Aufgabe 12.1. [Independent Steps] (10 points)
Prove the indep-steps lemma from the lecture (Slide 126)

\[ \langle pw \rangle ([c]) \xrightarrow{*} \langle p'w' \rangle (l') \iff \\
\exists c' l'' s_1 s_2. l' = c'l'' \land s \in s_1 \otimes s_2 \land \langle pw \rangle (\varepsilon) \xrightarrow{*} \langle p'w' \rangle (l'') \land c \xrightarrow{*} c' \]

Aufgabe 12.2. [Execution Trees for Data Races] (10 points)
Let \( P \) be a set of states, \( \Gamma \) be a stack alphabet. Moreover let \( \text{Act} := \{ R, W, \tau \} \) be a set of actions. Construct a tree automaton that describes all execution trees that have a data-race, i.e., that may simultaneously execute an \( R \) and \( W \), or two \( W \) actions.
Aufgabe 12.3. [Join] (20 bonus points)
Submission of this question on July 15

Bonus points count on your side, but not on the maximum reachable points.

Let’s extend the DPN-model by joins. We add an additional state $p_\bot$, which indicates that a thread has terminated. We assume that there are no transitions from $p_\bot$. Moreover, we add an action $\text{join} \in \text{Act}$. A transition $p\gamma \xrightarrow{\text{join}} p'\gamma'$ can only be executed if all (direct) children of the thread have terminated.

1. Translate the following program to a DPN

   ```
   p():
   spawn main;
   spawn main;
   join;
   write R;
   if (...) p();
   return;

   main():
   p()
   terminate;
   ```

2. Does the program have a data-race on `write R`? Why (not)?

3. Extend the semantics of DPNs to include joins, i.e., specify the step-relation.

4. Can we decide reachability of a configuration in DPNs with joins? (Hint: Try to find a regular constraint that characterizes execution trees that actually have a join-sensitive execution)

5. Now let’s include nested locks. Show that deciding reachability (already of a single program point) is PSPACE-hard. (Hint: Try to extend the NP-hardness result for DPNs from 3SAT to QBF)