in $\mathcal{LTS}(L_I, L_U)\backslash\mathcal{IOTS}(L_I, L_U)$, which implies that it does not make sense to have an underspecified specification in combination with $\leq_{ior}$.

With respect to the second condition, if specifications are input enabled, i.e., **ioco** is restricted to a relation *on* $\mathcal{IOTS}(L_I, L_U)$, and there are no underspecified traces anymore, then what remains turns out to be exactly the relation $\leq_{ior}$.

Proposition 2

1. $\sigma \in \mathit{Straces}(p)$ iff $\mathit{out}(\, p\, \textbf{after}\, \sigma\,) \neq \emptyset$
2. *If* $i, s \in \mathcal{IOTS}(L_I, L_U)$ *then* i **ioco** s iff $i \leq_{ior} s$
3. *If* $p, q \in \mathcal{IOTS}(L_I, L_U)$, *and* $s \in \mathcal{LTS}(L_I, L_U)$ *then* p **ioco** q and q **ioco** s imply p **ioco** s.
4. **ioco** *is a preorder on* $\mathcal{IOTS}(L_I, L_U)$.

Underspecified traces and uioco. Another relation in the family $\textbf{ioco}_{\mathcal{F}}$ is **uioco**. For the rationale for **uioco** consider r in Figure 2 as a specification with $L_I = \{?\mathit{but}\}$ and $L_U = \{!\mathit{liq}, !\mathit{choc}\}$. Since r is not input enabled, it is a partial specification. For example, $?\mathit{but}\cdot?\mathit{but}\cdot?\mathit{but}$ is an underspecified trace, and any behaviour is allowed after it. On the other hand, $?\mathit{but}$ is clearly specified; the allowed outputs after it are $!\mathit{liq}$ and δ. For the trace $?\mathit{but}\cdot?\mathit{but}$ the situation is less clear. According to **ioco** the expected output after $?\mathit{but}\cdot?\mathit{but}$ is $\mathit{out}(\, r\, \textbf{after}\, ?\mathit{but}\cdot?\mathit{but}\,) = \{!\mathit{choc}\}$. But suppose that in the first $?\mathit{but}$-transition r moves non-deterministically to state r_1 (the left branch) then one might argue that the second $?\mathit{but}$-transition is underspecified, and that, consequently, any possible behaviour is allowed in an implementation. This is exactly where **ioco** and **uioco** differ: **ioco** postulates that $?\mathit{but}\cdot?\mathit{but}$ is not an underspecified trace, because there exists a state where it is specified, whereas **uioco** states that $?\mathit{but}\cdot?\mathit{but}$ is underspecified, because there exists a state where it is underspecified.

Formally, **ioco** quantifies over $\mathcal{F} = \mathit{Straces}(s)$, which are all possible suspension traces of the specification s. The relation **uioco** quantifies over $\mathcal{F} = \mathit{Utraces}(s) \subseteq \mathit{Straces}(s)$, which are the suspension traces without the possibly underspecified traces, i.e., see Definition 15.1, all suspension traces σ of s for which it is *not* possible that a prefix σ_1 of σ ($\sigma = \sigma_1\cdot a\cdot\sigma_2$) leads to a state of s where the remainder $a\cdot\sigma_2$ of σ is underspecified, that is, a is refused.

An alternative characterization of **uioco** can be given by transforming a partial specification into an input enabled one with demonic completion using the chaos process χ, as explained in Example 5. In this way the specification makes explicit that after an underspecified trace anything is allowed.

Definition 15. *Let* $i \in \mathcal{IOTS}(L_I, L_U)$, *and* $s \in \mathcal{LTS}(L_I, L_U)$.

1. $\mathit{Utraces}(s) \;=_{\mathrm{def}}\; \{\; \sigma \in \mathit{Straces}(s) \;\mid\; \forall \sigma_1, \sigma_2 \in L_\delta^*,\; a \in L_I :$
   $\sigma = \sigma_1\cdot a\cdot\sigma_2 \text{ implies not } s\, \textbf{after}\, \sigma_1\, \textbf{refuses}\, \{a\} \;\}$
2. i **uioco** $s \;\;\Leftrightarrow_{\mathrm{def}}\;\; i\; \textbf{ioco}_{\mathit{Utraces}(s)}\; s$

Example 12. Because $\mathit{Utraces}(s) \subseteq \mathit{Straces}(s)$ it is clear (proposition 1.2) that **uioco** is not stronger than **ioco**. That it is strictly weaker follows from the following example. Take r in Figure 2 as (partial) specification, and consider