ability to perform sequences of observable actions. Such a sequence of observable actions is obtained from a sequence of actions under abstraction from the internal action $\tau$. If $q$ can perform the sequence of actions $a{\cdot}\tau{\cdot}\tau{\cdot}b{\cdot}c{\cdot}\tau$ $(a,b,c \in L)$, i.e., $q \xrightarrow{a\cdot\tau\cdot\tau\cdot b\cdot c\cdot\tau} q'$, then we write $q \xRightarrow{a\cdot b\cdot c} q'$ for the $\tau$-abstracted sequence of observable actions. We say that $q$ is able to perform the *trace* $a{\cdot}b{\cdot}c \in L^*$. These, and some other notations and properties are formally given in Definition 4.

**Definition 4.** *Let* $p = \langle Q, L, T, q_0\rangle$ *be a labelled transition system with* $q, q' \in Q$, $a, a_i \in L$, *and* $\sigma \in L^*$.

$$
\begin{array}{lll}
q \xRightarrow{\epsilon} q' & \Leftrightarrow_{\text{def}} & q = q' \;\text{ or }\; q \xrightarrow{\tau\cdot\ldots\cdot\tau} q' \\
q \xRightarrow{a} q' & \Leftrightarrow_{\text{def}} & \exists q_1, q_2 :\; q \xRightarrow{\epsilon} q_1 \xrightarrow{a} q_2 \xRightarrow{\epsilon} q' \\
q \xRightarrow{a_1\cdot\ldots\cdot a_n} q' & \Leftrightarrow_{\text{def}} & \exists q_0 \ldots q_n :\; q = q_0 \xRightarrow{a_1} q_1 \xRightarrow{a_2} \ldots \xRightarrow{a_n} q_n = q' \\
q \xRightarrow{\sigma} & \Leftrightarrow_{\text{def}} & \exists q' :\; q \xRightarrow{\sigma} q' \\
q \overset{\sigma}{\not\Rightarrow} & \Leftrightarrow_{\text{def}} & \textit{not}\; \exists q' : q \xRightarrow{\sigma} q'
\end{array}
$$

*Example 2.* In Figure 2:
$u_0 \xRightarrow{but\cdot liq\cdot but\cdot choc} u_0$, $v_0 \xRightarrow{but\cdot but\cdot but\cdot liq} v_0$, and $u_0 \overset{but\cdot but}{\not\Longrightarrow}$ .

In our reasoning about labelled transition systems we will not always distinguish between a transition system and its initial state. If $p = \langle Q, L, T, q_0\rangle$, we will identify the process $p$ with its initial state $q_0$, and, e.g., we write $p \xRightarrow{\sigma}$ instead of $q_0 \xRightarrow{\sigma}$. With this in mind, we give some additional definitions and notations in Definition 5, which are exemplified in Example 3.

**Definition 5.** *Let* $p$ *be a (state of a) labelled transition system, and* $\sigma \in L^*$.

1. $\mathit{init}(p) \;=_{\text{def}}\; \{\ \mu \in L \cup \{\tau\} \mid p \xrightarrow{\mu}\ \}$
2. $\mathit{traces}(p) \;=_{\text{def}}\; \{\ \sigma \in L^* \mid p \xRightarrow{\sigma}\ \}$
3. $p\;\textbf{after}\;\sigma \;=_{\text{def}}\; \{\ p' \mid p \xRightarrow{\sigma} p'\ \}$
4. $P\;\textbf{after}\;\sigma \;=_{\text{def}}\; \bigcup\ \{\ p\;\textbf{after}\;\sigma \mid p \in P\ \}$, *where* $P$ *is a set of states.*
5. $P\;\textbf{refuses}\;A \;=_{\text{def}}\; \exists p \in P,\ \forall \mu \in A \cup \{\tau\} :\ p \overset{\mu}{\not\longrightarrow}$ ,
   *where* $P$ *and* $A$ *are sets of states and labels, respectively.*
6. $\mathit{der}(p) \;=_{\text{def}}\; \{\ p' \mid \exists \sigma \in L^* :\ p \xRightarrow{\sigma} p'\ \}$
7. $p$ *has* finite behaviour *if there is a natural number* $n$ *such that all traces in* $\mathit{traces}(p)$ *have length smaller than* $n$.
8. $p$ *is* finite state *if the number of reachable states* $\mathit{der}(p)$ *is finite.*
9. $p$ *is* deterministic *if, for all* $\sigma \in L^*$, $p\;\textbf{after}\;\sigma$ *has at most one element.*
   *If* $\sigma \in \mathit{traces}(p)$, *then* $p\;\textbf{after}\;\sigma$ *may be overloaded to denote this element.*
10. $p$ *is* image finite *if, for all* $\sigma \in L^*$, $p\;\textbf{after}\;\sigma$ *is finite.*
11. $p$ *is* strongly converging *if there is no state of* $p$ *that can perform an infinite sequence of internal transitions.*
12. $\mathcal{LTS}(L)$ *is the class of all image finite and strongly converging labelled transition systems with labels in* $L$.

if and only if all possible test runs lead to the verdict **pass**. This means that each test case must be executed several times in order to explore all possible non-deterministic behaviours of the implementation, and, moreover, that a particular fairness must be assumed on implementations, i.e., it is assumed that an implementation by re-execution of a test case shows all its possible non-deterministic behaviours with that test case.

**Definition 16.** *Let $t \in \mathcal{TTS}(L_U, L_I)$ and $i \in \mathcal{IOTS}(L_I, L_U)$.*

1. Running *a test case $t$ with an implementation $i$ is expressed by the parallel operator* $\rceil\!|\ : \mathcal{TTS}(L_U, L_I) \times \mathcal{IOTS}(L_I, L_U) \to \mathcal{LTS}(L_I \cup L_U \cup \{\theta\})$ *which is defined by the following inference rules:*

$$\frac{i \xrightarrow{\tau} i'}{t\rceil\!|\, i \xrightarrow{\tau} t\rceil\!|\, i'} \qquad \frac{t \xrightarrow{a} t',\ i \xrightarrow{a} i'}{t\rceil\!|\, i \xrightarrow{a} t'\rceil\!|\, i'}\ a \in L_I \cup L_U \qquad \frac{t \xrightarrow{\theta} t',\ i \xrightarrow{\delta}}{t\rceil\!|\, i \xrightarrow{\theta} t'\rceil\!|\, i}$$

2. *A* test run *of $t$ with $i$ is a trace of $t\rceil\!|\, i$ leading to one of the states* **pass** *or* **fail** *of $t$:*

$$\sigma \textit{ is a test run of } t \textit{ and } i \quad \Leftrightarrow_{\mathrm{def}} \quad \exists i' : \ t\rceil\!|\, i \xRightarrow{\sigma} \mathbf{pass}\rceil\!|\, i' \ \text{ or } \ t\rceil\!|\, i \xRightarrow{\sigma} \mathbf{fail}\rceil\!|\, i'$$

3. *Implementation $i$* passes *test case $t$ if all test runs go to the* **pass***-state of $t$:*

$$i \ \mathbf{passes}\ t \quad \Leftrightarrow_{\mathrm{def}} \quad \forall \sigma \in L_\theta^*,\ \forall i' : \ t\rceil\!|\, i \stackrel{\sigma}{\not\Longrightarrow} \mathbf{fail}\rceil\!|\, i'$$

4. *An implementation $i$* passes *a test suite $T$ if it passes all test cases in $T$:*

$$i \ \mathbf{passes}\ T \quad \Leftrightarrow_{\mathrm{def}} \quad \forall t \in T : \ i \ \mathbf{passes}\ t$$

*If $i$ does not pass the test suite, it fails:* $i$ **fails** $T \ \Leftrightarrow_{\mathrm{def}}\ \exists t \in T : i \ \cancel{\mathbf{passes}}\ t$.

*Example 14.* Consider the test cases in Figure 7 and the implementations in Figure 4. The only test run of $t_1$ with $k_1$ is $t_1\rceil\!|\, k_1 \xRightarrow{?but\cdot!liq\cdot\theta} \mathbf{pass}\rceil\!|\, k_1'$, so $k_1$ **passes** $t_1$.
For $t_1$ with $k_2$ there are two test runs:
$t_1\rceil\!|\, k_2 \xRightarrow{?but\cdot!liq\cdot\theta} \mathbf{pass}\rceil\!|\, k_2'$, and $t_1\rceil\!|\, k_2 \xRightarrow{?but\cdot!choc} \mathbf{fail}\rceil\!|\, k_2''$, so $k_2$ **fails** $t_1$.
Also $k_3$ **fails** $t_1$: $t_1\rceil\!|\, k_3 \xRightarrow{?but\cdot!liq\cdot\theta} \mathbf{pass}\rceil\!|\, k_3'$, but also $t_1\rceil\!|\, k_3 \xRightarrow{?but\cdot\theta} \mathbf{fail}\rceil\!|\, k_3''$.
When $t_2$ is applied to $k_3$ we get:
$t_2\rceil\!|\, k_3 \xRightarrow{?but\cdot!liq\cdot?but\cdot\theta} \mathbf{pass}\rceil\!|\, k_3'$, $t_2\rceil\!|\, k_3 \xRightarrow{?but\cdot\theta\cdot?but\cdot!choc} \mathbf{fail}\rceil\!|\, k_3''$, so $k_3$ **fails** $t_2$.

### 5.2 Test Generation

Now all ingredients are there to present an algorithm to generate test cases from a labelled transition system specification, which test implementations for **ioco**-correctness. To see how such test cases may be constructed, we consider the