Definition:

A is recognizable with B, if there is an oracle TM $\underline{\underline{M}}$ with oracle B that recognizes A.

Definition:

A is decidable with B, if there is an oracle TM $\underline{\underline{M}}$ with oracle B that decides A.

Language A " $\underbrace{\text{Turing-reduces}}_{\text{oracle reduces}}$ " to B.

$\Downarrow$

if you can solve B, you can solve A.

$$A \leq_T B.$$

Example (1): A_{TM} is decidable with $HALT_{TM}$

$$(A_{TM} \leq_T HALT_{TM})$$

On input (M, w),
  if (M, w) is in $HALT_{TM}$ then
    run $M(w)$ and output its answer
  else REJECT

$\leq_T$ versus $\leq_m$.

Theorem if $A \leq_m B$ then $A \leq_T B$.

If $A \leq_m B$ then there is a computable function $f: \Sigma^* \to \Sigma^*$, where for every w, $w \in A \iff f(w) \in B$.

To decide A on the string w, just compute $f(w)$ and "call the oracle" for B.

Theorem: $\neg HALT_{TM} \leq_T HALT_{TM}$

Theorem: $\underbrace{\neg HALT_{TM}}_{\text{Not Turing recognizable.}} \not\leq_m HALT_{TM}$

easy

for an input to $\neg HALT_{TM}$, ask oracle for $HALT_{TM}$ and reverse the answer.

→ every problem is Turing-reducible to its negation.

Some Complexity classes with oracles.

$P^B = \{ L \mid L$ can be decided by some polynomial-time TM with an oracle for $B \}$.

P^{SAT} = The class of languages decidable in polynomial time with an oracle for SAT

P^{NP} = The class of languages decidable by some polynomial time oracle TM with an oracle for some B in NP

Providing an oracle to an underlying oracle Turing m/c M (resp. a complexity class C defined by OTMs) is called a relativization of M (resp. C)

Questions

(1) Is $P^{SAT} \subseteq P^{NP}$.

Is P^{SAT} contained in P^{NP} ?

(2) Is $P^{NP} \subseteq P^{SAT}$

(3) $NP \subseteq P^{NP}$.

Note

$NP^B = \{ L \mid L$ can be decided by a polynomial-time non-deterministic TM with an oracle for $B \}$.

Answers

(1) $P^{SAT} \subseteq P^{NP}$

yes, by definition

(2) $P^{NP} \subseteq P^{SAT}$.

yes.

because, every NP can be reduced to SAT.

For every poly-time TM M with oracle $B \in NP$, we can simulate every query z to oracle B by reducing z to a formula φ in poly-time, then asking an oracle for SAT instead.

(3) $NP \subseteq P^{NP}$

yes.

Just ask the oracle for the answer

for every $L \in NP$ define an oracle TM M^L which asks the oracle if the input is in L