Testing Techniques 2020 – 2021
Tentamen

January 12, 2021 – 8:30–11:30/12:00 h. – LIN 2 / HG00.058

1 Testing with ioco

Consider the specification $s \in \mathcal{LTS}(L_I, L_U)$, and the implementations $i_1, i_2 \in \mathcal{IOTS}(L_I, L_U)$ in Fig. 1, where $L = L_I \cup L_U$, with $L_I = \{?b\}$ and $L_U = \{!espr, !sugar\}$. The system s specifies a coffee machine (what else ...), this time producing different varieties of espresso. If the button $?b$ is pushed three times suffiently fast after each other, the machine just produces an *espresso*. After two times pushing the button an *espresso* with *sugar* is produced, and after having pushed $?b$ once a double *espresso* with *sugar* is produced. For a double *espresso* without *sugar* you have to push the button $?b$ again immediately after the first *espresso* has been produced, in order to "skip" the *sugar* transition.

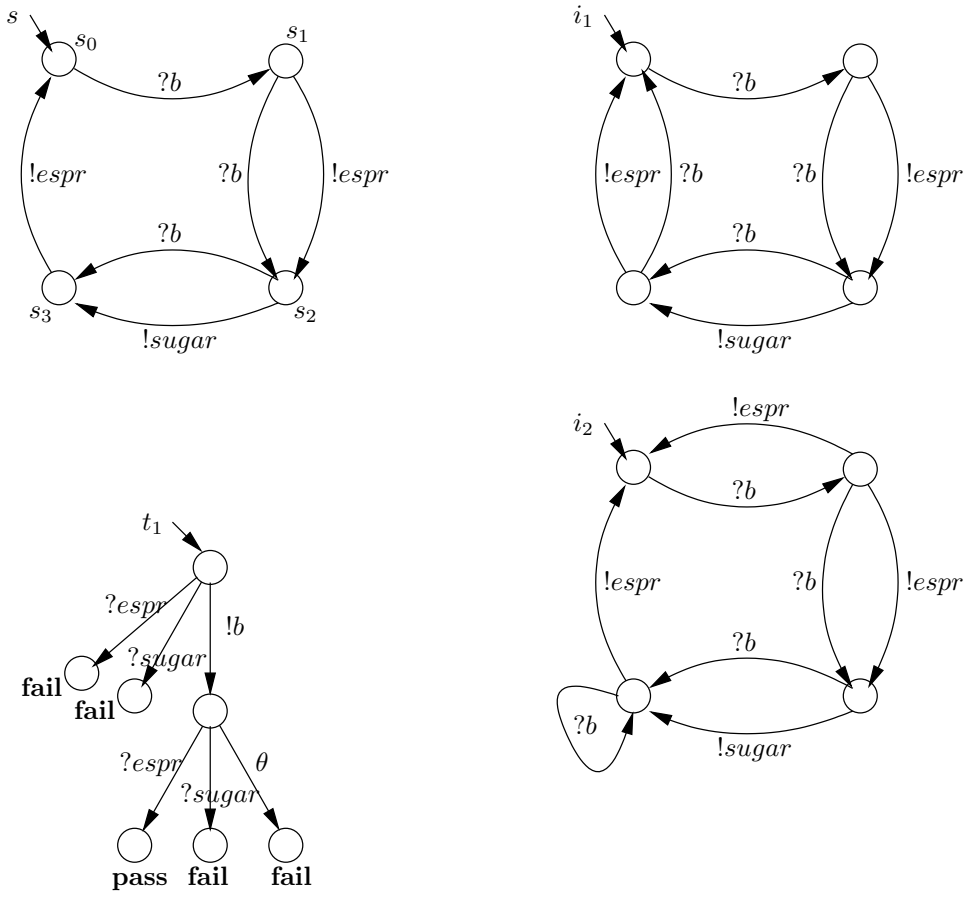

Figure 1:

a. Which states of s are *quiescent*?

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

$$i \textbf{ ioco } s \quad \Longleftrightarrow_{\text{def}} \quad \forall \sigma \in \mathit{Straces}(s) : \ \mathit{out}(\, i \textbf{ after } \sigma \,) \subseteq \mathit{out}(\, s \textbf{ after } \sigma \,) \tag{1}$$

Which of the implementations i_1, i_2 are **ioco**-correct with respect to s?

c. Use the *input-output refusal relation* (also called *repetitive quiescent trace preorder*) $\leq_{ior}$ as an implementation relation:

$$i \leq_{ior} s \quad \Longleftrightarrow_{\text{def}} \quad \forall \sigma \in (L \cup \{\delta\})^* : \ \mathit{out}(\, i \textbf{ after } \sigma \,) \subseteq \mathit{out}(\, s \textbf{ after } \sigma \,) \tag{2}$$

Which of the implementations i_1, i_2 are $\leq_{ior}$-correct with respect to s?

d. *Prove* that any $\leq_{ior}$-correct implementation is also **ioco**-correct, i.e., prove that $\leq_{ior} \subseteq$ **ioco**.

e. Figure 1 also gives a test case t_1. Give the test runs and determine the verdict of executing t_1 on i_1 and i_2.

f. A test suite T is *exhaustive* for a specification model m iff all **ioco**-incorrect implementations of m are detected by T, i.e.,

$$T \text{ is } \textit{exhaustive} \text{ for } m \quad \Longleftrightarrow_{\text{def}} \quad \forall i \in \mathcal{IOTS}(L_I, L_U) : \ i \textbf{ io}\not\textbf{o } m \ \Rightarrow \ i \textbf{ fails } T$$

Show that the test suite $\{t_1\}$ is *not exhaustive* for s of Fig. 1.
(*Hint: Combine some results of the previous questions.*)

g. Make a test case derived from s using the **ioco**-test generation algorithm that checks whether a double *espresso* without *sugar* can be obtained,

h. Obtaining a double *espresso* without *sugar* may lead to a *race condition* between a user (or tester) trying to push button $?b$ and the machine trying to produce $!sugar$. Explain how the **ioco**-test of the previous question deals with this race condition.