“Logic and Complexity Theory”

Neil Immerman

Computer Science Dept.
UMass. Amherst

www.cs.umass.edu/∼immerman

In the beginning there was Cook.

Theorem: [Cook: 1971] SAT is **NP**-complete.

This was followed by.

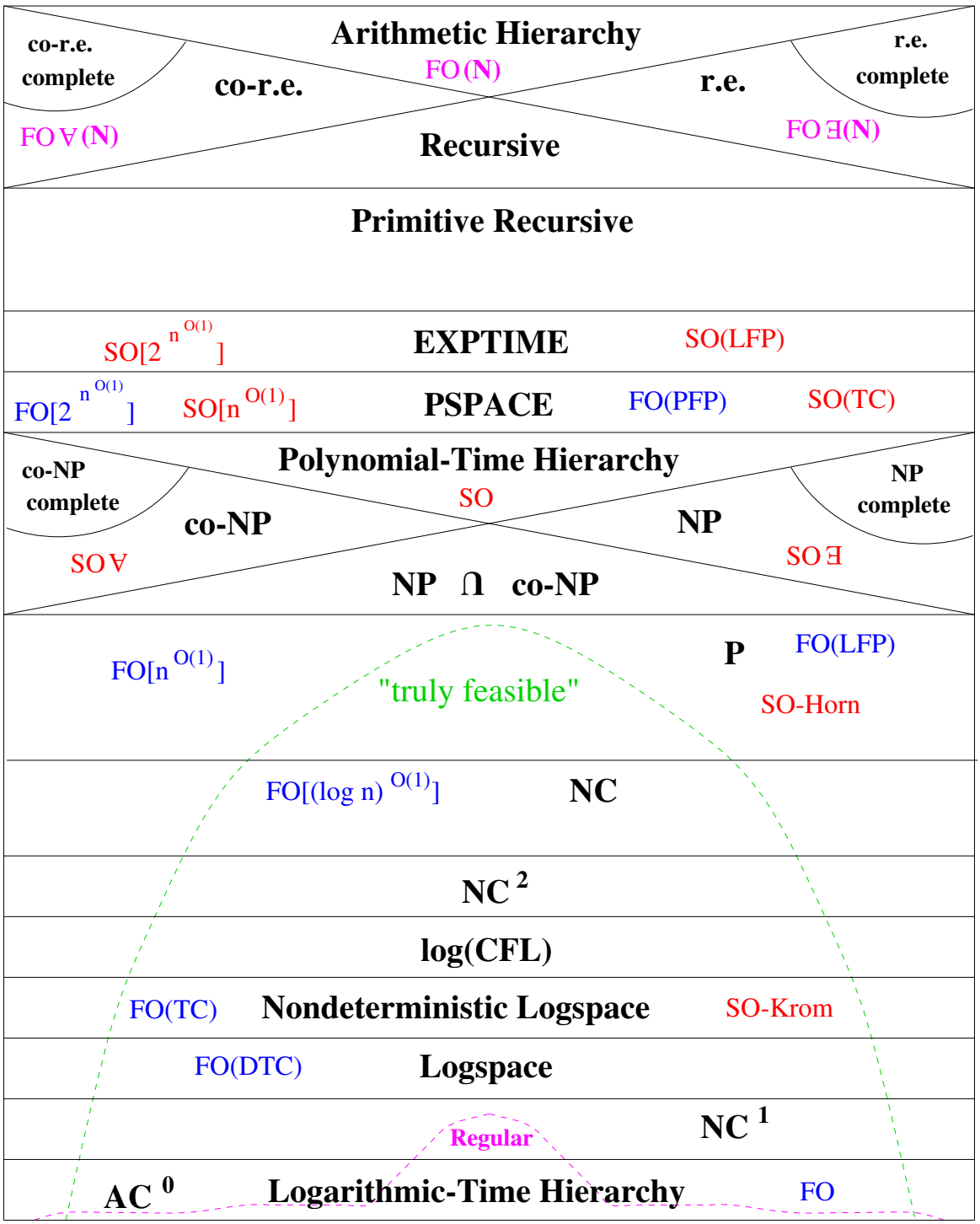
Theorem: [Fagin: 1974] **NP** $=$ SO$\exists$

3-COLOR $\equiv$

$$(\exists RYB)(\forall xy)((R(x) \vee Y(x) \vee B(x)) \wedge (E(x,y) \rightarrow (\neg(R(x) \wedge R(y)) \wedge \neg(Y(x) \wedge Y(y)) \wedge \neg(B(x) \wedge B(y)))))$$

The complexity of computing a query is closely tied to the complexity of describing the query, especially when the input is thought of as a **finite, ordered** *structure.*

Natural complexity classes have natural descriptive characterizations.


Arithmetic Hierarchy
co-r.e. complete
co-r.e.
FO(N)
r.e.
r.e. complete
FO∀(N)
Recursive
FO∃(N)
Primitive Recursive
SO[2^n^O(1)]
EXPTIME
SO(LFP)
FO[2^n^O(1)]
SO[n^O(1)]
PSPACE
FO(PFP)
SO(TC)
Polynomial-Time Hierarchy
co-NP complete
co-NP
SO
NP
NP complete
SO∀
SO∃
NP ∩ co-NP
FO[n^O(1)]
"truly feasible"
P
FO(LFP)
SO-Horn
FO[(log n)^O(1)]
NC
NC^2
log(CFL)
FO(TC)
Nondeterministic Logspace
SO-Krom
FO(DTC)
Logspace
NC^1
Regular
AC^0
Logarithmic-Time Hierarchy
FO

Corollary: $\mathbf{P} = \mathbf{NP} \quad \Leftrightarrow \quad \mathrm{FO}(\mathbf{LFP}) = \mathrm{SO}$

Theorem: [Fagin: 74] Connectivity $\notin$ SO$\exists$(monadic)

Theorem: [de Rougemont: 84] Connectivity $\notin$ SO$\exists$(monadic, suc)

Theorem: [Schwentick: 94] Connectivity $\notin$ SO$\exists$(monadic, $\leq$)

proofs: Ehrenfeucht-Fraïssé games.

Many, many separations such as $\mathrm{FO}_{(\mathrm{wo}\leq)}(\mathbf{TC}) \subsetneqq \mathrm{FO}_{(\mathrm{wo}\leq)}(\mathbf{LFP})$

Theorem: [Abitebout-Vianu: 91] The following conditions are equivalent:

1. $\mathrm{FO}_{(\mathrm{wo}\leq)}(\mathbf{LFP}) = \mathrm{FO}_{(\mathrm{wo}\leq)}(\mathbf{PFP})$
2. $\mathrm{FO}(\mathbf{LFP}) = \mathrm{FO}(\mathbf{PFP})$
3. $\mathbf{P} = \mathbf{PSPACE}$

Theorem: [Grohe: 96] There is a strict arity hierarchy for $\mathrm{FO}_{(\mathrm{wo}\leq)}(\mathbf{TC})$, $\mathrm{FO}_{(\mathrm{wo}\leq)}(\mathbf{LFP})$, and $\mathrm{FO}_{(\mathrm{wo}\leq)}(\mathbf{PFP})$

Theorem: [Bradfield: 96] There is a strict alternation hierarchy for the μ calculus, i.e., $\mathrm{FO}_{(\mathrm{wo}\leq)}(\mathbf{LFP})$(monadic).

Theorem: [Matz-Thomas: 97] There is a strict quantifier-alternation hierarchy for SO(monadic).

Theorem: [Courcelle: 90] SO$\exists$(monadic) restricted to graphs of bounded tree width is contained in linear time.

Theorem: [Cai-Fürer-Immerman: 92]
FO$_{(\text{wo}\leq)}(\mathbf{LFP}, \text{COUNT}) \subsetneq \mathbf{P}$.

In fact, there is an $\Omega(n)$ lower bound on the number of variables needed to identify graphs, cf. Weisfeiler-Lehman.

Theorem: [Grohe: 97] Recognition and isomorphism of planar graphs is expressible in FO($\mathbf{LFP}$).

Theorem: [Grohe-Mariño: 98] Recognition and isomorphism of graphs of bounded tree width is expressible in FO($\mathbf{LFP}$).

Fact: For constructible $t(n)$, $\text{FO}[t(n)] \;=\; \textbf{CRAM}[t(n)]$

Fact: For $k = 1, 2, \ldots,$ $\text{VAR}[k{+}1] \;=\; \textbf{DSPACE}[n^k]$

Facts:

$$\begin{aligned} \text{FO}(\textbf{TC}) &= \textbf{NSPACE}[\log n] \\ \text{FO}(\textbf{LFP}) &= \text{PTIME} \\ \text{SO}\exists &= \text{NP} \\ \text{FO}(\textbf{PFP}) = \text{SO}(\textbf{TC}) &= \textbf{PSPACE} \end{aligned}$$

Natural complexity classes have natural descriptive characterizations.

Theorem: [Papadimitriou-Yannakakis: 91] We can use descriptive characterization of NP optimization problems á la Fagin's Theorem to determine information about their approximability, e.g., definition of MAX-SNP.

Theorem: [AroraLundMotwaniSudanSzegedy: 92 + Hastad: 97]

$$\textbf{NP} \quad = \quad \text{PCP}[\log n, 3]$$

$$S \quad = \quad \{x \mid (\exists \Pi)\varphi(x, \Pi)\}$$

Corollary: If **P** $\neq$ **NP** then 2^{-3} is the optimal approximation ratio for MAX-3SAT.

Many other improved upper and lower bounds on approximation ratios for NP optimization problems.

Reductions and Uniformity

- Complexity Classes (except for hierarchies) tend to have complete problems.
- Natural complete problems remain complete via simple reductions i.e., first-order translations, even first-order projections [Valiant: 82, Immerman: 87].
- Problems complete for a complexity class via fops and other simple translations are isomorphic [Allender-Balcázar-Immerman: 93, Agrawal-Allender-Impagliazzo-Pitassi-Rudich: 97, Cai-Sivakumar: 95]
- Circuit complexity classes can be defined via simple uniformity, e.g, first-order.
- However, Integer-Division in PTIME-uniform $\mathbf{NC}^1$, $\mathbf{ThC}^0$ [Beame-Cook-Hoover: 86].
- Integer-Division in $\mathrm{FO}(M) =$ FO-uniform $\mathbf{ThC}^0$ [Hesse: 01]!

Dichotomy Results and Conjectures

- Often natural problems are complete for natural complexity classes, e.g., **NP**, **P**, **NSPACE**$[\log n]$.
- Sometimes we can look at the syntax of a query and automatically say, "**NP**-complete, non-approximable", "**NP**-complete, approximable", "**P**-complete", "**NSPACE**$[\log n]$-complete", or "Can't tell."
- SO$\exists$ formulas: [Medina-Immerman: 94]
- Constraint Programming: [Feder-Vardi: 98, Kolaitis-Vardi: 98]

Other Future Directions

- Improve game arguments, with ordering.
- Further study tradeoffs, e.g. number of variables versus quantifier-depth.
- Improve understanding of dichotomies and automatic algorithms to determine complexity of descriptions.
- Extend use of fop's and other weak reductions: upper bounds: programming by reduction; and, lower bounds, e.g., proving lower bounds on arity of fop from 3-COLOR to REACH.
- Continue and strengthen ties with applications, e.g., databases model checking, e.g., reading complexity and optimizing via descriptions.