Three results about BPP

BPP is in P/poly (Adleman)
BPP is in the polynomial time hierarchy (Sipser-Gacs-Lautemann)
If NP is in BPP then the polynomial time hierarchy collapses (Karp-Lipton)

Sharpening our bounds for BPP

- Some conjecture that BPP = P
- Adleman showed a weaker result:

$$\text{BPP} \subseteq \text{P/poly}$$

- P/poly is the class of languages that are accepted by *TM's with advice*, or equivalently, *polynomial-sized circuit families*

P/poly and TMs with advice

- A *TM with advice* is a deterministic TM that, in addition to its input, also gets an "advice" string A(n) which depends on the input length n but not otherwise on the input

- The TM gets the *same* advice for all inputs of length n

- L is in P/poly iff there is a poly time bounded TM M with advice, and a sequence of polynomial-length advice strings {A(n)| n in $\mathbb{N}$ } such that for all inputs *w*,

  w is in L iff M accepts on (*w*, A(|*w*|)

Example: TM with advice

- Let M_n be the TM encoded by the binary representation of the number n
- Let Unary-Halt be the undecidable language

$$\{1^n : M_n \text{ outputs 1 on input } 1^n\}$$

Example: TM with advice

- Let M_n be the TM encoded by the binary representation of the number n
- Let Unary-Halt be the undecidable language

$$\{1^n : M_n \text{ outputs } 1 \text{ on input } 1^n\}$$

- There is a halting TM with advice that accepts Unary-Halt: simply let A(n) be 1 if M_n outputs 1 on input 1^n and 0 otherwise!

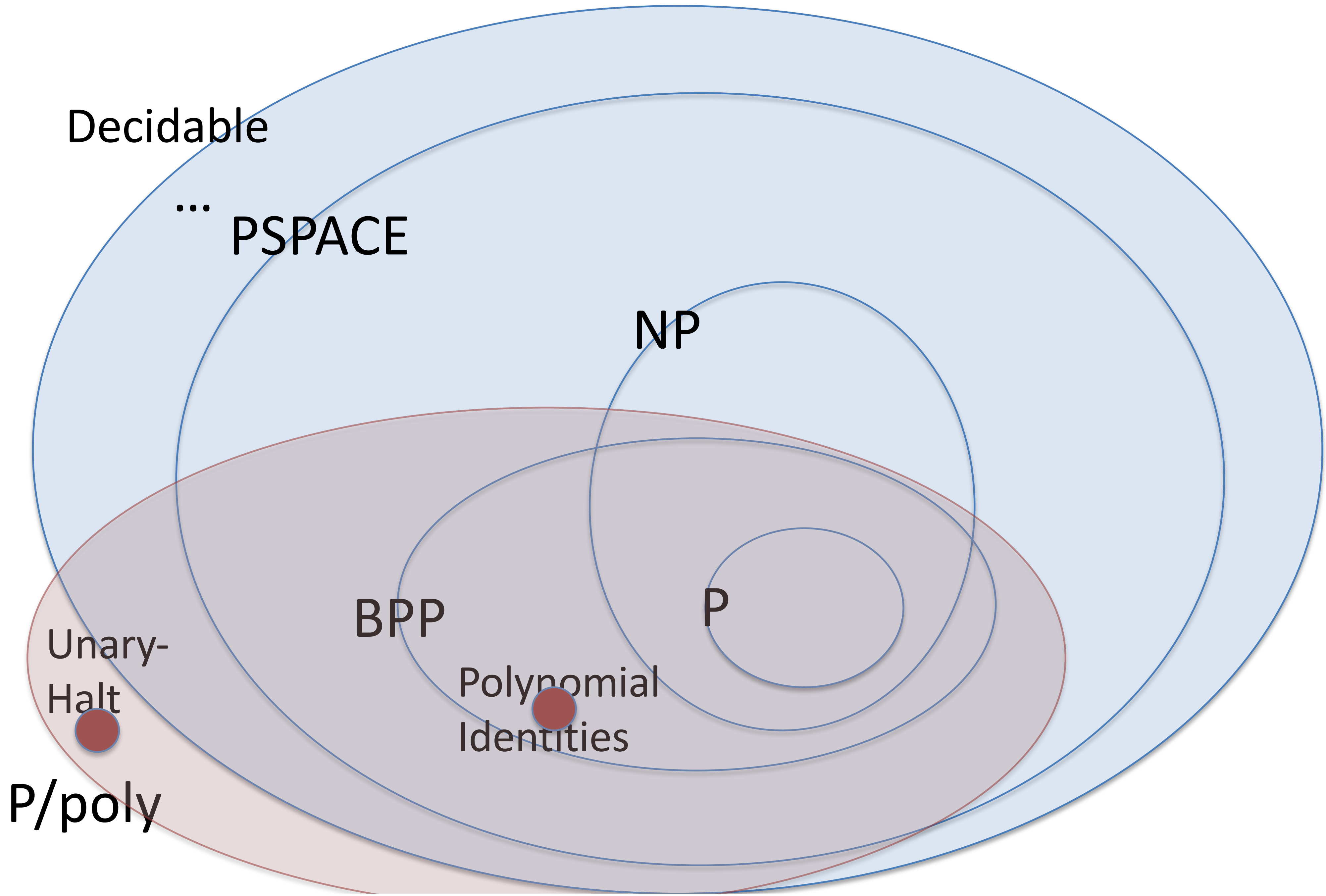
Example: TM with advice

- Let M_n be the TM encoded by the binary representation of the number n
- Let Unary-Halt be the undecidable language

$$\{1^n : M_n \text{ outputs } 1 \text{ on input } 1^n\}$$

- There is a halting TM with advice that accepts Unary-Halt: simply let A(n) be 1 if M_n outputs 1 on input 1^n and 0 otherwise!
- There is also a family of *nonuniform* circuits that accepts Unary-Halt

Where BPP lies in our complexity classes



BPP is in P/poly

BPP is in the Polynomial Time Hierarchy (PH)

- Before defining PH, we'll first introduce new variants of SAT

A Hierarchy of Quantified SAT Problems

- $\sum_k$ SAT: set of true quantified formulas of the form

$$\exists X_1 \, \forall X_2 \, \ldots \, Q_k X_k \, \phi(X_1, X_2, \ldots, X_k)$$

  where for some $n \geq 0$
  - ϕ is a Boolean formula over kn variables
  - $X_i = x_{i1}, \ldots, x_{in}$ for all i is a truth assignment to variables of ϕ

A Hierarchy of Quantified SAT Problems

- $\sum_k$ SAT: set of true quantified formulas of the form

  $$\exists X_1 \, \forall X_2 \ldots Q_k X_k \, \phi(X_1, X_2, \ldots, X_k)$$

  where for some $n \geq 0$
  - ϕ is a Boolean formula over kn variables
  - $X_i = x_{i1}, \ldots, x_{in}$ for all i is a truth assignment to variables of ϕ

- $\prod_k$ SAT: similar, but starts with a $\forall$ quantifier

- In contrast, TQBF has no fixed limit k on the number quantifier alternations

The Polynomial Time Hierarchy

The Polynomial Time Hierarchy

- Σ_1^p: set of languages L for which there is a polynomial-time DTM* M such that

  w is in L iff $\exists Z_1$ M(w,Z_1) accepts

  where $|Z_1|$ is polynomial in $|w|$
- Σ_1^p is a fancy name for NP

* DTM: Deterministic Turing Machine

The Polynomial Time Hierarchy

- Σ_1^p: set of languages L for which there is a polynomial-time DTM M such that

  w is in L iff $\exists Z_1$ M(w,Z_1) accepts

  where $|Z_1|$ is polynomial in $|w|$
- Σ_1^p is a fancy name for NP

- Π_1^p: like Σ_1^p, but with a $\forall$ quantifier, so Π_1^p = co-NP

- Note that if $\Pi_1^p \subseteq \Sigma_1^p$ then in fact $\Pi_1^p = \Sigma_1^p$

The Polynomial Time Hierarchy

Claim: If $\Pi_1^p \subseteq \Sigma_1^p$ then in fact $\Pi_1^p = \Sigma_1^p$.

Proof: Let S be any set of languages, and let

co-S = { $\overline{L}$ | L is in S }.

Suppose that co-S $\subseteq$ S; we'll show that S $\subseteq$ co-S, and so S = co-S.

Let L be in S. Since co-S $\subseteq$ S, $\overline{L}$ must also be in S. And then, since $\overline{L}$ is in S, L must be in co-S. So S $\subseteq$ co-S.

The Polynomial Time Hierarchy

- Σ_2^p: set of languages L for which there is a polynomial-time DTM M such that

  w is in L iff $\exists Z_1\ \forall Z_2\ M(w, Z_1, Z_2)$ accepts

  where $|Z_1| = |Z_2|$ is polynomial in $|w|$

The Polynomial Time Hierarchy

- $\sum_2^p$: set of languages L for which there is a polynomial-time DTM M such that

  w is in L iff $\exists Z_1 \ \forall Z_2 \ M(w, Z_1, Z_2)$ accepts

  where $|Z_1| = |Z_2|$ is polynomial in $|w|$

- *Example*: MIN–EQ-DNF =

  { $\langle \phi, k \rangle$: there is a DNF ϕ' of size $\leq$ k that is equivalent to DNF ϕ}.

- MIN–EQ-DNF is in $\sum_2^p$ and is not known to be in NP

The Polynomial Time Hierarchy

- Σ_2^p: set of languages L for which there is a polynomial-time DTM M such that

  w is in L iff $\exists Z_1\ \forall Z_2\ M(w, Z_1, Z_2)$ accepts

  where $|Z_1| = |Z_2|$ is polynomial in $|w|$

- Π_2^p: similar to Σ_2^p, but starts with a $\forall$ quantifier

  w is in L iff $\forall Z_1\ \exists Z_2\ M(w, Z_1, Z_2)$ accepts
- Π_k^p and Σ_k^p: generalizations for k > 2
- Polynomial time hierarchy (PH) is $\bigcup_{k>0} (\Pi_k^p \cup \Sigma_k^p)$

The Polynomial Time Hierarchy

- Σ_k SAT is complete for Σ_k^p
- Π_k SAT is complete for Π_k^p
- Note that if $\Pi_k^p \subseteq \Sigma_k^p$ then in fact $\Pi_k^p = \Sigma_k^p$

The Polynomial Time Hierarchy