# Descriptive Complexity

# Space-Based Complexity

**Definition (class DSPACE):** Let $T : \mathbb{N} \to \mathbb{N}$ be some function. A language $L$ is in $\mathbf{DSPACE}(T(n))$ iff there is a Turing machine that uses $O\big(T(n)\big)$ cells of the tape and decides $L$.

**Definition (class NSPACE):** Similarly defined using NDTM.

$\mathbf{PSPACE} = \ \bigcup_{c\geq 1} \mathbf{DSPACE}(n^c)$

**Savitch' Theorem (1970):**

$$\mathbf{PSPACE} = \bigcup_{c\geq 1} \mathbf{NSPACE}(n^c)$$

**Similar Definitions:**

- $\mathbf{DLOGSPACE}$, $\mathbf{NLOGSPACE}$

# Polynomial Hierarchy

**Definition (class PH):** Languages in $\mathbf{PH}$ are those accepted, in polynomial time, by a NTDM can make "calls" to other similar machines.

# The Complexity Zoo

Exponential classes

- $\mathbf{EXPTIME} = \bigcup_{c\geq 1} \mathbf{DTIME}(2^{n^c})$
- $\mathbf{NEXPTIME} = \bigcup_{c\geq 1} \mathbf{NTIME}(2^{n^c})$
- $\mathbf{EXPSPACE} = \bigcup_{c\geq 1} \mathbf{DSPACE}(2^{n^c})$

What do we know?

$$\mathbf{DLOGSPACE} \subseteq \mathbf{NLOGSPACE} \subseteq \mathbf{P} \subseteq \left\{\begin{matrix}\mathbf{NP}\\ \mathbf{coNP}\end{matrix}\right\} \subseteq \mathbf{PH} \subseteq \mathbf{PSPACE} \subseteq \mathbf{EXPTIME} \subseteq \mathbf{NEXPTIME} \subseteq \mathbf{EXPSPACE}$$

Proper subsets?

- All we know is $\mathbf{NLOGSPACE} \subsetneq \mathbf{PSPACE} \subsetneq \mathbf{EXPSPACE}$, $\mathbf{P} \subsetneq \mathbf{EXPTIME}$, $\mathbf{NP} \subsetneq \mathbf{NEXPTIME}$

# Logic vs. Complexity

| Problem | Logical characterization | Complexity |
|---|---|---|
| GCD test | $\exists p, q(p \cdot z = x \wedge q \cdot z = y)$ $\vee \neg\exists z', p', q'(z' > z \wedge p' \cdot z' = x \wedge q' \cdot z' = y)$ | $\mathbf{LOGSPACE}$ |
| Graph Connectivity | $\forall x \forall y \big(x \neq y \rightarrow \mathbf{TC}(E)(x, y)\big)$ | $\mathbf{P}$ |
| Graph Hamiltonicity | $\exists L \exists S \begin{pmatrix} \text{linearOrder}(L) \\ \wedge \text{"}S \text{ is the successor relation of } L\text{"} \\ \wedge \forall x \exists y \big(L(x, y) \vee L(y, x)\big) \\ \wedge \forall x \forall y \big(S(x, y) \rightarrow E(x, y)\big) \end{pmatrix}$ | $\mathbf{NP}$-complete |

# Logic vs. Complexity

**Definition:** A logic $\mathcal{L}$ captures a complexity class $\mathcal{K}$ if:

- The data complexity of $\mathcal{L}$ is $\mathcal{K}$: that is, for every $\mathcal{L}$-sentence $\Phi$, deciding if $\mathcal{A} \vDash \Phi$ for a *finite structure* $\mathcal{A}$ is in $\mathcal{K}$.
- For every property $\mathcal{P}$ that can be decided with complexity $\mathcal{K}$, there is a sentence $\Phi_{\mathcal{P}}$ of $\mathcal{L}$ such that $\mathcal{A} \vDash \Phi_{\mathcal{P}}$ iff $\mathcal{A}$ has the property $\mathcal{P}$, for every finite structure $\mathcal{A}$.

# $\exists SO = \mathbf{NP}$

**Fagin's Theorem (1973):** $\exists SO$ captures $\mathbf{NP}$.

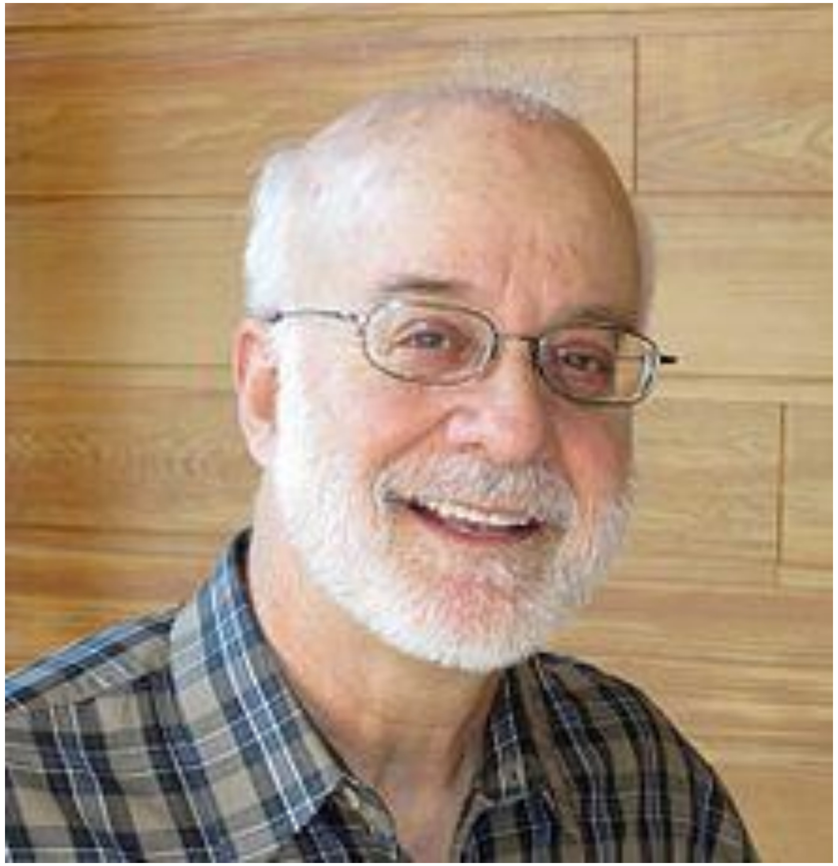
($\exists SO$ consists of SO sentences of the form $\exists X_1 \ldots \exists X_n \, \varphi$ where $\varphi$ is a FO formula with $X_1, \ldots, X_n$)

To show a problem is in $\mathbf{NP}$, it suffices to write a $\exists SO$ sentence capturing the desired property! (e.g., Graph Hamiltonicity)

**Corollary:** $\forall SO$ captures $\mathbf{coNP}$.

**Open problem:** $\exists SO \neq \forall SO$ over finite structures?

($\exists SO \neq \forall SO \;\; \Rightarrow \;\; \mathbf{NP} \neq \mathbf{coNP} \;\; \Rightarrow \;\; \mathbf{P} \neq \mathbf{NP}$)



# $\exists SO = \mathbf{NP}$

**Corollary (Cook-Levin Theorem):** The satisfiability problem for propositional logic is $\mathbf{NP}$ -complete.

**Proof:** Let $\mathcal{P}$ be a problem in $\mathbf{NP}$. Then by Fagin's theorem, there is a $\exists SO$ sentence $\Phi \equiv \exists S_1 \dots \exists S_n\ \varphi$ such that $\mathcal{A} \vDash \Phi$ iff $\mathcal{A}$ is in $\mathcal{P}$.

Now to check $\mathcal{A} \vDash \Phi$, we can construct a propositional formula from $\Phi$ :

- $\mathcal{A}$ is finite! (with universe $A$)
- Replace $\exists x(\dots x \dots)$ with $\bigvee_{a \in A}(\dots a \dots)$
- Replace $\forall x(\dots x \dots)$ with $\bigwedge_{a \in A}(\dots a \dots)$
- Replace every $a \in S_i$ with a proposition "$a\_in\_S_i$"
- Everything else can be evaluated to true or false on $\mathcal{A}$

# $\exists SO = \mathbf{NP}$

**Proof idea:**

- $\exists SO \subseteq \mathbf{NP}$: Suppose $\Phi = \exists X_1 \dots \exists X_n\, \varphi$ . Given $\mathcal{A}$, the NTDM first guesses $S_1, \dots, S_n$ as the certificate and checks if $\varphi(S_1, \dots, S_n)$ holds, which can be done in polynomial time.

- $\mathbf{NP} \subseteq \exists SO$: Yet another encoding of Turing machine, but more challenging!
  - NDTM now, and for arbitrarily long input
  - Solution: explicitly encode the tape as a SO relation $L$
  - Time and position are bounded by $n^k$ so can be encoded as a $k$-tuple!

# A Logic for P?

$\mathbf{NP} = \exists SO$, $\mathbf{coNP} = \forall SO$, $\mathbf{P} = ?$

**Gurevich's Conjecture:** There is no logic that captures $\mathbf{P}$.

- If true, then $\mathbf{P} \neq \mathbf{NP}$!

Some logics do capture $\mathbf{P}$ on *ordered structures*

# $FO(LFP) = \mathbf{P}$ on ordered structures

**Immerman-Vardi Theorem (1982):** $FO(LFP) +\!<$ captures $\mathbf{P}$.

$FO(LFP)$ extends FO with an $\mathbf{lfp}$ operator

- For every monotone operator $F$, the sequence $\emptyset, F(\emptyset), F^2(\emptyset), \ldots$ stabilizes at the least fixed point, denoted by $\mathbf{lfp}(F)$
- E.g., if $F(S) = \{x \mid \forall y\ (E(y,x) \to y \in S)\}$, then $\forall u.\ [\mathbf{lfp}\ F](u)$ says a graph is acyclic

$<$ is a linear order of the structure

# More results for ordered structures

$FO(PFP) +\!<$ captures $\mathbf{PSPACE}$

- Partial fixed point: $\mathbf{pfp}\ F \triangleq \begin{cases} F^{\infty}(\emptyset) & \text{if the sequence stabilizes} \\ \emptyset & \text{othewise} \end{cases}$

$FO(TC) +\!<$ captures $\mathbf{NLOGSPACE}$

- Transitive closure (TC) is a special LFP

$FO(DTC) +\!<$ captures $\mathbf{LOGSPACE}$

- Deterministic transitive closure (e.g., TC applied on the successor relation)

$SO +\!<$ captures $\mathbf{PH}$