Lecture 24

Baker Gill Solovay's Theorem

Baker Gill Solovay’s Theorem

Baker Gill Solovay's Theorem

Theorem (BGS75): There exist oracles A and B such that $\mathsf{P}^A = \mathsf{NP}^A$ and $\mathsf{P}^B \neq \mathsf{NP}^B$.

Baker Gill Solovay's Theorem

Theorem (BGS75): There exist oracles A and B such that $\mathsf{P}^A = \mathsf{NP}^A$ and $\mathsf{P}^B \neq \mathsf{NP}^B$.

Proof:

Baker Gill Solovay's Theorem

Theorem (BGS75): There exist oracles A and B such that $\mathsf{P}^A = \mathsf{NP}^A$ and $\mathsf{P}^B \neq \mathsf{NP}^B$.

Proof: We already know that $A = \mathit{EXPCOM}$.

Baker Gill Solovay's Theorem

Theorem (BGS75): There exist oracles A and B such that $\mathsf{P}^A = \mathsf{NP}^A$ and $\mathsf{P}^B \neq \mathsf{NP}^B$.

Proof: We already know that $A = \mathit{EXPCOM}$.

We want to find a B such that $\mathsf{P}^B \subset \mathsf{NP}^B$.

Baker Gill Solovay's Theorem

Theorem (BGS75): There exist oracles A and B such that $\mathsf{P}^A = \mathsf{NP}^A$ and $\mathsf{P}^B \neq \mathsf{NP}^B$.

Proof: We already know that $A = EXPCOM$.

We want to find a B such that $\mathsf{P}^B \subset \mathsf{NP}^B$.

For any oracle (or language) B, define L_B as:

Baker Gill Solovay's Theorem

Theorem (BGS75): There exist oracles A and B such that $\mathsf{P}^A = \mathsf{NP}^A$ and $\mathsf{P}^B \neq \mathsf{NP}^B$.

Proof: We already know that $A = EXPCOM$.

We want to find a B such that $\mathsf{P}^B \subset \mathsf{NP}^B$.

For any oracle (or language) B, define L_B as:

$$L_B = \{1^n \mid B \text{ has a string of length } n \text{ in it}\}$$

Baker Gill Solovay's Theorem

Theorem (BGS75): There exist oracles A and B such that $\mathsf{P}^A = \mathsf{NP}^A$ and $\mathsf{P}^B \neq \mathsf{NP}^B$.

Proof: We already know that $A = EXPCOM$.

We want to find a B such that $\mathsf{P}^B \subset \mathsf{NP}^B$.

For any oracle (or language) B, define L_B as:

$$L_B = \{1^n \mid B \text{ has a string of length } n \text{ in it}\}$$

Claim:

Baker Gill Solovay's Theorem

Theorem (BGS75): There exist oracles A and B such that $\mathsf{P}^A = \mathsf{NP}^A$ and $\mathsf{P}^B \neq \mathsf{NP}^B$.

Proof: We already know that $A = EXPCOM$.

We want to find a B such that $\mathsf{P}^B \subset \mathsf{NP}^B$.

For any oracle (or language) B, define L_B as:

$$L_B = \{1^n \mid B \text{ has a string of length } n \text{ in it}\}$$

Claim: $L_B \in \mathsf{NP}^B$ for any B.

Baker Gill Solovay's Theorem

Theorem (BGS75): There exist oracles A and B such that $\mathsf{P}^A = \mathsf{NP}^A$ and $\mathsf{P}^B \neq \mathsf{NP}^B$.

Proof: We already know that $A = EXPCOM$.

We want to find a B such that $\mathsf{P}^B \subset \mathsf{NP}^B$.

For any oracle (or language) B, define L_B as:

$$L_B = \{1^n \mid B \text{ has a string of length } n \text{ in it}\}$$

Claim: $L_B \in \mathsf{NP}^B$ for any B.

Proof:

Baker Gill Solovay's Theorem

Theorem (BGS75): There exist oracles A and B such that $\mathsf{P}^A = \mathsf{NP}^A$ and $\mathsf{P}^B \neq \mathsf{NP}^B$.

Proof: We already know that $A = EXPCOM$.

We want to find a B such that $\mathsf{P}^B \subset \mathsf{NP}^B$.

For any oracle (or language) B, define L_B as:

$$L_B = \{1^n \mid B \text{ has a string of length } n \text{ in it}\}$$

Claim: $L_B \in \mathsf{NP}^B$ for any B.

Proof: Oracle **NP** machine on input 1^n

Baker Gill Solovay's Theorem

Theorem (BGS75): There exist oracles A and B such that $\mathsf{P}^A = \mathsf{NP}^A$ and $\mathsf{P}^B \neq \mathsf{NP}^B$.

Proof: We already know that $A = EXPCOM$.

We want to find a B such that $\mathsf{P}^B \subset \mathsf{NP}^B$.

For any oracle (or language) B, define L_B as:

$$L_B = \{1^n \mid B \text{ has a string of length } n \text{ in it}\}$$

Claim: $L_B \in \mathsf{NP}^B$ for any B.

Proof: Oracle **NP** machine on input 1^n will guess all strings of length n

Baker Gill Solovay's Theorem

Theorem (BGS75): There exist oracles A and B such that $\mathsf{P}^A = \mathsf{NP}^A$ and $\mathsf{P}^B \neq \mathsf{NP}^B$.

Proof: We already know that $A = EXPCOM$.

We want to find a B such that $\mathsf{P}^B \subset \mathsf{NP}^B$.

For any oracle (or language) B, define L_B as:

$$L_B = \{1^n \mid B \text{ has a string of length } n \text{ in it}\}$$

Claim: $L_B \in \mathsf{NP}^B$ for any B.

Proof: Oracle **NP** machine on input 1^n will guess all strings of length n and ask

Baker Gill Solovay's Theorem

Theorem (BGS75): There exist oracles A and B such that $\mathsf{P}^A = \mathsf{NP}^A$ and $\mathsf{P}^B \neq \mathsf{NP}^B$.

Proof: We already know that $A = EXPCOM$.

We want to find a B such that $\mathsf{P}^B \subset \mathsf{NP}^B$.

For any oracle (or language) B, define L_B as:

$$L_B = \{1^n \mid B \text{ has a string of length } n \text{ in it}\}$$

Claim: $L_B \in \mathsf{NP}^B$ for any B.

Proof: Oracle **NP** machine on input 1^n will guess all strings of length n and ask oracle whether generated string belongs to B

Baker Gill Solovay's Theorem

Theorem (BGS75): There exist oracles A and B such that $\mathsf{P}^A = \mathsf{NP}^A$ and $\mathsf{P}^B \neq \mathsf{NP}^B$.

Proof: We already know that $A = \mathit{EXPCOM}$.

We want to find a B such that $\mathsf{P}^B \subset \mathsf{NP}^B$.

For any oracle (or language) B, define L_B as:

$$L_B = \{1^n \mid B \text{ has a string of length } n \text{ in it}\}$$

Claim: $L_B \in \mathsf{NP}^B$ for any B.

Proof: Oracle **NP** machine on input 1^n will guess all strings of length n and ask oracle whether generated string belongs to B and answer accordingly.

Baker Gill Solovay's Theorem

Theorem (BGS75): There exist oracles A and B such that $\mathsf{P}^A = \mathsf{NP}^A$ and $\mathsf{P}^B \neq \mathsf{NP}^B$.

Proof: We already know that $A = EXPCOM$.

We want to find a B such that $\mathsf{P}^B \subset \mathsf{NP}^B$.

For any oracle (or language) B, define L_B as:

$$L_B = \{1^n \mid B \text{ has a string of length } n \text{ in it}\}$$

Claim: $L_B \in \mathsf{NP}^B$ for any B.

Proof: Oracle **NP** machine on input 1^n will guess all strings of length n and ask oracle whether generated string belongs to B and answer accordingly.

We now want to construct a B so that L_B cannot be decided

Baker Gill Solovay's Theorem

Theorem (BGS75): There exist oracles A and B such that $\mathsf{P}^A = \mathsf{NP}^A$ and $\mathsf{P}^B \neq \mathsf{NP}^B$.

Proof: We already know that $A = EXPCOM$.

We want to find a B such that $\mathsf{P}^B \subset \mathsf{NP}^B$.

For any oracle (or language) B, define L_B as:

$$L_B = \{1^n \mid B \text{ has a string of length } n \text{ in it}\}$$

Claim: $L_B \in \mathsf{NP}^B$ for any B.

Proof: Oracle **NP** machine on input 1^n will guess all strings of length n and ask oracle whether generated string belongs to B and answer accordingly.

We now want to construct a B so that L_B cannot be decided by any polytime

Baker Gill Solovay's Theorem

Theorem (BGS75): There exist oracles A and B such that $\mathsf{P}^A = \mathsf{NP}^A$ and $\mathsf{P}^B \neq \mathsf{NP}^B$.

Proof: We already know that $A = EXPCOM$.

We want to find a B such that $\mathsf{P}^B \subset \mathsf{NP}^B$.

For any oracle (or language) B, define L_B as:

$$L_B = \{1^n \mid B \text{ has a string of length } n \text{ in it}\}$$

Claim: $L_B \in \mathsf{NP}^B$ for any B.

Proof: Oracle **NP** machine on input 1^n will guess all strings of length n and ask oracle whether generated string belongs to B and answer accordingly.

We now want to construct a B so that L_B cannot be decided by any polytime oracle DTM with access to B.

Baker Gill Solovay's Theorem

Theorem (BGS75): There exist oracles A and B such that $\mathsf{P}^A = \mathsf{NP}^A$ and $\mathsf{P}^B \neq \mathsf{NP}^B$.

Proof: We already know that $A = EXPCOM$.

We want to find a B such that $\mathsf{P}^B \subset \mathsf{NP}^B$.

For any oracle (or language) B, define L_B as:

$$L_B = \{1^n \mid B \text{ has a string of length } n \text{ in it}\}$$

Claim: $L_B \in \mathsf{NP}^B$ for any B.

Proof: Oracle **NP** machine on input 1^n will guess all strings of length n and ask oracle whether generated string belongs to B and answer accordingly.

We now want to construct a B so that L_B cannot be decided by any polytime oracle DTM with access to B.

...

Baker Gill Solovay’s Theorem

Baker Gill Solovay's Theorem

Warmup Claim:

Baker Gill Solovay's Theorem

Warmup Claim: For any polytime, oracle DTM M, there exists a language B

Baker Gill Solovay's Theorem

Warmup Claim: For any polytime, oracle DTM M, there exists a language B such that M^B does not decide L_B.

Baker Gill Solovay's Theorem

Warmup Claim: For any polytime, oracle DTM M, there exists a language B such that M^B does not decide L_B.

Proof:

Baker Gill Solovay's Theorem

Warmup Claim: For any polytime, oracle DTM M, there exists a language B such that M^B does not decide L_B.

Proof: Let $p(n)$ be M's runtime.

Baker Gill Solovay's Theorem

Warmup Claim: For any polytime, oracle DTM M, there exists a language B such that M^B does not decide L_B.

Proof: Let $p(n)$ be M's runtime. Let n be an integer such that $2^n > p(n)$.

Baker Gill Solovay's Theorem

Warmup Claim: For any polytime, oracle DTM M, there exists a language B such that M^B does not decide L_B.

Proof: Let $p(n)$ be M's runtime. Let n be an integer such that $2^n > p(n)$.

Idea: Exploit the fact that on input 1^n, M cannot query on all strings of length n.

Baker Gill Solovay's Theorem

Warmup Claim: For any polytime, oracle DTM M, there exists a language B such that M^B does not decide L_B.

Proof: Let $p(n)$ be M's runtime. Let n be an integer such that $2^n > p(n)$.

Idea: Exploit the fact that on input 1^n, M cannot query on all strings of length n.

We define B in the following way:

Baker Gill Solovay's Theorem

Warmup Claim: For any polytime, oracle DTM M, there exists a language B such that M^B does not decide L_B.

Proof: Let $p(n)$ be M's runtime. Let n be an integer such that $2^n > p(n)$.

Idea: Exploit the fact that on input 1^n, M cannot query on all strings of length n.

We define B in the following way:

- Run $M(1^n)$ and answer "no" to all the queries.

Baker Gill Solovay's Theorem

Warmup Claim: For any polytime, oracle DTM M, there exists a language B such that M^B does not decide L_B.

Proof: Let $p(n)$ be M's runtime. Let n be an integer such that $2^n > p(n)$.

Idea: Exploit the fact that on input 1^n, M cannot query on all strings of length n.

We define B in the following way:

- Run $M(1^n)$ and answer "no" to all the queries.
- Let b be the output of M

Baker Gill Solovay's Theorem

Warmup Claim: For any polytime, oracle DTM M, there exists a language B such that M^B does not decide L_B.

Proof: Let $p(n)$ be M's runtime. Let n be an integer such that $2^n > p(n)$.

Idea: Exploit the fact that on input 1^n, M cannot query on all strings of length n.

We define B in the following way:

- Run $M(1^n)$ and answer "no" to all the queries.
- Let b be the output of M and $Q = \{q_1, q_2, \ldots, \}$ be the set of queries of length n.

Baker Gill Solovay's Theorem

Warmup Claim: For any polytime, oracle DTM M, there exists a language B such that M^B does not decide L_B.

Proof: Let $p(n)$ be M's runtime. Let n be an integer such that $2^n > p(n)$.

Idea: Exploit the fact that on input 1^n, M cannot query on all strings of length n.

We define B in the following way:

- Run $M(1^n)$ and answer "no" to all the queries.
- Let b be the output of M and $Q = \{q_1, q_2, \ldots, \}$ be the set of queries of length n.
- Take any $x \in \{0,1\}^n \backslash Q$.

Baker Gill Solovay's Theorem

Warmup Claim: For any polytime, oracle DTM M, there exists a language B such that M^B does not decide L_B.

Proof: Let $p(n)$ be M's runtime. Let n be an integer such that $2^n > p(n)$.

Idea: Exploit the fact that on input 1^n, M cannot query on all strings of length n.

We define B in the following way:

- Run $M(1^n)$ and answer "no" to all the queries.
- Let b be the output of M and $Q = \{q_1, q_2, \ldots, \}$ be the set of queries of length n.
- Take any $x \in \{0,1\}^n \backslash Q$. (Such an x exists because $2^n > p(n)$)

Baker Gill Solovay's Theorem

Warmup Claim: For any polytime, oracle DTM M, there exists a language B such that M^B does not decide L_B.

Proof: Let $p(n)$ be M's runtime. Let n be an integer such that $2^n > p(n)$.

Idea: Exploit the fact that on input 1^n, M cannot query on all strings of length n.

We define B in the following way:

- Run $M(1^n)$ and answer "no" to all the queries.
- Let b be the output of M and $Q = \{q_1, q_2, \ldots, \}$ be the set of queries of length n.
- Take any $x \in \{0,1\}^n \backslash Q$. (Such an x exists because $2^n > p(n)$)
- If $b = 0$, then put x in B

Baker Gill Solovay's Theorem

Warmup Claim: For any polytime, oracle DTM M, there exists a language B such that M^B does not decide L_B.

Proof: Let $p(n)$ be M's runtime. Let n be an integer such that $2^n > p(n)$.

Idea: Exploit the fact that on input 1^n, M cannot query on all strings of length n.

We define B in the following way:

- Run $M(1^n)$ and answer "no" to all the queries.
- Let b be the output of M and $Q = \{q_1, q_2, \ldots, \}$ be the set of queries of length n.
- Take any $x \in \{0,1\}^n \backslash Q$. (Such an x exists because $2^n > p(n)$)
- If $b = 0$, then put x in B, else, keep B empty.

Baker Gill Solovay's Theorem

Warmup Claim: For any polytime, oracle DTM M, there exists a language B such that M^B does not decide L_B.

Proof: Let $p(n)$ be M's runtime. Let n be an integer such that $2^n > p(n)$.

Idea: Exploit the fact that on input 1^n, M cannot query on all strings of length n.

We define B in the following way:

- Run $M(1^n)$ and answer "no" to all the queries.
- Let b be the output of M and $Q = \{q_1, q_2, \ldots, \}$ be the set of queries of length n.
- Take any $x \in \{0,1\}^n \backslash Q$. (Such an x exists because $2^n > p(n)$)
- If $b = 0$, then put x in B, else, keep B empty.

Now, $M^B(1^n)$ will be wrong by construction of B.

Baker Gill Solovay's Theorem

Warmup Claim: For any polytime, oracle DTM M, there exists a language B such that M^B does not decide L_B.

Proof: Let $p(n)$ be M's runtime. Let n be an integer such that $2^n > p(n)$.

Idea: Exploit the fact that on input 1^n, M cannot query on all strings of length n.

We define B in the following way:

- Run $M(1^n)$ and answer "no" to all the queries.
- Let b be the output of M and $Q = \{q_1, q_2, \ldots, \}$ be the set of queries of length n.
- Take any $x \in \{0,1\}^n \setminus Q$. (Such an x exists because $2^n > p(n)$)
- If $b = 0$, then put x in B, else, keep B empty.

Now, $M^B(1^n)$ will be wrong by construction of B.

■

Baker Gill Solovay’s Theorem

Baker Gill Solovay's Theorem

Let's design a B so that

Baker Gill Solovay's Theorem

Let's design a B so that no polytime oracle DTM with access to B can decide L_B.

Baker Gill Solovay's Theorem

Let's design a B so that no polytime oracle DTM with access to B can decide L_B.

Consider a sequence of $M_1, M_2, \ldots$ of oracle DTMs with runtime $p_1(n), p_2(n), \ldots$.

Baker Gill Solovay's Theorem

Let's design a B so that no polytime oracle DTM with access to B can decide L_B.

Consider a sequence of $M_1, M_2, \ldots$ of oracle DTMs with runtime $p_1(n), p_2(n), \ldots$.

We will build B inductively.

Baker Gill Solovay's Theorem

Let's design a B so that no polytime oracle DTM with access to B can decide L_B.

Consider a sequence of $M_1, M_2, \ldots$ of oracle DTMs with runtime $p_1(n), p_2(n), \ldots$.

We will build B inductively. Let $B_0 = \varnothing, n_0 = 1, p_0 = c$.

Baker Gill Solovay's Theorem

Let's design a B so that no polytime oracle DTM with access to B can decide L_B.

Consider a sequence of $M_1, M_2, \ldots$ of oracle DTMs with runtime $p_1(n), p_2(n), \ldots$.

We will build B inductively. Let $B_0 = \emptyset, n_0 = 1, p_0 = c$.

In the ith iteration:

Baker Gill Solovay's Theorem

Let's design a B so that no polytime oracle DTM with access to B can decide L_B.

Consider a sequence of $M_1, M_2, \ldots$ of oracle DTMs with runtime $p_1(n), p_2(n), \ldots$.

We will build B inductively. Let $B_0 = \emptyset, n_0 = 1, p_0 = c$.

In the ith iteration:

- Let n_i be the smallest integer s.t. $2^{n_i} > p_i(n_i)$ and $n_i > p_j(n_j)$ for all $1 \leq j < i$.

Baker Gill Solovay's Theorem

Let's design a B so that no polytime oracle DTM with access to B can decide L_B.

Consider a sequence of $M_1, M_2, \ldots$ of oracle DTMs with runtime $p_1(n), p_2(n), \ldots$.

We will build B inductively. Let $B_0 = \varnothing$, $n_0 = 1$, $p_0 = c$.

In the ith iteration:

- Let n_i be the smallest integer s.t. $2^{n_i} > p_i(n_i)$ and $n_i > p_j(n_j)$ for all $1 \leq j < i$.
- We define B_i in the following way:

Baker Gill Solovay's Theorem

Let's design a B so that no polytime oracle DTM with access to B can decide L_B.

Consider a sequence of $M_1, M_2, \ldots$ of oracle DTMs with runtime $p_1(n), p_2(n), \ldots$.

We will build B inductively. Let $B_0 = \emptyset$, $n_0 = 1$, $p_0 = c$.

In the ith iteration:

- Let n_i be the smallest integer s.t. $2^{n_i} > p_i(n_i)$ and $n_i > p_j(n_j)$ for all $1 \leq j < i$.
- We define B_i in the following way:
  - Run $M_i(1^{n_i})$ and respond to its queries according to B_{i-1}.

Baker Gill Solovay's Theorem

Let's design a B so that no polytime oracle DTM with access to B can decide L_B.

Consider a sequence of $M_1, M_2, \ldots$ of oracle DTMs with runtime $p_1(n), p_2(n), \ldots$.

We will build B inductively. Let $B_0 = \emptyset$, $n_0 = 1$, $p_0 = c$.

In the ith iteration:

- Let n_i be the smallest integer s.t. $2^{n_i} > p_i(n_i)$ and $n_i > p_j(n_j)$ for all $1 \leq j < i$.
- We define B_i in the following way:
  - Run $M_i(1^{n_i})$ and respond to its queries according to B_{i-1}.
  - Let b be the output of M_i

Baker Gill Solovay's Theorem

Let's design a B so that no polytime oracle DTM with access to B can decide L_B.

Consider a sequence of $M_1, M_2, \ldots$ of oracle DTMs with runtime $p_1(n), p_2(n), \ldots$.

We will build B inductively. Let $B_0 = \emptyset$, $n_0 = 1$, $p_0 = c$.

In the ith iteration:

- Let n_i be the smallest integer s.t. $2^{n_i} > p_i(n_i)$ and $n_i > p_j(n_j)$ for all $1 \leq j < i$.
- We define B_i in the following way:
  - Run $M_i(1^{n_i})$ and respond to its queries according to B_{i-1}.
  - Let b be the output of M_i and $Q = \{q_1, q_2, \ldots, \}$ be the set of queries of length n_i.

Baker Gill Solovay's Theorem

Let's design a B so that no polytime oracle DTM with access to B can decide L_B.

Consider a sequence of $M_1, M_2, \ldots$ of oracle DTMs with runtime $p_1(n), p_2(n), \ldots$.

We will build B inductively. Let $B_0 = \emptyset, n_0 = 1, p_0 = c$.

In the ith iteration:

- Let n_i be the smallest integer s.t. $2^{n_i} > p_i(n_i)$ and $n_i > p_j(n_j)$ for all $1 \leq j < i$.
- We define B_i in the following way:
  - Run $M_i(1^{n_i})$ and respond to its queries according to B_{i-1}.
  - Let b be the output of M_i and $Q = \{q_1, q_2, \ldots, \}$ be the set of queries of length n_i.
  - Take any $x \in \{0,1\}^{n_i} \backslash Q$.

Baker Gill Solovay's Theorem

Let's design a B so that no polytime oracle DTM with access to B can decide L_B.

Consider a sequence of $M_1, M_2, \ldots$ of oracle DTMs with runtime $p_1(n), p_2(n), \ldots$.

We will build B inductively. Let $B_0 = \varnothing, n_0 = 1, p_0 = c$.

In the ith iteration:

- Let n_i be the smallest integer s.t. $2^{n_i} > p_i(n_i)$ and $n_i > p_j(n_j)$ for all $1 \leq j < i$.
- We define B_i in the following way:
  - Run $M_i(1^{n_i})$ and respond to its queries according to B_{i-1}.
  - Let b be the output of M_i and $Q = \{q_1, q_2, \ldots, \}$ be the set of queries of length n_i.
  - Take any $x \in \{0,1\}^{n_i} \setminus Q$. (Such an x exists because $2^{n_i} > p(n_i)$)

Baker Gill Solovay's Theorem

Let's design a B so that no polytime oracle DTM with access to B can decide L_B.

Consider a sequence of $M_1, M_2, \ldots$ of oracle DTMs with runtime $p_1(n), p_2(n), \ldots$.

We will build B inductively. Let $B_0 = \emptyset, n_0 = 1, p_0 = c$.

In the ith iteration:

- Let n_i be the smallest integer s.t. $2^{n_i} > p_i(n_i)$ and $n_i > p_j(n_j)$ for all $1 \leq j < i$.
- We define B_i in the following way:
  - Run $M_i(1^{n_i})$ and respond to its queries according to B_{i-1}.
  - Let b be the output of M_i and $Q = \{q_1, q_2, \ldots, \}$ be the set of queries of length n_i.
  - Take any $x \in \{0,1\}^{n_i} \backslash Q$. (Such an x exists because $2^{n_i} > p(n_i)$)
  - If $b = 0$, then set $B_i = B_{i-1} \cup x$

Baker Gill Solovay's Theorem

Let's design a B so that no polytime oracle DTM with access to B can decide L_B.

Consider a sequence of $M_1, M_2, \ldots$ of oracle DTMs with runtime $p_1(n), p_2(n), \ldots$.

We will build B inductively. Let $B_0 = \emptyset$, $n_0 = 1$, $p_0 = c$.

In the ith iteration:

- Let n_i be the smallest integer s.t. $2^{n_i} > p_i(n_i)$ and $n_i > p_j(n_j)$ for all $1 \leq j < i$.
- We define B_i in the following way:
  - Run $M_i(1^{n_i})$ and respond to its queries according to B_{i-1}.
  - Let b be the output of M_i and $Q = \{q_1, q_2, \ldots, \}$ be the set of queries of length n_i.
  - Take any $x \in \{0,1\}^{n_i} \backslash Q$. (Such an x exists because $2^{n_i} > p(n_i)$)
  - If $b = 0$, then set $B_i = B_{i-1} \cup x$, else, set $B_i = B_{i-1}$.

Baker Gill Solovay's Theorem

Let's design a B so that no polytime oracle DTM with access to B can decide L_B.

Consider a sequence of $M_1, M_2, \ldots$ of oracle DTMs with runtime $p_1(n), p_2(n), \ldots$.

We will build B inductively. Let $B_0 = \varnothing$, $n_0 = 1$, $p_0 = c$.

In the ith iteration:

- Let n_i be the smallest integer s.t. $2^{n_i} > p_i(n_i)$ and $n_i > p_j(n_j)$ for all $1 \leq j < i$.
- We define B_i in the following way:
  - Run $M_i(1^{n_i})$ and respond to its queries according to B_{i-1}.
  - Let b be the output of M_i and $Q = \{q_1, q_2, \ldots, \}$ be the set of queries of length n_i.
  - Take any $x \in \{0,1\}^{n_i} \setminus Q$. (Such an x exists because $2^{n_i} > p(n_i)$)
  - If $b = 0$, then set $B_i = B_{i-1} \cup x$, else, set $B_i = B_{i-1}$.

…

Baker Gill Solovay’s Theorem

Baker Gill Solovay's Theorem

We claim that L_B, where $B = \cup_i B_i$

Baker Gill Solovay's Theorem

We claim that L_B, where $B = \cup_i B_i$, can not be decided by any polytime oracle TM with

Baker Gill Solovay's Theorem

We claim that L_B, where $B = \cup_i B_i$, can not be decided by any polytime oracle TM with access to B.

Baker Gill Solovay's Theorem

We claim that L_B, where $B = \cup_i B_i$, can not be decided by any polytime oracle TM with access to B.

Suppose $\exists$ a polytime TM M_i with access to B that can decide L_B.

Baker Gill Solovay's Theorem

We claim that L_B, where $B = \cup_i B_i$, can not be decided by any polytime oracle TM with access to B.

Suppose $\exists$ a polytime TM M_i with access to B that can decide L_B.

What will be the output of $M_i(1^{n_i})$ when M_i has access to B_i?

Baker Gill Solovay's Theorem

We claim that L_B, where $B = \cup_i B_i$, can not be decided by any polytime oracle TM with access to B.

Suppose $\exists$ a polytime TM M_i with access to B that can decide L_B.

What will be the output of $M_i(1^{n_i})$ when M_i has access to B_i?

Observation: M_i on 1^{n_i} with access to B_i

Baker Gill Solovay's Theorem

We claim that L_B, where $B = \cup_i B_i$, can not be decided by any polytime oracle TM with access to B.

Suppose $\exists$ a polytime TM M_i with access to B that can decide L_B.

What will be the output of $M_i(1^{n_i})$ when M_i has access to B_i?

Observation: M_i on 1^{n_i} with access to B_i runs the same as

Baker Gill Solovay's Theorem

We claim that L_B, where $B = \cup_i B_i$, can not be decided by any polytime oracle TM with access to B.

Suppose ∃ a polytime TM M_i with access to B that can decide L_B.

What will be the output of $M_i(1^{n_i})$ when M_i has access to B_i?

Observation: M_i on 1^{n_i} with access to B_i runs the same as M_i on 1^{n_i} with access to B_{i-1}

Baker Gill Solovay's Theorem

We claim that L_B, where $B = \cup_i B_i$, can not be decided by any polytime oracle TM with access to B.

Suppose $\exists$ a polytime TM M_i with access to B that can decide L_B.

What will be the output of $M_i(1^{n_i})$ when M_i has access to B_i?

Observation: M_i on 1^{n_i} with access to B_i runs the same as M_i on 1^{n_i} with access to B_{i-1} as M_i cannot query on x by construction.

Baker Gill Solovay's Theorem

We claim that L_B, where $B = \cup_i B_i$, can not be decided by any polytime oracle TM with access to B.

Suppose $\exists$ a polytime TM M_i with access to B that can decide L_B.

What will be the output of $M_i(1^{n_i})$ when M_i has access to B_i?

Observation: M_i on 1^{n_i} with access to B_i runs the same as M_i on 1^{n_i} with access to B_{i-1} as M_i cannot query on x by construction. Now,

Baker Gill Solovay's Theorem

We claim that L_B, where $B = \cup_i B_i$, can not be decided by any polytime oracle TM with access to B.

Suppose $\exists$ a polytime TM M_i with access to B that can decide L_B.

What will be the output of $M_i(1^{n_i})$ when M_i has access to B_i?

Observation: M_i on 1^{n_i} with access to B_i runs the same as M_i on 1^{n_i} with access to B_{i-1} as M_i cannot query on x by construction. Now,

- If $M_i^{B_i}(1^{n_i}) = 1$, then B contains so string of length n_i.

Baker Gill Solovay's Theorem

We claim that L_B, where $B = \cup_i B_i$, can not be decided by any polytime oracle TM with access to B.

Suppose $\exists$ a polytime TM M_i with access to B that can decide L_B.

What will be the output of $M_i(1^{n_i})$ when M_i has access to B_i?

Observation: M_i on 1^{n_i} with access to B_i runs the same as M_i on 1^{n_i} with access to B_{i-1} as M_i cannot query on x by construction. Now,

- If $M_i^{B_i}(1^{n_i}) = 1$, then B contains so string of length n_i.
- If $M_i^{B_i}(1^{n_i}) = 0$, then B contains x, a string of length n_i.

Baker Gill Solovay's Theorem

We claim that L_B, where $B = \cup_i B_i$, can not be decided by any polytime oracle TM with access to B.

Suppose $\exists$ a polytime TM M_i with access to B that can decide L_B.

What will be the output of $M_i(1^{n_i})$ when M_i has access to B_i?

Observation: M_i on 1^{n_i} with access to B_i runs the same as M_i on 1^{n_i} with access to B_{i-1} as M_i cannot query on x by construction. Now,

- If $M_i^{B_i}(1^{n_i}) = 1$, then B contains so string of length n_i.
- If $M_i^{B_i}(1^{n_i}) = 0$, then B contains x, a string of length n_i.

Baker Gill Solovay's Theorem

We claim that L_B, where $B = \cup_i B_i$, can not be decided by any polytime oracle TM with access to B.

Suppose $\exists$ a polytime TM M_i with access to B that can decide L_B.

What will be the output of $M_i(1^{n_i})$ when M_i has access to B_i?

Observation: M_i on 1^{n_i} with access to B_i runs the same as M_i on 1^{n_i} with access to B_{i-1} as M_i cannot query on x by construction. Now,

- If $M_i^{B_i}(1^{n_i}) = 1$, then B contains so string of length n_i.
- If $M_i^{B_i}(1^{n_i}) = 0$, then B contains x, a string of length n_i.

Contradiction!

Baker Gill Solovay's Theorem

We claim that L_B, where $B = \cup_i B_i$, can not be decided by any polytime oracle TM with access to B.

Suppose $\exists$ a polytime TM M_i with access to B that can decide L_B.

What will be the output of $M_i(1^{n_i})$ when M_i has access to B_i?

Observation: M_i on 1^{n_i} with access to B_i runs the same as M_i on 1^{n_i} with access to B_{i-1} as M_i cannot query on x by construction. Now,

- If $M_i^{B_i}(1^{n_i}) = 1$, then B contains so string of length n_i.
- If $M_i^{B_i}(1^{n_i}) = 0$, then B contains x, a string of length n_i.

Contradiction!

Baker Gill Solovay's Theorem

We claim that L_B, where $B = \cup_i B_i$, can not be decided by any polytime oracle TM with access to B.

Suppose $\exists$ a polytime TM M_i with access to B that can decide L_B.

What will be the output of $M_i(1^{n_i})$ when M_i has access to B_i?

Observation: M_i on 1^{n_i} with access to B_i runs the same as M_i on 1^{n_i} with access to B_{i-1} as M_i cannot query on x by construction. Now,

- If $M_i^{B_i}(1^{n_i}) = 1$, then B contains so string of length n_i.
- If $M_i^{B_i}(1^{n_i}) = 0$, then B contains x, a string of length n_i.

Contradiction!

M_i's output on 1^{n_i} with access to be B_i and B is same

Baker Gill Solovay's Theorem

We claim that L_B, where $B = \cup_i B_i$, can not be decided by any polytime oracle TM with access to B.

Suppose $\exists$ a polytime TM M_i with access to B that can decide L_B.

What will be the output of $M_i(1^{n_i})$ when M_i has access to B_i?

Observation: M_i on 1^{n_i} with access to B_i runs the same as M_i on 1^{n_i} with access to B_{i-1} as M_i cannot query on x by construction. Now,

- If $M_i^{B_i}(1^{n_i}) = 1$, then B contains so string of length n_i.
- If $M_i^{B_i}(1^{n_i}) = 0$, then B contains x, a string of length n_i.

Contradiction!

M_i's output on 1^{n_i} with access to be B_i and B is same as M_i cannot query of strings in

Baker Gill Solovay's Theorem

We claim that L_B, where $B = \cup_i B_i$, can not be decided by any polytime oracle TM with access to B.

Suppose $\exists$ a polytime TM M_i with access to B that can decide L_B.

What will be the output of $M_i(1^{n_i})$ when M_i has access to B_i?

Observation: M_i on 1^{n_i} with access to B_i runs the same as M_i on 1^{n_i} with access to B_{i-1} as M_i cannot query on x by construction. Now,

- If $M_i^{B_i}(1^{n_i}) = 1$, then B contains so string of length n_i.
- If $M_i^{B_i}(1^{n_i}) = 0$, then B contains x, a string of length n_i.

Contradiction!

M_i's output on 1^{n_i} with access to be B_i and B is same as M_i cannot query of strings in $B \backslash B_i$ because string in $B \backslash B_i$ are of length larger than $p_i(n_i)$.

Baker Gill Solovay's Theorem

We claim that L_B, where $B = \cup_i B_i$, can not be decided by any polytime oracle TM with access to B.

Suppose $\exists$ a polytime TM M_i with access to B that can decide L_B.

What will be the output of $M_i(1^{n_i})$ when M_i has access to B_i?

Observation: M_i on 1^{n_i} with access to B_i runs the same as M_i on 1^{n_i} with access to B_{i-1} as M_i cannot query on x by construction. Now,

- If $M_i^{B_i}(1^{n_i}) = 1$, then B contains so string of length n_i.
- If $M_i^{B_i}(1^{n_i}) = 0$, then B contains x, a string of length n_i.

Contradiction!

M_i's output on 1^{n_i} with access to be B_i and B is same as M_i cannot query of strings in $B \backslash B_i$ because string in $B \backslash B_i$ are of length larger than $p_i(n_i)$. Hence, contradiction stays!

Baker Gill Solovay's Theorem

We claim that L_B, where $B = \cup_i B_i$, can not be decided by any polytime oracle TM with access to B.

Suppose $\exists$ a polytime TM M_i with access to B that can decide L_B.

What will be the output of $M_i(1^{n_i})$ when M_i has access to B_i?

Observation: M_i on 1^{n_i} with access to B_i runs the same as M_i on 1^{n_i} with access to B_{i-1} as M_i cannot query on x by construction. Now,

- If $M_i^{B_i}(1^{n_i}) = 1$, then B contains so string of length n_i.
- If $M_i^{B_i}(1^{n_i}) = 0$, then B contains x, a string of length n_i.

Contradiction!

M_i's output on 1^{n_i} with access to be B_i and B is same as M_i cannot query of strings in $B \backslash B_i$ because string in $B \backslash B_i$ are of length larger than $p_i(n_i)$. Hence, contradiction stays! $\blacksquare$