- So, we design a B s.t. $U_B \notin P^B$ by recursion.

- Let $\{M_i \mid i$ is an oracle TM description$\}$ be an enumeration of oracle TMs in the increasing order of i.

- We incrementally construct B. In the i-th stage we ensure that M_i^B <u>does not</u> decide U_B in (2^n-1) steps.

Initially, $B = \phi$.

- In the i-th stage:

We have declared only a finite number of strings in/out of B. Choose n_i to be larger than the length of those strings.

Run M_i^B on 1^{n_i} for $(2^{n_i}-1)$ steps as:

(1) If M_i queries B on strings whose status is undetermined, we declare them <u>not</u> in B.

(2) If M_i queries B on strings whose status is determined, then be consistent.

(3) If, eventually in $(2^{n_i}-1)$ steps, M_i^B accepts 1^{n_i}, then declare all strings of length n_i <u>out</u> of B.

Else, we put a string of length n_i in B that has not been queried by M_i. (There exists such a string!)

▷ At the i-th stage, M_i^B does not decide U_B in (2^n-1) steps.

- So, if $U_B \in P^B$ we can consider a large enough j s.t. M_j^B decides U_B in poly-time.

This gives a contradiction.

$\Rightarrow U_B \notin P^B$

$\Rightarrow \quad P^B \neq NP^B$.

$\square$

▷ A proof for $P \stackrel{?}{=} NP$ will be non-relativizing.

More on space complexity

Defn: • Nspace(f(n)) := $\{L \mid \exists$ NDTM M that decides L using $O(f(n))$ space$\}$.

• Npspace := $\bigcup_{c \in \mathbb{N}}$ Nspace(n^c).

• Pspace := $\bigcup_{c \in \mathbb{N}}$ Space(n^c).

• NL := Nspace$(\log n)$

• $\mathbb{L}$:= Space$(\log n)$.

- Eg. of a problem in $\mathbb{L}$?
  Addition, multiplication!

▷ $P^{Pspace} = NP^{Pspace} = Pspace$.

Proposition: (i) Dtime$(f) \subseteq$ Space$(f) \subseteq$ Nspace$(f) \subseteq$ Dtime$(2^{O(f)})$.

(ii) $P \subseteq NP \subseteq Pspace \subseteq EXP$.