A = { \langle G \rangle : G \text{ is a connected undirected graph} }

This is called an **encoding**

Typically for graphs $G = (V, E)$
valid
\{ 1, 2, 3 \} \cup \{ \{ 1, 2 \}, \{ 3 \} \} \text{ invalid}
\(A = \{ \langle G \rangle : G \text{ is a connected undirected graph} \}\)

\(M = \)

“On input \(\langle G \rangle\)

1. Mark an arbitrary vertex

2. Repeat until no new vertices are marked
   - For each vertex
     - Mark it if it is connected by an edge to a marked vertex

3. If all vertices are marked
   - then **accept**
   - else **reject**
Step 1: mark first node

\{1, 2, 3\} \{\{1, 2\}, \{2, 3\}\} \sqcup \ldots
Step 2: find an unmarked node

\{ \{ 1, 2 \}, \{ 1, 2 \}, \{ 2, 3 \} \} \sqcup \ldots
find a marked node
check for \{1, 2\} in edge set
exists so mark 2 and remove highlights

\{ 1, 2, 3 \} \{ \{ 1, 2 \}, \{ 2, 3 \} \} \cup \ldots

go to beginning of Step 2
Step 2: find an unmarked node
Step 2: find an unmarked node

\{ 1, 2, 3 \} \{ \{ 1, 2 \}, \{ 2, 3 \} \} \sqcup ...
find a marked node

\{ 1, 2, 3 \} \{ \{ 1, 2 \}, \{ 2, 3 \} \} \cup ... 

check for \{1, 3\} in edge set
doesn’t exist

move highlight to next marked node
check for \{1, 3\} in edge set  
doesn’t exist  
move highlight to next marked node

\{ 1, 2, 3 \} \{ \{ 1, 2 \}, \{ 2, 3 \} \} \cup \ldots

check for \{2, 3\} in edge set
check for \{2, 3\} in edge set
exists so mark 3

\{1, 2, 3\} \{\{1, 2\}, \{2, 3\}\} \cup \ldots

no more unmarked nodes, so done
$A_{TM} = \{ \langle M, w \rangle : M \text{ is a TM that accepts input } w \} \text{ is undecidable}$
$A_{TM} = \{ \langle M, w \rangle : M \text{ is a TM that accepts input } w \}$ is **undecidable**

Proof by **contradiction**
\[ A_{TM} = \{ \langle M, w \rangle : M \text{ is a TM that accepts input } w \} \]

is \textbf{undecidable}

Suppose \( A_{TM} \) is decidable.
Then there is a TM \( H \) that decides \( A_{TM} \).

\[ H( \langle M, w \rangle ) = \begin{cases} 
\text{accept} & \quad \text{iff } M \text{ accepts } w \\
\text{reject} & \quad \text{iff } M \text{ does not accept } w 
\end{cases} \]
\( A_{TM} = \{ \langle M, w \rangle : M \text{ is a TM that accepts input } w \} \) is \textbf{undecidable}

Construct TM \( D \) that uses TM \( H \) as a subroutine:

\[ D = \]

“On input \( \langle M \rangle \) where \( M \) is a TM

1. Run TM \( H \) on \( \langle M, \langle M \rangle \rangle \)
2. Output the opposite of \( H \)
   a) If \( H \) ends in an accept state, \textbf{reject}
   b) If \( H \) ends in a reject state, \textbf{accept}”
$A_{TM} = \{ \langle M, w \rangle : M \text{ is a TM that accepts input } w \}$

is **undecidable**

\[
D(\langle M \rangle) = \begin{cases} 
\text{reject} & \text{iff } M \text{ accepts } \langle M \rangle \\
\text{accept} & \text{iff } M \text{ does not accept } \langle M \rangle 
\end{cases}
\]
$A_{TM} = \{ \langle M, w \rangle : M \text{ is a TM that accepts input } w \}$

is undecidable

$$D( \langle D \rangle ) = \begin{cases} 
\text{reject} & \text{iff } D \text{ accepts } \langle D \rangle \\
\text{accept} & \text{iff } D \text{ does not accept } \langle D \rangle 
\end{cases}$$

Contradiction!