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

This is called an **encoding**

Typically for graphs \( G = (V, E) \)
\[ A = \{ \langle G \rangle : G \text{ is a connected undirected graph} \} \]

\[ M = \text{“On input } \langle G \rangle \text{“} \]

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

Decider
\[ A_{\text{DFA}} = \{ \langle D, w \rangle : D \text{ is a DFA that accepts input } w \} \]
\[ A_{\text{NFA}} = \{ \langle N, w \rangle : N \text{ is an NFA that accepts input } w \} \]
emptiness testing

$$E_{\text{DFA}} = \{ \langle D \rangle : D \text{ is an DFA and } L(D) = \emptyset \}$$
4.1 Answer all parts for the following DFA $M$ and give reasons for your answers.

a. Is $\langle M, 0100 \rangle \in A_{DFA}$?

b. Is $\langle M, 011 \rangle \in A_{DFA}$?

c. Is $\langle M \rangle \in A_{DFA}$?

d. Is $\langle M, 0100 \rangle \in A_{REX}$?

e. Is $\langle M \rangle \in E_{DFA}$?

f. Is $\langle M, M \rangle \in E_{Q_{DFA}}$?
emptiness testing

\[ E_{DFA} = \{ \langle D \rangle : D \text{ is an DFA and } L(D) = \emptyset \} \]

Try #1

\[ M_{DFA} = \]

"On input \langle D \rangle"

1. If \( D \) has any accept states
   \textbf{reject} else \textbf{accept}"
\( A = \{ \langle G \rangle : G \text{ is connected undirected graph} \} \)

\[ M = \]

“On input \(<G>\)

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
equivalence testing

\[
\text{EQ}_{\text{DFA}} = \{ \langle D_1, D_2 \rangle : D_1, D_2 \text{ are DFAs and } L(D_1) = L(D_2) \}\]
4.1 Answer all parts for the following DFA $M$ and give reasons for your answers.

- a. Is $\langle M, 0100 \rangle \in A_{DFA}$?
- b. Is $\langle M, 011 \rangle \in A_{DFA}$?
- c. Is $\langle M \rangle \in A_{DFA}$?
- d. Is $\langle M, 0100 \rangle \in A_{Rex}$?
- e. Is $\langle M \rangle \in E_{DFA}$?
- f. Is $\langle M, M \rangle \in E_{DFA}$?
equivalence testing

$$EQ_{DFA} = \{ \langle D_1, D_2 \rangle : D_1, D_2 \text{ are DFAs and } L(D_1) = L(D_2) \}$$

$$M_{EQDFA} = \text{``On input } \langle D_1, D_2 \rangle \text{,}
\begin{align*}
1. \text{ For every } w \in \Sigma^* \\
    \text{ If } D_1, D_2 \text{ either both accept or both reject } \\
    \text{ accept else reject}
\end{align*}$$
equivalence testing

$EQ_{DFA} = \{ \langle D_1, D_2 \rangle : D_1, D_2 \text{ are DFAs and } L(D_1) = L(D_2) \}$

idea: use **emptiness testing** of the **symmetric difference** of $L(D_1)$ and $L(D_2)$

$L(C) = (L(A) \cap \overline{L(B)}) \cup (\overline{L(A)} \cap L(B))$
\[ A_{TM} = \{ \langle M, w \rangle : M \text{ is a TM that accepts input } w \} \]

\( U = \)

"On input \( \langle M, w \rangle \)

1. Simulate \( M \) on \( w \)
2. If \( M \) ends in an accept state, accept  
   else if it ends in a reject state reject"

\( L(U) = A_{TM} \)

Decider or Recognizer?
\[ A_{TM} = \{ \langle M, w \rangle : M \text{ is a TM that accepts input } w \} \text{ is undecidable} \]
$A_{TM} = \{ \langle M, w \rangle : \text{M is a TM that accepts input } w \}$ is undecidable

Proof by contradiction
\( A_{\text{TM}} = \{ \langle M, w \rangle : M \text{ is a TM that accepts input } w \} \) is undecidable

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

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