \iff \mathfrak{A} \in \mathcal{P}$$

Why?

- machine-independent characterization of complexity classes;
- proving separation of complexity classes (es. $\mathbf{P} \overset{?}{=} \mathbf{NP}$).

$\mathcal{L}_1$ captures $\mathcal{K}_1$, $\mathcal{L}_2$ captures $\mathcal{K}_2$, $\quad \mathcal{L}_1 = \mathcal{L}_2 \iff \mathcal{K}_1 = \mathcal{K}_2$.

Definition 7.3

Existential SO logic, or $\exists\mathrm{SO}$, is defined as the restriction of SO that consists of the formulae of the form $\exists X_1 \ldots \exists X_n\ \varphi$, with $\varphi \in \mathrm{FO}$.

Analogous definition for universal SO logic, or $\forall\mathrm{SO}$.
A more intuitive notation could be $\exists^{\mathrm{SO}}\mathrm{FO}$ and $\forall^{\mathrm{SO}}\mathrm{FO}$.

Theorem 9.6 (Fagin)

$\exists\mathrm{SO}$ captures **NP**.

Corollary 9.7

$\forall\mathrm{SO}$ captures **coNP**.

Steps of the proof.

1. $\varphi \in \exists\mathrm{SO} \implies \left(\mathfrak{A} \stackrel{?}{\models} \varphi\right) \in \mathbf{NP}$
2. for any $\mathcal{P}$ s.t. $\left(\mathfrak{A} \stackrel{?}{\in} \mathcal{P}\right) \in \mathbf{NP}$, exists $\varphi_{\mathcal{P}} \in \exists\mathrm{SO}$ s.t.

$$\mathfrak{A} \models \varphi_{\mathcal{P}} \iff \mathfrak{A} \in \mathcal{P}$$

Proof.

1. Let $\varphi_{\mathcal{P}} = \exists S_1 \ldots \exists S_n\ \psi$, with $\psi \in \mathrm{FO}$. Given $\mathfrak{A}$, an NTM can guess non deterministically $S_1, \ldots, S_n$, then decides if $\mathfrak{A} \models \psi(S_1, \ldots, S_n)$ in polynomial time in $||\mathfrak{A}||$ plus the size of $S_1, \ldots, S_n$.

2. Let $\mathcal{P}$ be a property of σ-structures that can be tested by a polynomial NTM $M = (Q, \Sigma, \Delta, \delta, q_0, Q_a, Q_r)$ that has a one-way infinite tape and runs in n^k.
   Let $\sigma' = \sigma \cup \{\leq^2\} \cup \left\{(T_a)^{2k} \mid a \in \Delta\right\} \cup \left\{(H_q)^{2k} \mid q \in Q\right\}$.
   Intended meaning:
   - $\leq$ is a linear order on the universe.
   - $\leq_k$ is the lexicographic linear order on k-tuples (defined on $\leq$).
   - Positions on the tape ($\vec{p}$) and time ($\vec{t}$) by k-tuples of the elements of the universe.
   - $\{T_a\}$ are *tape* predicates: $T_a(\vec{p}, \vec{t})$ indicates that position $\vec{p}$ at time $\vec{t}$ contains a.
   - $\{H_q\}$ are *head* predicates: $H_q(\vec{p}, \vec{t})$ indicates that at time $\vec{t}$, the machine is in state q, and its head is in position $\vec{p}$.

Let $\varphi_{\mathcal{P}} = \exists L \exists T_{a_1} \ldots \exists T_{a_n} \exists H_{q_0} \ldots H_{q_{m-1}} \psi$ sentence over σ', with $\psi \in \mathrm{FO}$. ψ the conjunction of the sentences expressing the following properties:

- L is a linear order ($\leq$);
- in every configuration of M, each cell of the tape contains exactly one element of Δ;
- at any time the machine is in exactly one state;
- eventually, M enters a state $q \in Q_a$;
- $\{T_a\}$ and $\{H_q\}$ relations respect the transition of M:

$$\forall a \in \Delta \colon \forall q \in Q \colon \bigvee_{(q',b,m) \in \delta(q,a)} \alpha(q, a, q', b, m),$$

  where $m \in \{l, -, r\}$, and $\alpha(q, a, q', b, m)$ is the sentence describing the transition in which upon reading a in state q, the machine writes b, makes the move m, and enters state q'.
- At the beginning of the computation, the input tape contains $enc(\mathfrak{A})$. Suppose $\iota(\vec{p})$ and $\xi(\vec{p})$ are formulas such that:
  - $\mathfrak{A} \models \iota(\vec{p})$ iff the $\vec{p}$th position of $enc(\mathfrak{A})$ is 1;

- $\mathfrak{A} \models \xi(\vec{p})$ iff $\vec{p}$ exceeds the length of $enc(\mathfrak{A})$.

$$\forall \vec{p} \forall \vec{t} \left(\neg \exists \vec{u} \, (\vec{u} \leq_k \vec{t}) \rightarrow \begin{array}{c} (\iota(\vec{p}) \leftrightarrow T_1(\vec{t}, \vec{p})) \\ \wedge \, (\xi(\vec{p}) \leftrightarrow T_\#(\vec{t}, \vec{p})) \end{array} \right)$$

$\square$

Solving $\exists\mathrm{SO} \stackrel{?}{=} \forall\mathrm{SO} \implies$ interesting results:

- $\exists\mathrm{SO} \neq \forall\mathrm{SO} \implies \mathbf{NP} \neq \mathbf{coNP} \implies \mathbf{P} \neq \mathbf{NP}$
- $\exists\mathrm{SO} = \forall\mathrm{SO} \implies \mathbf{NP} = \mathbf{coNP} \implies \Sigma_n\mathbf{P} = \Pi_n\mathbf{P} = \Delta_n\mathbf{P} = \mathbf{NP}$
  The polynomial hierarchy collapses to the first level!
  (see Papadimitriou 1994, Theorem 17.9)

How to prove $\exists\mathrm{SO} \neq \forall\mathrm{SO}$? Find a property $\mathcal{P}$ expressible in $\exists\mathrm{SO}$ but not in $\forall\mathrm{SO}$ (or vice versa).
Some related results:

- on some infinite structures (e.g., $\langle\mathbb{N}, +, \cdot\rangle$) is known that $\exists\mathrm{SO} \neq \forall\mathrm{SO}$;
- $\exists\mathrm{MSO} \neq \forall\mathrm{MSO}$.

Definition 7.3

Existential monadic SO logic, or $\exists\mathrm{MSO}$, and universal monadic SO logic, or $\forall\mathrm{MSO}$, are defined as the restrictions of $\exists\mathrm{SO}$ and $\forall\mathrm{SO}$ respectively where all second-order quantifiers have arity 1.

Proposition 7.14 (Fagin, Stockmeyer, and M. Vardi 1995)

Graph connectivity is expressible in $\forall\mathrm{MSO}$, but is not expressible in $\exists\mathrm{MSO}$.

Hence, $\exists\mathrm{MSO} \neq \forall\mathrm{MSO}$.

Fagin's theorem can be extended to

- other fragments of **second order logic**
- and other computational classes in the **polynomial hierarchy**.

Which classes? $\Sigma_k^1, \Pi_k^1 \subseteq \mathrm{SO}$ and $\Sigma_k^{\mathbf{P}}, \Pi_k^{\mathbf{P}} \subseteq \mathbf{PH}$.

Σ^1_k and Π^1_k, fragments of SO

Remark

$\exists\mathrm{SO} = \{(\exists \ldots \exists)\,(\varphi) \mid \varphi \in \mathrm{FO}\}, \quad \forall\mathrm{SO} = \{(\forall \ldots \forall)\,(\varphi) \mid \varphi \in \mathrm{FO}\}.$

Definition (Väänänen 2020, Section 4)

Σ^1_k and Π^1_k are defined as the restrictions of SO that consist of the formulae of the forms

$\underbrace{(\exists \ldots \exists)(\forall \ldots \forall)(\exists \ldots \exists) \ldots}_{k}\,(\varphi)$ and $\underbrace{(\forall \ldots \forall)(\exists \ldots \exists)(\forall \ldots \forall) \ldots}_{k}\,(\varphi)$

respectively, with $\varphi \in \mathrm{FO}$. Also, $\Delta^1_k = \Sigma^1_k \cup \Pi^1_k$.

Examples

$\mathrm{FO} = \Sigma^1_0 = \Pi^1_0 = \Delta^1_0, \quad \exists\mathrm{SO} = \Sigma^1_1, \quad \forall\mathrm{SO} = \Pi^1_1.$

$\Sigma_k^{\mathbf{P}}$ and $\Pi_k^{\mathbf{P}}$, computational classes in **PH**

Definition 14.3 (Papadimitriou 1994)

A **Turing machine** $M^?$ **with oracle** is a machine that includes a *query string*, and three special states: $q_?$, the *query state*, and q_{YES}, q_{NO}, the *answer states*.
The computation of oracle machine $M^?$ with oracle $A \subseteq \Sigma^*$ proceeds ordinarily, except that the machine moves from the query state to one of the answer states depending on whether the current query string is in A or not.

$\mathcal{C}^{\mathcal{C}'}$ the class of all languages decided by machines having the same resources bounds as in $\mathcal{C}$, using as oracle a language of $\mathcal{C}'$.

Definition 17.2 (Papadimitriou 1994)

$\Delta_0^{\mathbf{P}} = \Sigma_0^{\mathbf{P}} = \Pi_0^{\mathbf{P}} = \mathbf{P}$.
$\Delta_{k+1}^{\mathbf{P}} = \mathbf{P}^{\Sigma_k^{\mathbf{P}}}$, $\Sigma_{k+1}^{\mathbf{P}} = \mathbf{NP}^{\Sigma_k^{\mathbf{P}}}$, $\Pi_{k+1}^{\mathbf{P}} = \mathbf{coNP}^{\Sigma_k^{\mathbf{P}}}$. $\mathbf{PH} = \bigcup_{k \geq 0} \Sigma_k^{\mathbf{P}}$

$$
\begin{tikzcd}
& \vdots & \\
\Sigma^1_3 & & \Pi^1_3 \\
& \Delta^1_2 \arrow[ul] \arrow[ur] & \\
\Sigma^1_2 \arrow[uu] \arrow[ur] & & \Pi^1_2 \arrow[uu] \arrow[ul] \\
& \Delta^1_1 \arrow[ul] \arrow[ur] & \\
\exists \mathrm{SO} = \Sigma^1_1 \arrow[uu] \arrow[ur] & & \Pi^1_1 = \forall \mathrm{SO} \arrow[uu] \arrow[ul] \\
& \Sigma^1_0 = \mathrm{FO} = \Pi^1_0 \arrow[ul] \arrow[ur] &
\end{tikzcd}
\qquad
\begin{tikzcd}
& \vdots & \\
\Sigma^{\mathbf{P}}_3 & & \Pi^{\mathbf{P}}_3 \\
& \Delta^{\mathbf{P}}_3 \arrow[ul] \arrow[ur] & \\
\Sigma^{\mathbf{P}}_2 \arrow[uu] \arrow[ur] & & \Pi^{\mathbf{P}}_2 \arrow[uu] \arrow[ul] \\
& \Delta^{\mathbf{P}}_2 \arrow[ul] \arrow[ur] & \\
\mathbf{NP} = \Sigma^{\mathbf{P}}_1 \arrow[uu] \arrow[ur] & & \Pi^{\mathbf{P}}_1 = \mathbf{coNP} \arrow[uu] \arrow[ul] \\
& \Sigma^{\mathbf{P}}_0 = \mathbf{P} = \Pi^{\mathbf{P}}_0 \arrow[ul] \arrow[ur] &
\end{tikzcd}
$$

Corollary 9.9

For each $k \geq 1$,

- Σ^1_k captures $\Sigma^{\mathbf{P}}_k$, and
- Π^1_k captures $\Pi^{\mathbf{P}}_k$.

In particular, SO captures **PH**.

Proof.

The base case is Fagin's theorem. If we consider a problem in $\Sigma^{\mathbf{P}}_{k+1} = \mathbf{NP}^{\Sigma^{\mathbf{P}}_k}$, there is an $\exists\mathrm{SO}$ sentence φ with additional predicates expressing $\Sigma^{\mathbf{P}}_k$ properties. We know, by inductive hypothesis, that these properties are definable by Σ^1_k. Then pushing the second-order quantifier outwards, we convert φ into a Σ^1_{k+1} sentence. $\square$

Example

$$\exists x\,(P(x) \wedge \neg\exists y\,(xRy)) \quad \longrightarrow \quad \exists x \forall y\,(P(x) \wedge \neg xRy)$$

Note that the corollary does **not** imply FO captures **P**!

Theorem

There are logics that capture **P** over the class of **ordered** structures (M. Y. Vardi 1982, Immerman 1986), such as $\mathrm{Horn}{-}\exists\mathrm{SO}$ (Grädel 1991, Papadimitriou 1994).

Conjecture (Gurevich 1988)

There is no logic that captures **P** over the class of **all** finite structures.

Finding a logic that captures **P** *may* solve $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$.

$$
\begin{array}{ccc}
\exists\mathrm{SO} = \forall\mathrm{SO} & \iff & \mathbf{NP} = \mathbf{coNP} \\
\Downarrow & & \Downarrow \\
\mathrm{SO} = \exists\mathrm{SO} & \iff & \mathbf{PH} = \mathbf{NP}
\end{array}
$$

References I

Commons, Wikimedia (2020). *File:Polynomial time hierarchy.svg — Wikimedia Commons, the free media repository*. URL: https://commons.wikimedia.org/w/index.php?title=File:Polynomial_time_hierarchy.svg&oldid=495206547.

Fagin, R., L.J. Stockmeyer, and M.Y. Vardi (1995). "On Monadic NP vs Monadic co-NP". In: *Information and Computation* 120.1, pp. 78–92. ISSN: 0890-5401. DOI: https://doi.org/10.1006/inco.1995.1100. URL: https://www.sciencedirect.com/science/article/pii/S0890540185711005.

Grädel, Erich (1991). "The expressive power of second order Horn logic". In: *Annual Symposium on Theoretical Aspects of Computer Science*. Springer, pp. 466–477.

Gurevich, Y (1988). *Logic and the challenge of Computer Science. Trends in Theoretical Computer Science ed. EB orger*.

References II

Immerman, Neil (1986). "Relational queries computable in polynomial time". In: *Information and control* 68.1-3, pp. 86–104.

Papadimitriou, C (1994). "Computational Complexity, Addison Welsey". In: *Reading, Massachusetts*.

Väänänen, Jouko (2020). "Second-order and Higher-order Logic". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Fall 2020. Metaphysics Research Lab, Stanford University.

Vardi, Moshe Y (1982). "The complexity of relational query languages". In: *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pp. 137–146.