# Testing Techniques 2021 – 2022
## *Tentamen*

January 24, 2022

## 1 Model-Based Testing

Consider the labelled transition systems $q_1$, $q_2$, $q_3$, and $q_4$ in Fig. 1. These systems model *queues* with input $?in$ and outputs $!out$ and $!full$.

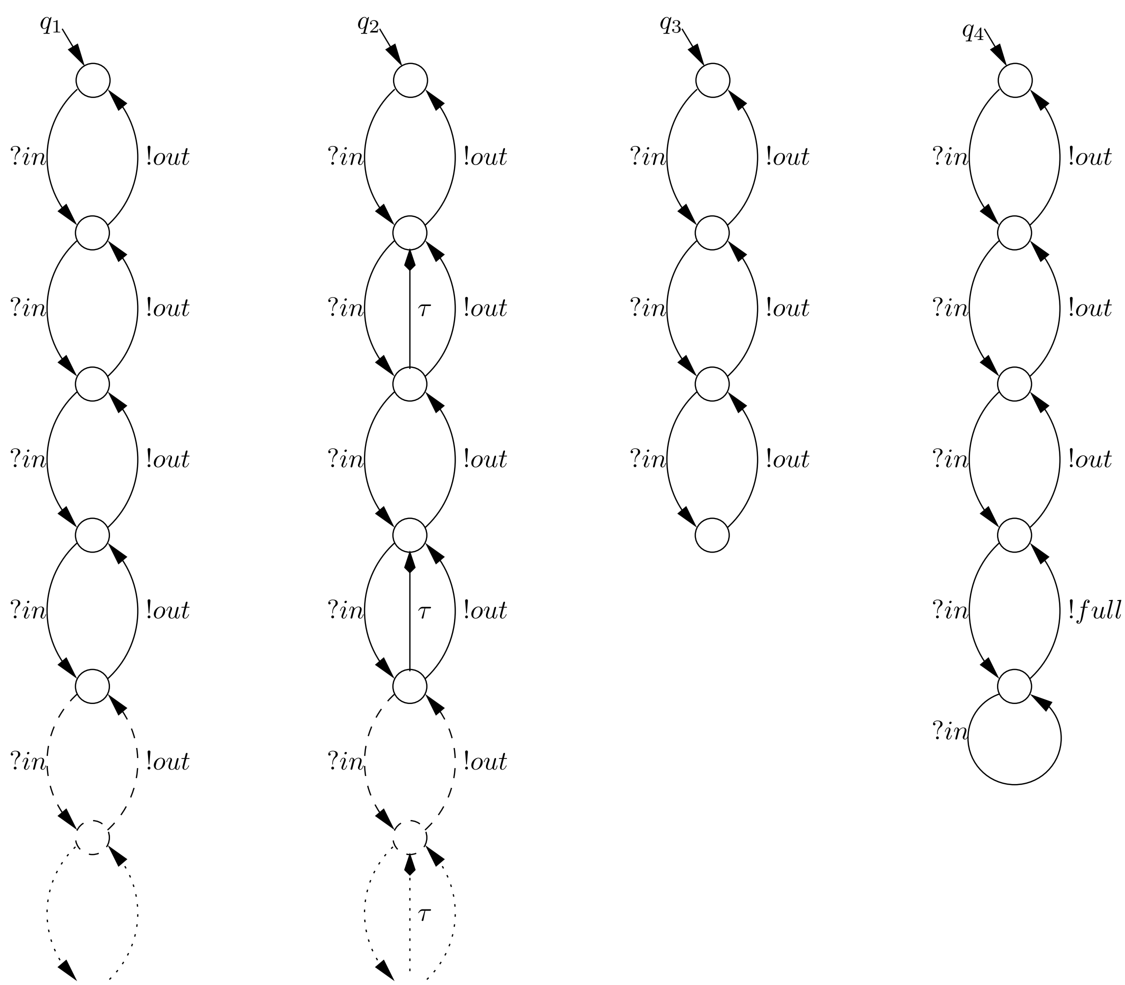


Figure 1: Four models of queues.

System $q_1$ represents an unbounded queue; the dotted lines at the bottom of $q_1$ are meant to indicate that there are infinitely many states, and that there is no bound on the number of $?in$ actions that can be performed after each other. System $q_2$ is also an unbounded queue, but it is a lossy queue: every second input may get lost. Queues $q_3$ and $q_4$ are bounded queues with capacity three, the difference being that $q_4$ gives an explicit message when the queue is full.

*a.* Which of the systems $q_1, q_2, q_3, q_4$ are *input-enabled*? Why?

*b.* Consider **ioco** as implementation relation:

$$i \,\mathbf{ioco}\, s \quad \Longleftrightarrow_{\text{def}} \quad \forall \sigma \in \mathit{Straces}(s) : \; \mathit{out}(\, i \,\mathbf{after}\, \sigma \,) \subseteq \mathit{out}(\, s \,\mathbf{after}\, \sigma \,)$$

Take $q_3$ as specification and $q_4$ as implementation. Is $q_4$ an **ioco**-correct implementation of $q_3$, i.e., does $q_4$ **ioco** $q_3$ hold? Explain.

*c.* Can an unbounded queue correctly implement a bounded queue specification, i.e., does $q_1$ **ioco** $q_3$ hold? And if the queue is lossy: does $q_2$ **ioco** $q_3$ hold? Explain.

*d.* Compare the two unbounded queues: does $q_1$ **ioco** $q_2$ or $q_2$ **ioco** $q_1$ hold?

*e.* Now compare the two unbounded queues for implementation relation **uioco**:

$$\begin{aligned}
\mathit{Utraces}(s) \quad &=_{\text{def}} \quad \{\ \sigma \in \mathit{Straces}(s) \ \mid\ \forall \sigma_1, \sigma_2 \in L_\delta^*,\ a \in L_I :\\
&\qquad\qquad \sigma = \sigma_1 {\cdot} a {\cdot} \sigma_2 \text{ implies not } s \,\mathbf{after}\, \sigma_1 \,\mathbf{refuses}\, \{a\}\ \}\\
i \,\mathbf{uioco}\, s \quad &\Longleftrightarrow_{\text{def}} \quad \forall \sigma \in \mathit{Utraces}(s) : \; \mathit{out}(\, i \,\mathbf{after}\, \sigma \,) \subseteq \mathit{out}(\, s \,\mathbf{after}\, \sigma \,)
\end{aligned}$$

What differs with respect to the previous question? Does $q_1$ **uioco** $q_2$ or $q_2$ **uioco** $q_1$ hold?

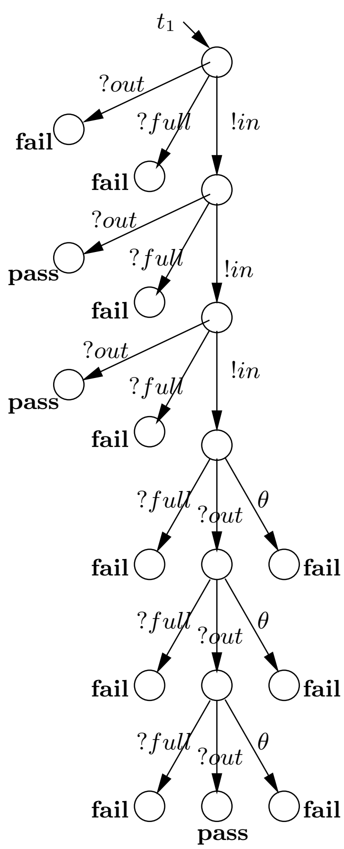


Figure 2: Test case $t_1$ for queue systems.

*f.* Fig. 2 gives a test case $t_1$. From which of the models $q_1$ and $q_2$ can this test case be generated using the **ioco**-test generation algorithm?

*g.* Give the test runs and determine the verdict of executing the test case $t_1$ on $q_2$.

*h.* Is test case $t_1$ *sound* for specification $q_1$ with respect to implementation relation **ioco**? And is it *sound* for $q_2$ with respect to **ioco**? Explain.

## 2 Labelled Transition Systems

Consider the processes $P_1$, $P_2$, and $P_3$, which are represented as labelled transition systems in Figure 3 with labelset $L = \{a, b, c, d\}$.

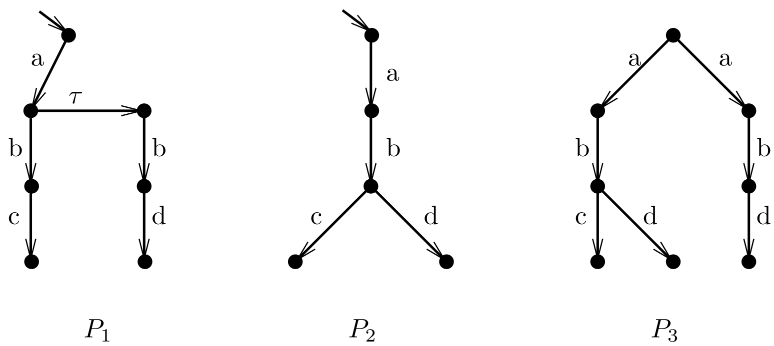


Figure 3:

*a.* Argue informally whether you consider the following pairs of processes to be equivalent or not, i.e., describe test experiments or observations that can distinguish the processes, or argue why they cannot be distinguished:

1. $P_1$ and $P_2$
2. $P_2$ and $P_3$
3. $P_1$ and $P_3$

*b.* Consider testing equivalence, that is, perform a trace of actions $\sigma$ and see which actions can then be refused. Formally:

$$p \approx_{te} q \quad \Longleftrightarrow_{\text{def}} \quad \forall \sigma \in L^*,\ \forall A \subseteq L :\ \ p \,\mathbf{after}\, \sigma \,\mathbf{refuses}\, A \ \text{ iff } \ q \,\mathbf{after}\, \sigma \,\mathbf{refuses}\, A$$

with $p \,\mathbf{after}\, \sigma \,\mathbf{refuses}\, A \quad \Longleftrightarrow_{\text{def}} \quad \exists p' :\ p \stackrel{\sigma}{\Longrightarrow} p' \ \text{ and } \ \forall a \in A \cup \{\tau\} :\ p' \stackrel{a}{\not\longrightarrow}$ .

Which of the following pairs of processes are $\approx_{te}$? Why?

1. $P_1$ and $P_2$
2. $P_2$ and $P_3$
3. $P_1$ and $P_3$

*c.* Now consider the processes as input-output transition systems with $L_I = \{a\}$ and $L_U = \{b, c, d\}$ and completed with self-loops in order to make them input-complete. Which of the following pairs of processes are **ioco**-related?

1. $P_1$ **ioco** $P_2$
2. $P_2$ **ioco** $P_3$
3. $P_3$ **ioco** $P_1$