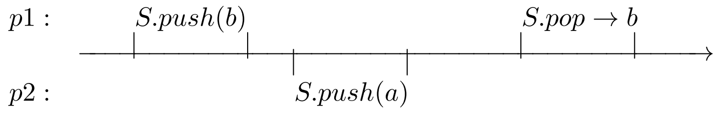
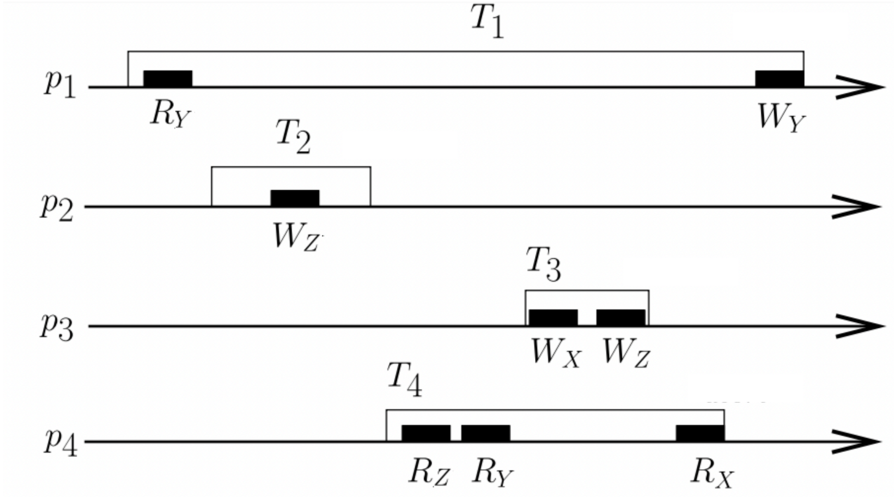
Fourth Homework of Concurrent Systems

Exercise 1. Let S be a stack and consider the history $\hat{H}$:



Is it linearizable? Is it sequentially consistent? In both cases, if the answer is positive, give a sequential history that testifies the property; if the answer is negative, say which requirement(s) of the definition cannot be satisfied.

Exercise 2. Consider the following set of transactions (where R_K and W_K respectively denote reading and writing on the atomic register $K \in \{X, Y, Z\}$):



Is this scenario opaque? Is it virtual world consistent? Provide justifications for your answers.

Exercise 3. Define $\to_{proc}$ as $\bigcup_{i=1}^{n} \to_i$, where $\to_i$ is the total order given by $\hat{H}|_{p_i}$. Show that every linearizable history is also sequentially consistent.

Exercise 4. Define a partial history $\hat{H}$ *linearizable* if $\hat{H}$ can be modified in such a way that every invocation of a pending operation is either removed or completed with a corresponding return event, and the resulting (complete) history is linearizable.

With this definition, prove the following claim:

> Let $inv[op(arg)\ on\ X\ by\ p]$ be the invocation event of a total operation that is pending in a linearizable partial history $\hat{H}$. Then, there exists a matching return event $ret[op(res)\ from\ X\ to\ p]$ such that the history $\hat{H}' = \hat{H}.ret[op(res)\ from\ X\ to\ p]$ is linearizable.