\( P \cup L(\{ou1\}^*) \) is language that accepts every possible string over the alphabet. \( P \) is a subset of \( L(\{ou1\}^*) \) and we can build a NFA that recognizes \( L \).
if \( A \) is not regular
then \( \overline{A} \) is not regular.

Suppose \( A \) is not regular and \( \overline{A} \) is regular.
so \( \overline{\overline{A}} \) is regular (by thm)
so \( A \) is regular \( \times \)

\[ \neg(p \Rightarrow q) \]
\[ \neg p \vee q \]
\[ p \wedge \neg q \]
T.M. accepts all \( s \in L \) is a recognizer for \( L \) so \( L \) is Turing recognizable

T.M. that recognizes \( L \) and halts on all input is a decider for \( L \) so \( L \) is Turing decidable
\[ x \rightarrow R \equiv x \rightarrow x, R \]
\[ 0, 1 \rightarrow R \equiv (O \rightarrow O, R) \cup (I \rightarrow I, R) \]
Jflap

- tape: 2 way \infty

- moves: L, R, S

\[
L = \{ w^\#w \mid w \in \{0, 1\}^* \}
\]

\$11 \# 11 \$

\$ \checkmark \$

\$ \checkmark \$

\$ 11 \# 11 \$

\$ \checkmark \$

\$ \checkmark \$