Beyond NP: The Work and Legacy of Larry Stockmeyer

Lance Fortnow

University of Chicago
Larry Joseph Stockmeyer

- 1948 – Born in Indiana
- 1974 – MIT Ph.D.
- IBM Research at Yorktown and Almaden for most of his career
- 82 Papers (11 JACM)
  - 49 Distinct Co-Authors
- 1996 – ACM Fellow
- Died July 31, 2004
The Universe
Computer of Protons
The Universe

11,000,000,000 Light Years
Computer of Protons

Radius $10^{-15}$ Meters
Computing with the Universe

- Universe can only have $10^{123}$ proton gates.
- Consider the true sentences of weak monadic second-order theory of the natural numbers with successor (EWS1S).
  - $\exists A \forall B \exists x (x \in A \rightarrow x+1 \in B)$
- Cannot solve EWS1S on inputs of size 616 in universe with proton-sized gates.
  - Stockmeyer Ph.D. Thesis 1974
  - Stockmeyer-Meyer JACM 2002
The Universe

11,000,000,000 Light Years
The Universe

78,000,000,000 Light Years
Computing with the Universe

- Universe can have $10^{123}$ proton gates.
Computing with the Universe

- Universe can have $3.5 \times 10^{125}$ proton gates.
Computing with the Universe

- Universe can have $3.5 \times 10^{125}$ proton gates.
- Cannot solve EWS1S on inputs of size 616 in universe with proton-sized gates.
Computing with the Universe

- Universe can have $3.5 \times 10^{125}$ proton gates.
- Cannot solve EWS1S on inputs of size 619 in universe with proton-sized gates.
Science Fiction?

- The complexity of algorithms tax even the resources of sixty billion gigabits---or of a universe full of bits; Meyer and Stockmeyer had proved, long ago, that, regardless of computer power, problems existed which could not be solved in the life of the universe.
Evolution of Complexity
Evolution of Complexity
Turing-Church-Kleene-Post 1936

Computably
Enumerable

Computable
Evolution of Complexity

Computably
Enumerable
Evolution of Complexity
Kleene 1956

- Computably Enumerable
- Regular Languages
  - Finite Automata
Evolution of Complexity

Chomsky Hierarchy 1956

Computably Enumerable

Regular Languages
Finite Automata
Evolution of Complexity

Chomsky Hierarchy 1956

- Computably Enumerable
  - Unrestricted Grammars

- Context-Sensitive Grammars
  - Linear-Bounded Automata

- Context-Free Grammars
  - Push-Down Automata

- Regular Languages
  - Finite Automata
  - Regular Grammars
Faster Computers
Evolution of Complexity

- Computably
- Enumerable

- Computable
Evolution of Complexity
Evolution of Complexity

Computable
Evolution of Complexity
Hartmanis-Stearns 1965

Computable

TIME(n²)
## Evolution of Complexity

Hartmanis-Stearns 1965

<table>
<thead>
<tr>
<th>Computable</th>
<th>TIME($2^n$)</th>
<th>TIME($n^5$)</th>
<th>TIME($n^2$)</th>
</tr>
</thead>
</table>
Limitations of DTIME(t(n))

- Not Machine Independent.
- Separations are by diagonalization and not by natural problems.
- No clear notion of efficient computation.
Evolution of Complexity
Cobham 1964 Edmonds 1965
Computable
Evolution of Complexity
Cobham 1964 Edmonds 1965

Computable

\[ P = \bigcup \text{DTIME}(n^k) \]
Evolution of Complexity

Cobham 1964 Edmonds 1965

Computable

- Matching

\[ P = \bigcup DTIME(n^k) \]
Evolution of Complexity

Computable

P
# Evolution of Complexity


<table>
<thead>
<tr>
<th>Computable</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAT</td>
</tr>
<tr>
<td>Clique</td>
</tr>
<tr>
<td>NP</td>
</tr>
<tr>
<td>Partition</td>
</tr>
<tr>
<td>Max Cut</td>
</tr>
<tr>
<td>P</td>
</tr>
</tbody>
</table>
State of Complexity 1972

<table>
<thead>
<tr>
<th>Computable</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP</td>
</tr>
<tr>
<td>P</td>
</tr>
</tbody>
</table>
Enter Larry Stockmeyer

- January 1972 – Bachelors/Masters at MIT
  - Bounds on Polynomial Evaluation Algorithms
- Can we find natural hard problems?
  - Diagonalization methods do not lead to natural problems.
  - There are natural NP-complete problems but cannot prove them not in P.
- With Advisor Albert Meyer
Regular Expressions with Squaring

- $(0+1)^*00(0+1)^*00(0+1)^*$
  - All strings with two sets of consecutive zeros.

- Allow Squaring operator: $r^2=rr$

- $(0+1)^*(0^2(0+1)^*)^2$

- No more expressive power but can be much shorter when used recursively.
  - $((((((0^2)^2)^2)^2)^2)^2)=\ldots$
Meyer-Stockmeyer 1972

\[ \text{REGSQ} = \{ R \mid L(R) \neq \Sigma^* \} \]

<table>
<thead>
<tr>
<th>Computable</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>REGSQ</td>
<td>EXPSPACE</td>
</tr>
<tr>
<td>PSPACE</td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td></td>
</tr>
</tbody>
</table>
Regular Expressions with Squaring

- Meyer and Stockmeyer, “The Equivalence Problem for Regular Expressions with Squaring Requires Exponential Space” – SWAT 1972

- MINIMAL
  - Set of Boolean formulas with no smaller equivalent formula.
Complexity of MINIMAL

<table>
<thead>
<tr>
<th>MINIMAL</th>
<th>NP</th>
<th>P</th>
</tr>
</thead>
</table>

Computable
MINIMAL

- MINIMAL
  - Set of Boolean formulas with no smaller equivalent formula.

- MINIMAL in NP?
  - Can’t check all smaller formulas.
**Meyer-Stockmeyer 1972**

Complexity of MINIMAL

<table>
<thead>
<tr>
<th>MINIMAL</th>
<th>MINIMAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP</td>
<td>P</td>
</tr>
</tbody>
</table>
MINIMAL

- Set of Boolean formulas with no smaller equivalent formula.

MINIMAL in NP?

- Can’t check all smaller formulas.

MINIMAL in NP?

- Can’t check equivalence.
MINIMAL

- MINIMAL
  - Set of Boolean formulas with no smaller equivalent formula.

- MINIMAL in NP?
  - Can’t check all smaller formulas.

- MINIMAL in NP?
  - Can’t check equivalence.

- MINIMAL is in NP with an “oracle” for equivalence.
MINIMAL in NP with Equivalence Oracle

\[(x \lor y) \land (x \lor y) \land z\]

Guess: \(x \land z\)

\[(x \land z, (x \lor y) \land (x \lor \bar{y}) \land z)\]

EQUIVALENT

Equivalence
MINIMAL

- MINIMAL is in NP with an “oracle” for equivalence or non-equivalence.
MINIMAL

- MINIMAL is in NP with an “oracle” for equivalence or non-equivalence.
- Since non-equivalence is in NP we can solve MINIMAL in NP with NP oracle.
MINIMAL is in NP with an “oracle” for equivalence or non-equivalence.

Since non-equivalence is in NP we can solve MINIMAL in NP with NP oracle.

Suggests a “hierarchy” above NP.
Meyer-Stockmeyer 1972
The Polynomial Time Hierarchy

\[ \text{MINIMAL} \]

\[ \text{NP} \]

\[ \text{NP}^{\text{NP}} \]
Meyer-Stockmeyer 1972

The Polynomial Time Hierarchy

\[ \text{NP}^\text{NP} \]

\[ \text{NP} = \Sigma_1^\text{P} \]

\[ \text{P} \]
The Polynomial Time Hierarchy

<table>
<thead>
<tr>
<th>$\text{NP}^{\Sigma_3^p} = \Sigma_4^p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{NP}^{\Sigma_2^p} = \Sigma_3^p$</td>
</tr>
<tr>
<td>$\text{NP}^{\Sigma_2^p} = \Sigma_3^p$</td>
</tr>
<tr>
<td>$\text{NP}^{\Sigma_1^p} = \Sigma_2^p$</td>
</tr>
<tr>
<td>$\text{NP} = \Sigma_1^p$</td>
</tr>
<tr>
<td>$\text{P}$</td>
</tr>
</tbody>
</table>
Meyer-Stockmeyer 1972
The Polynomial Time Hierarchy

| $\Sigma_4^p$ | $\text{co-NP}^{\Sigma_3^p} = \Pi_4^p$ |
| $\Sigma_3^p$ | $\text{co-NP}^{\Sigma_2^p} = \Pi_3^p$ |
| $\Sigma_2^p$ | $\text{MINIMAL}$ $\text{co-NP}^{NP} = \Pi_2^p$ |
| $\Sigma_1^p = \text{NP}$ | $\text{co-NP} = \Pi_1^p$ |
| **P** |  |
The Polynomial Time Hierarchy

<table>
<thead>
<tr>
<th>PH</th>
<th>( \Sigma_4^p )</th>
<th>( \Pi_4^p )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \Sigma_3^p )</td>
<td>( \Pi_3^p )</td>
</tr>
<tr>
<td>P( \Sigma_2^p = \Delta_3^p )</td>
<td>( \Sigma_2^p )</td>
<td>( \Pi_2^p )</td>
</tr>
<tr>
<td>P( \Sigma_1^p = \text{NP} )</td>
<td>( \Sigma_1^p = \text{NP} )</td>
<td>( \text{co-NP} = \Pi_1^p )</td>
</tr>
<tr>
<td>P( \Sigma_3^p = \Delta_4^p )</td>
<td>( \Sigma_4^p )</td>
<td>( \Pi_4^p )</td>
</tr>
<tr>
<td>P=( \Delta_1^p )</td>
<td>P=( \Delta_1^p )</td>
<td>P=( \Delta_1^p )</td>
</tr>
</tbody>
</table>
Properties of the Hierarchy

- Meyer-Stockmeyer, “The Equivalence Problem for Regular Expressions with Squaring Requires Exponential Space”, SWAT 1972
# Properties of the Hierarchy

<table>
<thead>
<tr>
<th>Π₁^p</th>
<th>Co-NP=Π₁^p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Π₂^p</td>
<td>P=Δ₁^p</td>
</tr>
<tr>
<td>Π₃^p</td>
<td></td>
</tr>
<tr>
<td>Π₄^p</td>
<td></td>
</tr>
<tr>
<td>Δ₄^p</td>
<td></td>
</tr>
<tr>
<td>Δ₃^p</td>
<td></td>
</tr>
<tr>
<td>Δ₂^p</td>
<td></td>
</tr>
<tr>
<td>Δ₁^p</td>
<td></td>
</tr>
<tr>
<td>Σ₄^p</td>
<td></td>
</tr>
<tr>
<td>Σ₃^p</td>
<td></td>
</tr>
<tr>
<td>Σ₂^p</td>
<td></td>
</tr>
<tr>
<td>Σ₁^p=NP</td>
<td></td>
</tr>
<tr>
<td>PSPACE</td>
<td></td>
</tr>
</tbody>
</table>

PH
## Properties of the Hierarchy

<table>
<thead>
<tr>
<th></th>
<th>$\Sigma_4^p$</th>
<th>$\Sigma_3^p$</th>
<th>$\Sigma_2^p$</th>
<th>$\Sigma_1^p=\text{NP}$</th>
<th>$P=\Delta_1^p$</th>
<th>$\text{Co-NP}=\Pi_1^p$</th>
<th>$\Pi_2^p$</th>
<th>$\Pi_3^p$</th>
<th>$\Pi_4^p$</th>
<th>$\text{PSPACE}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta_4^p$</td>
<td>$\Delta_3^p$</td>
<td>$\Delta_2^p$</td>
<td>$\text{NP}$</td>
<td>$P$</td>
<td>PH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- $\Sigma_4^p$ is the fourth level of the polynomial hierarchy.
- $\Pi_4^p$ is the fourth level of the polynomial hierarchy, but in the co-side.
- $\text{Co-NP}=\Pi_1^p$ refers to the co-nondeterministic polynomial time complexity class.
- $P=\Delta_1^p$ indicates that $P$ is the first level of the polynomial hierarchy.
- The hierarchy includes $\text{PSPACE}$, which is the class of problems solvable by a polynomial-space Turing machine.

The diagram illustrates the relationships and inclusions between these complexity classes.
## Properties of the Hierarchy

<table>
<thead>
<tr>
<th></th>
<th>$\Sigma_4^p$</th>
<th>$\Sigma_3^p = \Delta_3^p$</th>
<th>$\Sigma_2^p$</th>
<th>$\Sigma_1^p = \text{NP}$</th>
<th>$\Delta_2^p$</th>
<th>$\Delta_4^p$</th>
<th>$\Pi_4^p$</th>
<th>$\Pi_3^p$</th>
<th>$\Pi_2^p$</th>
<th>$\Pi_1^p$</th>
<th>$\text{Co-NP} = \Pi_1^p$</th>
<th>$\text{PSPACE}$</th>
<th>$\text{P} = \Delta_1^p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Sigma_4^p$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Delta_4^p$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Pi_4^p$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Pi_3^p$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Pi_2^p$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Pi_1^p$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{Co-NP} = \Pi_1^p$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{P} = \Delta_1^p$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- $\Sigma_k^p$: Class of languages decidable by a deterministic Turing machine in time $P(2^{O(n)})$, where $P$ is a polynomial function of $n$.
- $\Pi_k^p$: Class of languages complementable by a deterministic Turing machine in time $P(2^{O(n)})$.
- $\Delta_k^p$: Class of languages decidable by a deterministic Turing machine in time $P(2^{O(n)})$ and its complement is also in the same class.
- $\text{PSPACE}$: Class of languages decidable by a nondeterministic Turing machine in polynomial space.
- $\text{PH}$: Polynomial Hierarchy.
## Properties of the Hierarchy

<table>
<thead>
<tr>
<th>PSPACE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PH</td>
<td></td>
</tr>
<tr>
<td>(\Sigma_4^p)</td>
<td>(\Pi_4^p)</td>
</tr>
<tr>
<td>(\Delta_4^p)</td>
<td>(\Pi_3^p)</td>
</tr>
<tr>
<td>(\Sigma_3^p=\Delta_3^p)</td>
<td>(\Pi_2^p)</td>
</tr>
<tr>
<td>(\Sigma_1^p=\text{NP})</td>
<td>(\Delta_2^p)</td>
</tr>
<tr>
<td>(\text{Co-NP}=\Pi_1^p)</td>
<td>(\text{P}=\Delta_1^p)</td>
</tr>
</tbody>
</table>
## Properties of the Hierarchy

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Sigma_2^p$</td>
<td>PH</td>
<td>$\Sigma_3^p$</td>
<td>$\Delta_3^p$</td>
<td>$\Pi_3^p$</td>
<td>$\Pi_2^p$</td>
</tr>
<tr>
<td>$\Sigma_1^p=NP$</td>
<td>$\Delta_2^p$</td>
<td>Co-NP</td>
<td>$\Pi_1^p$</td>
<td>$P=\Delta_1^p$</td>
<td>PSPACE</td>
</tr>
</tbody>
</table>
## Properties of the Hierarchy

If $P = NP$

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSPACE</td>
</tr>
<tr>
<td></td>
<td>$P=NP=PH$</td>
</tr>
</tbody>
</table>
Properties of the Hierarchy

\[
\begin{array}{c}
\text{PSPACE} \\
\text{PH} \\
\Sigma_4^p \\
\Sigma_3^p \\
\Sigma_2^p \\
\Sigma_1^p = \text{NP} \\
P = \Delta_1^p \\
\end{array}
\]

\[
\begin{array}{c}
\Delta_4^p \\
\Delta_3^p \\
\Delta_2^p \\
\Pi_4^p \\
\Pi_3^p \\
\Pi_2^p \\
\Pi_1^p = \text{Co-NP} \\
\end{array}
\]
# Properties of the Hierarchy

<table>
<thead>
<tr>
<th></th>
<th>PSPACE</th>
<th>PH</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Sigma_4^p$</td>
<td></td>
<td>$\Delta_4^p$</td>
<td></td>
</tr>
<tr>
<td>$\Sigma_3^p$</td>
<td>$\Sigma_4^p$</td>
<td>$\Delta_3^p$</td>
<td>$\Pi_3^p$</td>
</tr>
<tr>
<td>$\Sigma_2^p$</td>
<td>$\Sigma_3^p$</td>
<td>$\Delta_2^p$</td>
<td>$\Pi_2^p$</td>
</tr>
<tr>
<td>$\Sigma_1^p=NP$</td>
<td>$\Sigma_2^p$</td>
<td>Co-NP=Π₁^p</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\Sigma_1^p=NP$</td>
<td>$\Sigma_1^p$</td>
<td>$\Delta_1^p$</td>
</tr>
</tbody>
</table>
Properties of the Hierarchy

\[
\begin{array}{c|c|c|c|c}
& \Sigma_4^p & \Sigma_3^p & \Sigma_2^p & \Sigma_1^p=NP \\
\hline
PH=PSPACE & \Delta_4^p & \Delta_3^p & \Delta_2^p & \text{Co-NP}=\Pi_1^p \\
\hline
\Pi_4^p & \Pi_3^p & \Pi_2^p & \text{Co-NP}=\Pi_1^p & \text{P}=\Delta_1^p \\
\end{array}
\]
### Properties of the Hierarchy

<table>
<thead>
<tr>
<th>Layer</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Sigma_1^p = \text{NP} )</td>
<td>Co-NP = ( \Pi_1^p )</td>
</tr>
<tr>
<td>( \Sigma_2^p )</td>
<td>( \Pi_2^p )</td>
</tr>
<tr>
<td>( \Delta_3^p )</td>
<td>( \Pi_3^p )</td>
</tr>
<tr>
<td>( \Sigma_3^p )</td>
<td>( \Pi_4^p )</td>
</tr>
<tr>
<td>( \Delta_4^p )</td>
<td>PH = PSPACE</td>
</tr>
<tr>
<td>( \Sigma_4^p )</td>
<td>P = ( \Delta_4^p )</td>
</tr>
</tbody>
</table>
Properties of the Hierarchy

\[ \Sigma_2^p = \text{PSPACE} = \Sigma_3^p = \Delta_3^p = \Pi_3^p \]

\[ \Sigma_1^p = \text{NP} \]

\[ \Delta_2^p = \text{Co-NP} = \Pi_1^p \]

\[ \text{P} = \Delta_1^p \]
Quantifier Characterization

A language $L$ is in $\Sigma_3^P$ if for all $x$ in $\Sigma^*$

$x$ is in $L \iff \exists u \ \forall v \ \exists w \ P(x,u,v,w)$

A language $L$ is in $\Pi_3^P$ if for all $x$ in $\Sigma^*$

$x$ is in $L \iff \forall u \ \exists v \ \forall w \ P(x,u,v,w)$
We define $B_3$ by the set of true quantified formula of the form

$$\exists x_1 \exists x_2 \cdots \exists x_n \forall y_1 \cdots \forall y_n \exists z_1 \cdots \exists z_n \varphi(x_1, \ldots, x_n, y_1, \ldots, y_n, z_1, \ldots, z_n)$$
Complete Sets in the Hierarchy

\[
\begin{array}{c}
P=\Delta_1^p \\
\Pi_4^p = \text{Co-NP} \\
\Pi_3^p \\
\Pi_2^p \\
\Delta_3^p \\
\Delta_4^p \\
\Sigma_3^p \\
\Sigma_4^p \\
\end{array}
\]
Natural Complete Sets

- **N-INEQ** – Inequivalence of Integer Expressions with union and addition.
  
  \[(50 + (40 \cup 20 \cup 15)) \cup ((2 \cup 5) + (7 \cup 9))\]

- **Meyer-Stockmeyer 1973 Stockmeyer 1977**
  - N-INEQ is \(\Sigma_2^p\)-complete

- **Umans 1999**
  - Succinct Set Cover is \(\Sigma_2^p\)-complete

- **Schafer 1999**
  - Succinct VC Dimension is \(\Sigma_3^p\)-complete
The $\omega$-jump of the Hierarchy

- Meyer-Stockmeyer 1973, Stockmeyer 1977
  \[ B_\omega = \bigcup B_k \]
- Quantified Boolean Formula with an unbounded number of alterations.
- Now called QBF or TQBF.
Complexity of $\omega$-jump

<table>
<thead>
<tr>
<th></th>
<th>$\mathcal{B}_\omega$ (TQBF)</th>
<th>PSPACE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Sigma_4^p$</td>
<td>$B_4$</td>
<td></td>
</tr>
<tr>
<td>$\Sigma_3^p$</td>
<td>$B_3$</td>
<td>$\Delta_4^p$</td>
</tr>
<tr>
<td>$\Sigma_2^p$</td>
<td>$B_2$</td>
<td>$\Delta_3^p$</td>
</tr>
<tr>
<td>$\Sigma_1^p=NP$</td>
<td>$B_1=\text{SAT}$</td>
<td>$\Delta_2^p$</td>
</tr>
<tr>
<td>$\Sigma_1^p=NP$</td>
<td>$B_1=\text{SAT}$</td>
<td>$\Delta_2^p$</td>
</tr>
<tr>
<td>$P=\Delta_1^p$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Alternation

- Chandra-Kozen-Stockmeyer JACM 1981
- Chandra-Stockmeyer STOC 1976
- Kozen FOCS 1976
Alternation
Alternation
Alternation

Acc Acc Acc Acc Acc Acc Acc AccAccAcc Acc Acc Acc Acc Acc Acc
Alternation
Alternation Theorems

- Chandra-Kozen-Stockmeyer
- $\text{ATIME}(t(n)) \subseteq \text{DSPACE}(t(n))$
- $\text{NSPACE}(s(n)) \subseteq \text{ATIME}(s^2(n))$
- $\text{ASPACE}(s(n)) = \bigcup \text{DTIME}(c^{s(n)})$

$L \subseteq P \subseteq \text{PSPACE} \subseteq \text{EXP} \subseteq \text{EXPSPACE} \subseteq \ldots$

$\text{II} \quad \text{II} \quad \text{II} \quad \text{II}$

$\text{AL} \subseteq \text{AP} \subseteq \text{APSPACE} \subseteq \text{AEXP} \subseteq \ldots$
Alternate Characterization of $\Sigma_2^p$
Other Alternating Models

Chandra-Kozen-Stockmeyer 1981

- **Log-Space Hierarchy**
  - Collapses to NL (Immerman-Szelepcsényi ’88)

- **Alternating Finite State Automaton**
  - Same power as DFA but doubly exponential blowup in states.

- **Alternating Push-Down Automaton**
  - Accepts exactly $E = \text{DTIME}(2^{O(n)})$
  - Strictly stronger than PDAs
  - Inclusion due to Ladner-Lipton-Stockmeyer ’78
Alternation as a Game
Alternation as a Game
Alternation as a Game

- Rej
- Rej
- Acc
- Acc
- Acc
- Rej
- Rej
- Rej
- Rej
- Acc
- Rej
- Rej
Alternation as a Game
Alternation as a Game
Complete Sets Via Games

- Stockmeyer-Chandra 1979
- Can use problems based on games to get completeness results for PSPACE and EXP.
- Create a combinatorial game that is EXP-complete and thus not decidable in P.
- First complete sets for PSPACE and EXP not based on machines or logic.
Checkers
Generalized Checkers
Generalized Checkers

- PSPACE-hard
  - Fraenkel et al. 1978
- EXP-complete
  - Robson 1984
Approximate Counting

- **#P – Valiant 1979**
  - Functions that count solutions of NP problems.
  - Permanent is #P-complete

- **Stockmeyer 1985 building on Sipser 1983**
  - Can approximate any #P function $f$ in polytime with an oracle for $\Sigma^p_2$.

- **Toda 1991**
  - Every language in PH reducible to #P
## Complexity of #$P$

<table>
<thead>
<tr>
<th></th>
<th>$P$</th>
<th>$\Sigma_1^P=NP$</th>
<th>$\Sigma_2^P$</th>
<th>$\Sigma_3^P$</th>
<th>$\Sigma_4^P$</th>
<th>$\Pi_2^P$</th>
<th>$\Pi_3^P$</th>
<th>$\Pi_4^P$</th>
<th>$\Delta_2^P$</th>
<th>$\Delta_3^P$</th>
<th>$\Delta_4^P$</th>
<th>$\Sigma#P$</th>
<th>$P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Sigma_1^P=NP$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Sigma_2^P$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Sigma_3^P$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Sigma_4^P$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Pi_2^P$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Pi_3^P$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Pi_4^P$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Delta_2^P$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Delta_3^P$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Delta_4^P$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Sigma#P$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$P$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**
- $\Sigma_i^P$ and $\Pi_i^P$ denote the $i$-th level of the Polynomial Hierarchy (PH).
- $\Delta_i^P$ denotes the $i$-th level of the Polynomial Hierarchy (PH).
- $P$ and $NP$ are classes of decision problems.
- $P#P$ is the class of counting problems.
- $Perm$ is the class of polynomial-time computable permutations.
- $\text{Co-NP} = \Pi_1^P$ is the class of problems whose complements are in $\Pi_1^P$.
- $\text{Approx-}#P$ is the class of counting problems that are approximable within a factor of $2+\epsilon$ for any $\epsilon > 0$.

**Abbreviations:**
- $\Sigma$, $\Pi$, and $\Delta$ denote the classes of the Polynomial Hierarchy (PH).
- $P$ denotes the class of decision problems solvable in polynomial time.
- $NP$ denotes the class of decision problems whose complements are in $P$.
- $PSPACE$ denotes the class of decision problems solvable in polynomial space.
- $PH$ denotes the Polynomial Hierarchy.
Legacy of Larry Stockmeyer

- Circuit Complexity
- Infinite Hierarchy Conjecture
- Probabilistic Computation
- Interactive Proof Systems
Circuit Complexity

- **Baker-Gill-Solovay ’75: Relativization Paper**
  - Open: Is PH infinite relative to an oracle?
- **Sipser ’83: Strong lower bounds on depth d circuits simulating depth d+1 circuits.**
- **Yao ’85: “Separating the Polynomial-Time Hierarchy by Oracles”**
- Led to future circuit results by Håstad, Razborov, Smolensky and many others.
Infinite Hierarchy Conjecture

- Is the Polynomial-Time Hierarchy Infinite?
- Best Evidence: Yao’s result shows alternating log-time hierarchy infinite.
- Many complexity results
  - If PROP then the polynomial-time hierarchy collapses.
  - If PH is infinite then NOT PROP.
- Gives evidence for NOT PROP.
If Hierarchy is Infinite ...

- SAT does not have small circuits.
  - Karp-Lipton 1980

- Graph isomorphism is not NP-complete.
  - Goldreich-Micali-Wigderson 1991
  - Goldwasser-Sipser 1989
  - Boppana-Håstad-Zachos 1987

- Boolean hierarchy is infinite.
  - Kadin 1988
Boolean Hierarchy

- $BH_1 = NP$
- $BH_{k+1} = \{ B - C \mid B \text{ in NP and } C \text{ in } BH_k\}$
- $\{ (G,k) \mid \text{Max clique of } G \text{ has size } k \}$ in $BH_2$
- Kadin: If $BH_k = BH_{k+1}$ then $PH = \Sigma_3^p$. 
Probabilistic Computation

\[
\begin{array}{c|c|c|c|c}
 & \Sigma_4^p & \Delta_4^p & \Pi_4^p \\
\hline
P\#P & & & \\
\hline
PH & & & \\
\hline
\Sigma_3^p & \Delta_3^p & \Pi_3^p & \\
\hline
\Sigma_2^p & \Delta_2^p & \Pi_2^p & \\
\hline
\Sigma_1^p = NP & \Delta_2^p & & Co-NP = \Pi_1^p \\
\hline
P = \Delta_1^p & & & \\
\end{array}
\]
Probabilistic Computation
Sipser-Gács-Lautemann 1983

\[
\begin{array}{c|c}
\text{PSPACE} & \\
\hline
\text{P}^\# & \\
\hline
\Sigma_4^p & \Pi_4^p \\
\hline
\Sigma_3^p & \Delta_4^p & \Pi_3^p \\
\hline
\Sigma_2^p & \Delta_3^p & \Pi_2^p \\
\hline
\Sigma_1^p=\text{NP} & \text{BPP} & \text{Co-NP}=\Pi_1^p \\
\hline
\text{P}=\Delta_1^p & \\
\end{array}
\]
Interactive Proof Systems

- Papadimitriou 1985 – Alternation between nondeterministic and probabilistic players
- Interactive Proof Systems
  - Public Coin: Babai-Moran 1988
  - Private Coin: Goldwasser-Micali-Rackoff 1989
  - Equivalent: Goldwasser-Sipser 1989
Interactive Proof Systems
Babai-Moran 1988

\[
\begin{array}{c}
\text{PSPACE} \\
\text{P#P} \\
\text{PH} \\
\Sigma_4^p \\
\Sigma_3^p \\
\Sigma_2^p \\
\Sigma_1^p = \text{NP} \\
\end{array}
\]

\[
\begin{array}{c}
P = \Delta_1^p \\
\text{BPP, MA} \\
\Delta_2^p \\
\Delta_3^p \\
\Delta_4^p \\
\Pi_3^p \\
\Pi_4^p \\
\end{array}
\]

\[
\begin{array}{c}
PSPACE \\
P#P \\
PH \\
\Sigma_4^p \\
\Sigma_3^p \\
\Sigma_2^p \\
\Sigma_1^p = \text{NP} \\
\end{array}
\]
Interactive Proof Systems
LFKN, Shamir 1992

<table>
<thead>
<tr>
<th>PSPACE=IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P^{#P} )</td>
</tr>
<tr>
<td>( \Sigma_4^p )</td>
</tr>
<tr>
<td>( \Sigma_3^p )</td>
</tr>
<tr>
<td>( \Sigma_2^p )</td>
</tr>
<tr>
<td>( \Sigma_1^p=NP )</td>
</tr>
<tr>
<td>( P^{\Delta_2^P} )</td>
</tr>
<tr>
<td>( \Delta_3^p )</td>
</tr>
<tr>
<td>( \Delta_4^p )</td>
</tr>
<tr>
<td>( \Pi_3^p )</td>
</tr>
<tr>
<td>( \Pi_4^p )</td>
</tr>
<tr>
<td>( P=\Delta_1^p )</td>
</tr>
<tr>
<td>( \text{BPP, MA, AM} )</td>
</tr>
<tr>
<td>( \text{Co-NP=} )</td>
</tr>
<tr>
<td>( \Pi_2^p )</td>
</tr>
</tbody>
</table>
Interactive Proof Systems

- Hardness of Approximation
  - Feige-Goldwasser-Lovász-Safra-Szegedy 1996

- Probabilistically Checkable Proofs
  - NP in PCPs with $O(\log n)$ coins and constant number of queries.

- Interactive Proofs with Finite State Verifiers
  - Dwork and Stockmeyer
Larry Stockmeyer contributed much more to complexity and important work in other areas including automata theory and parallel and distributed computing.

Most Cited Article (CiteSeer):

Conclusion

- What natural problems can’t we compute?
- Led to exciting work on polynomial-time hierarchy, alternation, approximation and much more.
- These idea affect much of computational complexity today and the legacy will continue for generations in the future.
Remembering

- Other members of our community that we have recently lost...
Seymour Ginsburg
Clemens Lautemann
Carl Smith