Problem

Show that if every NP-hard language is also PSPACE-hard, then PSPACE = NP.

Step-by-step solution

Step 1 of 1

Given statement,

Every *NP* – hard language is also $PSPACE - \text{hard}$. That means NP $\subseteq$PSPACE.

We have to show that

$PSPACE = NP$.

<u>*NP* – complete:</u> “A language *B* is *NP*- complete if it satisfies two conditions.

1. *B* is in *NP*, and

2. Every *A* in *NP* is polynomial time reducible to *B*”

If *B* satisfies condition 2, we say that it is *NP*- hard.

- From the hypothesis, Every *NP*- hard language is also *PSPACE*- hard.
- By the definition, *NP* – hard contains all of the *NP* – complete problem.
- So every *NP* – complete language is also *PSPACE*-hard. We know that *SAT* is *PSPACE*- hard.
- For any language *A* is *PSPACE,* A reduces to *SAT*.
- Assume that *A* is in *NP*.

Create a Turing Machine (TM), *M* as follows

$M =$”On input *x*

1. compute $f(x)$. The poly – time nondeterministic algorithm between *A* and *SAT*.

2. If $f(x)$ is satisfible, accept *x*.

3. Else reject.”

Claim *M* decides *A* since *x* is in *A* iff $f(x)$ is in *SAT*. Also *M* is an *NP* machine since computing *SAT* is in *NP*.

Thus, language *A* is in *PSPACE* then *A* is in *NP*

So, $PSPACE = NP$.

Comment