Exercise 1: Transition predicate abstraction
Consider the following modified version of the TPA algorithm. (Modification underlined: \( T \) composed with the abstraction of \( \rho_T \))

\[
\text{Algorithm (TPA}^{cl}\text{)}
\]

\[\begin{align*}
\text{Input:} & \quad \text{program } P = (\Sigma, T, \rho) \\
& \text{set of transition predicates } \mathcal{P} \\
& \text{abstraction } \alpha \text{ defined by } \mathcal{P} \\
\text{Output:} & \quad \text{set of abstract transitions } P^\# = \{T_1, \ldots, T_n\} \\
& \text{such that } T_1 \cup \cdots \cup T_n \text{ is a transition invariant}
\end{align*}\]

\[P^\# := \{\alpha(\rho_T) \mid \tau \in T\}\]

\[\text{repeat} \quad P^\# := P^\# \cup \{\alpha(T \circ \alpha(\rho_T)) \mid T \in P^\#, \ \tau \in T, \ \alpha(T \circ \alpha(\rho_T)) \neq \emptyset\} \]

\[\text{until no change}\]

(a) Prove or refute the following claim:

The set of abstract transitions computed by TPA^{cl} is a disjunctively well-founded transition invariant iff the set of abstract transitions computed by TPA is a disjunctively well-founded transition invariant.

(b) Think about a setting where we reapply the algorithm multiple times for the same set of transition predicates. What can be a possible advantage of TPA^{cl} over TPA?

Exercise 2: TPA with initial states
So far we considered only programs \( P = (\Sigma, T, \rho) \) where every state is an initial state.
Let us now consider programs \( P = (\Sigma, \Sigma_{\text{init}}, T, \rho) \) where only the states in \( \Sigma_{\text{init}} \subseteq \Sigma \) are initial states.

(a) Give a program \( P = (\Sigma, \Sigma_{\text{init}}, T, \rho) \) whose transition relation \( R_P \) is not well-founded, but \( R_P \) restricted to the reachable states of \( P \) is well-founded. Give an informal explanation why for each set of transition predicates \( \mathcal{P} \) the set of abstract transitions \( P^\# \) is not disjunctively well-founded.

(b) Assume you have a tool that does a reachability analysis and you have a tool that computes the TPA algorithm. Describe a termination analysis that uses both tools and can be used to show termination of your program stated in part (a).