*Technical Report*
*Lehrstuhl für Regelungstechnik*
*Friedrich-Alexander-Universität Erlangen-Nürnberg*
*Cauerstraße 7, D-91058 Erlangen*

# Technical details regarding compositional verification with event priorities

Yiheng Tang, Thomas Moor, April 24th, 2022

**Abstract:** In this study, we address the verification of non-blockingness for modular discrete-event systems, i.e., discrete-event systems that are composed from component models by synchronous composition. Specifically, we extend the approach of compositional verification as poposed by Flordal and Malik (2009) and Pilbrow and Malik (2015) to prioritised events. This report is meant as a supplement to a manuscript submitted to WODES 2022 in that it contains various technical proofs omitted due to page restrictions. In turn, we do neither repeat the introduction, nor the application examples in this document. [1]

## 1 Preliminaries

We recall some common notation regarding finite automata as relevant for the present paper.

An *alphabet* $A$ is a finite set of symbols, also referred to as *events*. Given an alphabet $A$, its *Kleene closure* $A^*$ denotes the set of all finite strings, i.e., sequences of events. By convention, $A^*$ includes the empty string $\epsilon \notin A$. The *concatenation* of two strings $s, t \in A^*$ is written $st \in A^*$. We say that $s$ is a *prefix* of $r$ if there exists $t$ such that $st = r$. This is denoted $s \leq r$. Given $r,\ t \in A^*$, there can exist at most one $s \in A^*$ which satisfies $st = r$. If such $s$ exists, it is denoted $s = r/t$.

A *non-deterministic finite automaton over* $A$ is a tuple $G := \langle Q, A, \rightarrow, Q^\circ \rangle$ with

[1] In this revision, we fixed some annoying typos in the original report, dated March 29th, 2022

the finite *state set* $Q$, the *transition relation* $\rightarrow \subseteq Q \times A \times Q$, and the set of *initial states* $Q^0 \subseteq Q$. Using infix form, we write $x \overset{\alpha}{\rightarrow} y$ or $x \overset{\alpha}{\not\rightarrow} y$ whenever $(x, \alpha, y) \in\rightarrow$ or $(x, s, y) \notin \rightarrow$, respectively. Throughout this paper, when using the infix form for a relation, left out parameters are interpreted as existential quantification, e.g., the expression $(x \overset{\alpha}{\rightarrow})$ evaluates true if and only if $(\exists y: \ x \overset{\alpha}{\rightarrow} y)$. For a state $x$, the set of *enabled events* is denoted $G(x) := \{\alpha \in A \,|\, x \overset{\alpha}{\rightarrow}\}$. A sequence of states related via transitions is referred to as a *trace*, written $x_0 \xrightarrow{\alpha_1} x_1 \xrightarrow{\alpha_2} x_2 \xrightarrow{\alpha_3} \cdots \xrightarrow{\alpha_k} x_k$, or, when the intermediate states are regarded not relevant, more concisely $x_0 \xrightarrow{\alpha_1}\xrightarrow{\alpha_2}\xrightarrow{\alpha_3} \cdots \xrightarrow{\alpha_k} x_k$ or $x \overset{s}{\rightarrow} y$ with $x = x_0$, $s = \alpha_1\alpha_2\cdots\alpha_k$ and $y = x_k$. This effectively extends the transition relation to string-valued labels in the common way. Here, we stipulate $x \overset{\epsilon}{\rightarrow} x$ for all $x \in Q$. For further notational convenience, we write $X \overset{s}{\rightarrow} Y$ with $X, Y \subseteq Q$ if there exist $x \in X$ and $y \in Y$ such that $x \overset{s}{\rightarrow} y$. Likewise, $X \overset{s}{\rightarrow}$ and $G \overset{s}{\rightarrow}$ are short forms for $X \overset{s}{\rightarrow} Q$ and $Q^0 \overset{s}{\rightarrow} Q$, respectively. We say $x \in Q$ is *reachable* if $G \overset{s}{\rightarrow} x$.

Regarding termination, we consider the distinguished *termination event* $\omega \in A$ and require the existence of a unique terminal state $x^{\mathrm{T}} \in Q$ with the properties (i) for all $x \overset{\alpha}{\rightarrow} y$ we have that $y = x^{\mathrm{T}}$ if and only if $\alpha = \omega$, and (ii) $x^{\mathrm{T}} \overset{\omega}{\rightarrow} x^{\mathrm{T}}$. In graphical representations, predecessors of $x^{\mathrm{T}}$ are depicted as full black circles and $\omega$-transitions are omitted. A state $x \in Q$ is *co-reachable* if it can be continued to attain $x^{\mathrm{T}}$, i.e., if there exists $s \in A^*$ such that $x \overset{s}{\rightarrow} x^{\mathrm{T}}$. The latter condition is equivalent to $x \xrightarrow{s\omega}$. An automaton $G$ is *non-blocking* if all reachable states are co-reachable. Provided that the state count of $G$ is not too high, non-blockingness can be verified by enumeration based methods, i.e., by explicitly computing the sets of reachable and co-reachable states, respectively.

## 2 Prioritised Events and Compositional Verification

Consider a *universe of events* $U$ together with a map $\mathsf{prio}\colon U \rightarrow \mathbb{N}$ to assign a priority to each individual event. From now on, we will implicitly assume $A \subseteq U$ for any alphabet relevant for our study. Note that we read priorities as ordinal numbers, i.e., $1 \in \mathbb{N}$ for *first priority*, $2 \in \mathbb{N}$ for *second priority*, and so on: the lower the number, the higher the priority, with the highest priority is stets 1. For notational convenience, we introduce the following short forms regarding priorities for an alphabet $A \subseteq U$ and an automaton $G$, respectively:

(a) events of priority higher than $n \in \mathbb{N}$
$A^{<n} := \{\alpha \in A \,|\, \mathsf{prio}(\alpha) < n\}$ ;

(b) events of priority higher than $\mathsf{prio}(\alpha)$ for $\alpha \in U$
$A^{<\alpha} := A^{<\mathsf{prio}(\alpha)}$;

(c) lowest priority within $A$
$\mathsf{lo}(A) := \mathsf{max}\{\mathsf{prio}(\alpha) \,|\, \alpha \in A\}$;

(d) enabled events at state $x$ with priority above $n \in \mathbb{N}$
$G^{<n}(x) := G(x) \cap U^{<n}$;

We shall now formally represent the behavioural restriction caused by event priorities imposed on an automaton $G = \langle Q, A, \rightarrow, Q^{\circ} \rangle$. In any state $x$, if some event $\alpha$ is enabled, it preempts any transition labeled by an event $\alpha'$ with lower priority, i.e., with $\mathsf{prio}(\alpha) < \mathsf{prio}(\alpha')$. The following shaping operator removes the affected transitions.

**Definition 1.** Given an automaton $G = \langle Q, A, \rightarrow, Q^{\circ} \rangle$, the *shaping operator* $S(\cdot)$ is defined by $\mathcal{S}(G) := \langle Q, \Sigma, \rightarrow^{\mathcal{S}}, Q^{\circ} \rangle$ where $x \overset{\alpha}{\rightarrow}{}^{\mathcal{S}} y$if and only if $x \overset{\alpha}{\rightarrow} y$ and $G^{<\alpha}(x) = \emptyset$. □

With this definition, $S(G)$ represents the behaviour of $G$ with prioritised events as specified by the map $\mathsf{prio}\colon U \rightarrow \mathbb{N}$. It should be noted that shaping can turn a blocking automaton into a non-blocking one and vice versa. Hence, to verify non-blockingness of a system with event priorities, we may first set up $G$, second apply $\mathcal{S}(\cdot)$ and finally perform a reachability analysis, e.g. on enumeration basis.

We now turn to a variation of the common synchronous composition in order to address modular systems with event priorities. Technically, we refer to a disjoint union composition $U = \Sigma \,\dot{\cup}\, \Upsilon$ of our universe of events, with $\Sigma$ the *regular events* and $\Upsilon$ the *silent events* $\Upsilon$. The latter are not subject to synchronisation. Since, on the other hand, termination is meant to be synchronous, we have $\omega \in \Sigma$. Moreover, it is assumed that for each regular event $\sigma \in \Sigma$ there exists a unique silent event $\tau \in \Upsilon$ with matching priority and this event is denoted $\tau =: \mathsf{hide}(\sigma)$; i.e., we have $\mathsf{hide}\colon \Sigma \rightarrow \Upsilon$ with $\mathsf{prio}(\mathsf{hide}(\sigma)) = \mathsf{prio}(\sigma)$. In graphical representations, we use the convention $\tau_{(n)} := \mathsf{hide}(\sigma)$ for $\sigma \in \Sigma$ with $\mathsf{prio}(\sigma) = n$. The decomposition $U = \Sigma \,\dot{\cup}\, \Upsilon$ and the semantics to be introduced in the sequel are seen as a generalisation of a *single distinguished silent event* $\Upsilon = \{\tau\}$, as commonly used in the context of compositional verification in the absence of event priorities; see e.g. Flordal and Malik (2009); Milner (1989).

Before proceeding with our discussion, we introduce additional notational conventions for a concise reference to the partition into silent and regular events:

(e) the natural projection denoted $\mathsf{p}\colon U^* \rightarrow \Sigma^*$ removes silent events from strings in $U^*$, see e.g., Cassandras and Lafortune (2008) for a formal definition;

(f) the abstract transition relation $\Rightarrow \subseteq Q \times \Sigma^* \times Q$, defined by $x \overset{s}{\Rightarrow} y$ for $s \in \Sigma^*$ if and only if there exists some $s' \in U^*$ such that $\mathsf{p}(s') = s$ and $x \overset{s}{\rightarrow} y$;

(g) we may omit explicit intermediate states, e.g., we write $x \overset{s}{\rightarrow}\overset{t}{\Rightarrow} y$ as a short form for the existence of $z \in Q$ such that $x \overset{s}{\rightarrow} z$ and $z \overset{t}{\Rightarrow} y$;

(h) a trace is *silent* if all its event labels belong to $\Upsilon$;

(i) enabled silent events (with priority above $n \in \mathbb{N}$)
$G_{\text{slnt}}(x) := G(x) \cap \Upsilon$; $G^{<n}_{\text{slnt}}(x) := G^{<n}(x) \cap \Upsilon$;

(j) enabled regular events (with priority above $n \in \mathbb{N}$)
$G_{\text{rglr}}(x) := G(x) - \Upsilon$; $G^{<n}_{\text{rglr}}(x) := G^{<n}(x) - \Upsilon$.

Considering the composition of two specific automata over alphabets $A_1$ and $A_2$, respectively, $A_1 \cap A_2 \cap \Sigma$ are called the *shared events*, while all other events from $A_1 \cup A_2$ are *private events*. By the following definition, the composition of two automata will synchronise the execution of shared events while allowing private events to be executed independently.

**Definition 2.** Given two automata $G_1 = \langle Q_1, A_1, \rightarrow_1, Q^0_1 \rangle$ and $G_2 = \langle Q_2, A_2, \rightarrow_2, Q^0_2 \rangle$, their synchronous composition is defined by

$$G_1 \parallel G_2 := \langle Q_1 \times Q_2, A_1 \cup A_2, \rightarrow, Q^0_1 \times Q^0_2 \rangle$$

where $(x_1, x_2) \xrightarrow{\alpha} (y_1, y_2)$ if and only if one of the following three conditions is satisfied:

$$\alpha \in (A_1 \cap A_2) - \Upsilon, x_1 \xrightarrow{\alpha}_1 y_1, \text{ and } x_2 \xrightarrow{\alpha}_2 y_2\,; \tag{1}$$

$$\alpha \in (A_1 - A_2) \cup \Upsilon, x_1 \xrightarrow{\alpha}_1 y_1, \text{ and } x_2 = y_2\,; \tag{2}$$

$$\alpha \in (A_2 - A_1) \cup \Upsilon, x_1 = y_1, \text{ and } x_2 \xrightarrow{\alpha}_2 y_2\,. \tag{3}$$

We say that the transition $(x_1, x_2) \xrightarrow{\alpha} (y_1, y_2)$ is *driven by* $G_1$ if $x_1 \xrightarrow{\alpha}_1 y_1$, or, *driven by* $G_2$ if $x_2 \xrightarrow{\alpha}_1 y_2$. □

Now consider again a modular system $M = G_1 \parallel G_2 \parallel \cdots \parallel G_n$, however, with event priorities as defined above. Here, we would like to verify non-blockingness of $\mathcal{S}(M)$. In other words, we consider event priority as having a global effect on $M$, e.g., a high priority event in one component is meant to preempt lower-priority events in other components.

**Definition 3.** A family of automata $(G_i)_{1 \leq i \leq n}$ is *non-conflicting w.r.t. prioritised events* if $\mathcal{S}(G_1 \parallel G_2 \parallel \cdots \parallel G_n)$ is non-blocking. □

For the scope of the present paper, the above property is also concisely referred to as *non-conflicting* and it is precisely this property, that we seek to verify in an efficient manner. If it was that the shaping operator distributed over the synchronous composition, we could utilize exactly the same abstraction methods as those established for the situation without event priorities. Unfortunately this is not the case and, for our situation, a suitable notion of conflict equivalence will need to explicitly refer to $\mathcal{S}(\cdot)$. Since the synchronous composition is commutative, we focus attention without loss of generality on an abstraction of $G_1$. Technically, we consider the situation of

$$\mathcal{S}(\underbrace{G_1}_{:=G} \parallel \underbrace{G_2 \parallel \cdots \parallel G_k}_{:=H}), \tag{4}$$

and ask for an abstraction $G'$ of $G$ such that $\mathcal{S}(G' \parallel H)$ is non-blocking if and only if $\mathcal{S}(G \parallel H)$ is so.

A first and rather simplistic candidate for a suitable abstraction is to obtain $G'$ from $G$ by re-labeling any transition with a private but regular event $\sigma \in \Sigma$ by its silent counterpart $\mathsf{hide}(\sigma)$. This substitution is referred to as *hiding of private events*. It is immediate from Definitions 1 and 2 that this abstraction does not affect blockingness in the shaped product with one and the same automaton $H$. Thus, from now on we will assume without loss of generality that all private events of $G$ in $G \| H$ are silent.

A second and likewise simple candidate for a suitable abstraction is to obtain $G'$ from $G$ by shaping w.r.t. silent events only. Given $G = \langle Q, A, \rightarrow, Q^\circ \rangle$, we define the $\Upsilon$-*shaping operator* by $\mathcal{S}_\Upsilon(G) := \langle Q, \Sigma, \rightarrow^{\mathcal{S}_\Upsilon}, Q^\circ \rangle$ where $x \overset{\alpha}{\rightarrow}{}^{\mathcal{S}_\Upsilon} y$ if $x \overset{\alpha}{\rightarrow} y$ and $G^{<\alpha}_{\mathsf{slnt}}(x) = \emptyset$. In other words, $\mathcal{S}_\Upsilon(\cdot)$ discards all transitions which are preempted by a silent transition. As an immediate consequence from Definitions 1 and 2 we obtain $\mathcal{S}(G' \parallel H) = \mathcal{S}(G \parallel H)$ for the abstraction $G' = \mathcal{S}_\Upsilon(G)$. In particular, this abstraction does not affect blockingness in the shaped product with any automaton $H$. Thus, from now on we will assume without loss of generality that $G$ has is $\Upsilon$-shaped, i.e., that $G = \mathcal{S}_\Upsilon(G)$.

Most relevant for practical purposes, the two abstraction rules identified so far can be applied without the potentially intractable evaluation of the transition relation of $H = G_2 \parallel \cdots \parallel G_k$. This concept is made explicit in the following formal definition of *conflict equivalence w.r.t. event priorities*, which, as in e.g. Flordal and Malik (2009); Mohajerani et al. (2014), is inspired by test-theory; see e.g. Nicola and Hennessy (1984).

**Definition 4.** Two automata $G$ and $G'$ are *conflict equivalent w.r.t. prioritised events*, denoted $G \simeq_{\mathcal{S}} G'$, if for any *test-automaton* $T$, $G$ and $T$ are non-conflicting w.r.t. prioritised events if and only if $G'$ and $T$ are non-conflicting w.r.t. prioritised events. □

For the scope of the present paper, the above property is also concisely referred to as *conflict equivalence*.

*Conventions.* Whenever discussing the product $G \parallel T$, we assume implicitly and without loss of generality that all private events are silent and that $G$ is $\Upsilon$-shaped by suitable pre-processing. For the sake of a concise notation, we indicate states from $G$ with a subscript $(\cdot)_G$ and states from $T$ with a subscript $(\cdot)_T$ and assume all state sets to be disjoint. In consequence, we can at most instances omit the respective subscripts for transition relations, since e.g. $x_G \overset{\alpha}{\to} y_G$ implies the transition to be in $G$. □

## 3 Abstraction rules based on Prioritised weak bisimulation

A generic approach to obtain an abstraction with reduced state count of an automaton $G = \langle Q, A, \to, Q^\circ \rangle$ is to consider an equivalence relation $\sim \subseteq Q \times Q$ on $Q$ and to merge states per equivalence class $[x] := \{x' \in Q \,|\, (x, x') \in \sim\}$ to obtain the so called *quotient automaton*. For our situation with prioritised silent events, an adaptation that addresses silent live-locks turns out useful.

**Definition 5.** Given an $\Upsilon$-shaped automaton $G = \langle Q, A, \to, Q^\circ \rangle$, an *n-live-lock* in $G$ is a silent trace

$$x_0 \xrightarrow{\tau_1} x_1 \xrightarrow{\tau_2} \cdots \xrightarrow{\tau_k} x_k = x_0 \tag{5}$$

where $k \geq 1$, $\mathsf{lo}(\{\tau_1, \cdots, \tau_k\}) = n$ and for any $i \in \{1, \cdots, k\}$, $x \in Q$ and $\tau \in \Upsilon$, $x_i \overset{\tau}{\to} x$ implies that there exists some $j \in \{1, \cdots, k\}$ so that $x = x_j$. □

We use the short hand $\alpha$-live-lock to denote $\mathsf{prio}(\alpha)$-live-lock for some $\alpha \in A$. Note that due to event priority, live-locks may indefinitely *trap* other automata under synchronisation, i.e. when in an $n$-live-lock of some automaton $G$, a synchronised automaton $T$ can never execute a silent event $\tau$ with $\mathsf{prio}(\tau) > n$.

With this notion of $n$-live-locks, we define the *quotient automata* as follows.

**Definition 6.** Given an $\Upsilon$-shaped automaton $G = \langle Q, A, \to, Q^\circ \rangle$ and an equivalence relation $\sim \subseteq Q \times Q$, the *quotient automaton* $G/\!\sim$ of $G$ w.r.t. $\sim$ is defined by $G/\!\sim\; := \langle Q/\!\sim, \Sigma, \to_\sim, \tilde{Q}^\circ, M \rangle$ where $Q/\!\sim\; = \{[x] \,|\, x \in Q\}$, $\tilde{Q}^\circ = \{[x^\circ] \,|\, x^\circ \in Q^\circ\}$ and

$$\begin{aligned}\to_\sim = \{[x] \overset{\alpha}{\to} [y] \,|\, x \overset{\alpha}{\to} y\}\\ - \{[x] \overset{\tau}{\to} [x] \,|\, \tau \in \Upsilon \text{ and not all states of a } \tau\text{-live-lock are in } [x]\}\end{aligned}$$ □

Comparing with the conventional quotient automata construction, our definition avoids introducing inexistent live-locks while the existent live-locks are still preserved. This potentially renders the trapping power before and after abstraction consistent.

**Lemma 7.** Given a wf automaton $G = \langle Q, A, \to, Q^\circ \rangle$ and an equivalence relation $\sim \subseteq Q \times Q$. If $[x] \overset{\alpha}{\to}_\sim [y]$ in $G/\!\sim$ for some $x, y \in Q$ and $\alpha \in \Sigma_\Upsilon$, then there exist $x' \in [x]$, $y' \in [y]$ so that $x' \overset{\alpha}{\to} y'$ in $G$. □

Based on the conventional process algebra CCS (Milner (1989)), Lüttgen (1998) introduced the variant $\mathrm{CCS}^{\mathrm{ch}}$ to model concurrent systems with global event priority. In fact, the semantics inferred by a shaped automaton in our framework are quite similar to the operational semantics of $\mathrm{CCS}^{\mathrm{ch}}$. By extending the well-known *weak bisimulation* from CCS, Lüttgen (1998) defines the *prioritised weak bisimulation* (PWB) as a reasoning framework in $\mathrm{CCS}^{\mathrm{ch}}$. Following the convention in Lüttgen (1998), we distinguish certain types of transitions in order to give a definition of PWB for the context of the present paper.

**Definition 8.** Given an $\Upsilon$-shaped automaton $G = \langle Q, A, \rightarrow, Q^\circ \rangle$, define the following extended transition relations for $\Delta \subseteq \Sigma$:

(T1) $\xrightarrow[\Delta:n]{} \subseteq Q \times A \times Q$: $x \xrightarrow[\Delta:n]{\alpha} y$ if and only if $x \xrightarrow{\alpha} y$ and $G^{<n}_{\mathrm{rglr}}(x) \subseteq \Delta$;

(T2) $\xRightarrow[\Delta:n]{} \subseteq Q \times \{\epsilon\} \times Q$: $x \xRightarrow[\Delta:n]{\epsilon} y$ if and only if $x \xrightarrow[\Delta:n]{\tau_1}\xrightarrow[\Delta:n]{\tau_2} \cdots \xrightarrow[\Delta:n]{\tau_k} y$, $k \geq 0$ and $\tau_1 \cdots \tau_k \in (\Upsilon^{<(n+1)})^*$. □

**Definition 9.** Let $G = \langle Q, A, \rightarrow, Q^\circ \rangle$ be an $\Upsilon$-shaped automaton. A symmetric relation $\approx \subseteq Q \times Q$ is a *prioritised weak bisimulation on G* (PWB) if for any $x, x' \in Q$ so that $x \approx x'$, the following hold:

(P1) If $G^{<n}_{\mathrm{slnt}}(x) = \emptyset$ for some $n \geq 0$, then there exists $y'$ so that $x \approx y'$, $G^{<n}_{\mathrm{slnt}}(y') = \emptyset$, $G^{<n}_{\mathrm{rglr}}(y') \subseteq \Delta$ and $x' \xRightarrow[\Delta:n]{\epsilon} y'$ where $\Delta = G^{<n}_{\mathrm{rglr}}(x)$;

(P2) If $x \xrightarrow{\alpha} y$, then there exists $y'$ so that $y \approx y'$ and $x' \xRightarrow[\Delta:\alpha]{\epsilon}\xrightarrow[\Delta:\alpha]{\mathsf{p}(\alpha)}\xRightarrow[\Sigma:1]{\epsilon} y'$ where $\Delta = G^{<\alpha}_{\mathrm{rglr}}(x)$. □

In the original literature Lüttgen (1998), PWB as a binary relation over automata in $\mathrm{CCS}^{\mathrm{ch}}$ has been shown to be a congruence w.r.t. composition "|" and restriction "$/L$". in $\mathrm{CCS}^{\mathrm{ch}}$, which is synonymous to synchronous composition without private events for automata. [2] Thus, by a similar line of thought as in Malik et al. (2004), PWB implies conflict equivalence w.r.t. event priority. To prove this, the following proposition is a useful preparation.

**Proposition 10.** Let $G = \langle Q_G, A, \rightarrow_G, Q^\circ_G \rangle$ an $\Upsilon$-shaped automaton with a PWB $\approx \subseteq Q_G \times Q_G$ on $G$. The following two statements hold for any automaton $T = \langle Q_T, A, \rightarrow_T, Q^\circ_T \rangle$, any $x_G, y_G \in Q_G$, any $\alpha \in A$ and any $x_T, y_T \in Q_T$:

(C1) if $([x_G], x_T) \xrightarrow{\alpha}^{\mathcal{S}} ([y_G], y_T)$ in $\mathcal{S}(G/\!\approx \,\|\, T)$, then for all $x'_G \in [x_G]$, there exists some $y'_G \in [y_G]$ so that $(x_G, x_T) \xRightarrow{\mathsf{p}(\alpha)}^{\mathcal{S}} (y_G, y_T)$ in $\mathcal{S}(G \,\|\, T)$.

(C2) if $(x_G, x_T) \xrightarrow{\alpha}^{\mathcal{S}} (y_G, y_T)$ in $\mathcal{S}(G \,\|\, T)$, then $([x_G], x_T) \xrightarrow{\mathsf{p}(\alpha)}^{\mathcal{S}} ([y_G], y_T)$ in $\mathcal{S}(G/\!\approx \,\|\, T)$.

[2] Generally, combining the $\mathrm{CCS}^{\mathrm{ch}}$ composition combinator and restriction combinator results in a binary operation which is synonymous to shaping the synchronous composition of two automata in our framework. This was also mentioned in the original CCS Milner (1989), where the composition of automata was referred to as *conjunction*.

*Proof.* (C1): There are two cases:

(Case 1) If $([x_G], x_T) \xrightarrow{\alpha}{}^{\mathcal{S}} ([y_G], y_T)$ is driven by $G/\!\approx$, then from (P2), for all $x'_G \in [x_G]$, there exists some $\bar{x}_G \in Q_G$, $\bar{y}_G \in Q_G$ and $y'_G \in [y_G]$ so that $x'_G \xRightarrow[\Delta:\alpha]{\epsilon} \bar{x}_G \xrightarrow[\Delta:\alpha]{\mathsf{p}(\alpha)} \bar{y}_G \xRightarrow[\Sigma:1]{\epsilon} y'_G$ where $\Delta = G^{<\alpha}_{\text{rglr}}(x_G)$. Note that $\Delta \subseteq G/\!\approx^{<\alpha}_{\text{rglr}}([x_G])$. This indeed enables a transition $(x'_G, x_T) \xRightarrow[\Delta:\alpha]{\epsilon}{}^{\mathcal{S}} (\bar{x}_G, x_T) \xrightarrow[\Delta:\alpha]{\mathsf{p}(\alpha)}{}^{\mathcal{S}} (\bar{y}_G, y_T) \xRightarrow[\Sigma:1]{\epsilon}{}^{\mathcal{S}} (y'_G, y_T)$ in $\mathcal{S}(G \parallel T)$.

(Case 2) Otherwise, $([x_G], x_T) \xrightarrow{\alpha}{}^{\mathcal{S}} ([y_G], y_T)$ is not driven by $G/\!\approx$. This implies that $[x_G] = [y_G]$, $\alpha \in \Upsilon$ and $G/\!\approx^{<\alpha}_{\text{slnt}}([x_G]) = \emptyset$. Then from (P1), for all $x'_G \in [x_G]$, there exists $y'_G \in [y_G] = [x_G]$ so that $G^{<\alpha}_{\text{slnt}}(y'_G) = \emptyset$, $G^{<\alpha}_{\text{rglr}}(y'_G) \subseteq \Delta$ and $x'_G \xRightarrow[\Delta:n]{\epsilon} y'_G$ where $\Delta = G^{<n}_{\text{rglr}}(x_G)$. This indeed enables a transition $(x'_G, x_T) \xRightarrow{\epsilon}{}^{\mathcal{S}} (y'_G, x_T) \xrightarrow{\alpha}{}^{\mathcal{S}} (y'_G, y_T)$ in $\mathcal{S}(G \parallel T)$

(C2): There are two cases:

(Case 1) Let $(x_G, x_T) \xrightarrow{\alpha}{}^{\mathcal{S}} (y_G, y_T)$ be driven by $G$. In this case, if $x_G \approx y_G$ and $\alpha \in \Upsilon$, then we have a trivial transition $(x_G, x_T) \xrightarrow{\epsilon}{}^{\mathcal{S}} (y_G, y_T) = (x_G, x_T)$ in $\mathcal{S}(G \parallel T)$. Otherwise, suppose $([x_G], x_T) \not\xrightarrow{\alpha}{}^{\mathcal{S}} ([y_G], y_T)$ in $\mathcal{S}(G \parallel T)$. There must then exist some $\alpha' \in A$ so that $([x_G], x_T) \xrightarrow{\alpha'}{}^{\mathcal{S}}$ in $\mathcal{S}(G/\!\approx \parallel T)$ and $\mathsf{prio}(\alpha') < \mathsf{prio}(\alpha)$. Clearly, $([x_G], x_T) \xrightarrow{\alpha'}{}^{\mathcal{S}}$ cannot be driven by $T$ from $(x_G, x_T) \xrightarrow{\alpha}{}^{\mathcal{S}} (y_G, y_T)$. There are two further sub-cases:

(i) $\alpha' \in \Upsilon$. Note that in this case, $\alpha'$ cannot appear as a self-loop over $[x_G]$ in $G/\!\approx$. If so, then $[x_G]$ contains some $\alpha'$-live-lock in $G$. Note that $G^{<\alpha}_{\text{slnt}}(x_G) = \emptyset$ must hold from the $\Upsilon$-shapedness. Then from (P1), $[x_G]$ cannot contain such $\alpha'$-live-locks. Thus, there exists some $x'_G \in [x_G]$ and $y_G \in Q_G - [x_G]$ so that $x'_G \xrightarrow{\alpha} y_G$. From (P2), it implies the existence of some $\tau \in G_{\text{slnt}}(x_G)$ so that $\mathsf{prio}(\tau) \leq \mathsf{prio}(\alpha')$, which contradicts $(x_G, x_T) \xrightarrow{\alpha}{}^{\mathcal{S}} (y_G, y_T)$.

(ii) If $\alpha' \in \Sigma$, then similar to (i), there must exist some $\alpha'' \in G^{<\alpha}(x_G)$ so that $\mathsf{prio}(\alpha'') \leq \mathsf{prio}(\alpha')$, which contradicts $(x_G, x_T) \xrightarrow{\alpha}{}^{\mathcal{S}} (y_G, y_T)$.

(Case 2) Otherwise, $(x_G, x_T) \xrightarrow{\alpha}{}^{\mathcal{S}} (y_G, y_T)$ is not driven by $G$. This case can be reasoned from (i) and (ii) as in Case 1 of C2 directly. □

By performing a simple induction on the result of the above proposition, we are in the position to state the following result.

**Theorem 11.** Let $G = \langle Q_G, A, \rightarrow_G, Q^{\circ}_G \rangle$ be an $\Upsilon$-shaped automata with an PWB $\approx \subseteq Q \times Q$. It then holds that $G \simeq_{\mathcal{S}} (G/\!\approx)$.

*Proof.* Let $T = \langle Q_T, A, \rightarrow_T, Q^{\circ}_T \rangle$ be any automaton. Suppose $\mathcal{S}(G \parallel T)$ is non-blocking, we shall attempt to prove that $\mathcal{S}(G/\!\approx \parallel T)$ must be non-blocking (The

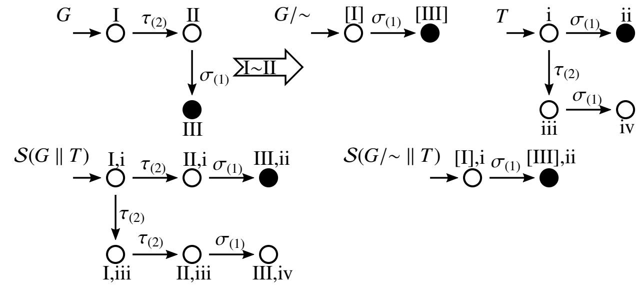

*Figure 1: a silent step with priority lower then its delayed non-silent action may not be mergable*

proof for the conversed case is similar). Pick any $y_G \in Q_G$ so that $([x_G^\circ], x_T^\circ) \overset{s}{\Rightarrow}^{\mathcal{S}} ([y_G], y_T)$ for some $s \in \Sigma^*$, $x_G^\circ \in Q_G^\circ$, $x_T^\circ \in Q_T^\circ$ and $y_T \in Q_T$. Note that $[x_G^\circ] \in \tilde{Q}_G$ must hold. By Proposition 10.(C1), it follows from induction on concatenated transitions of any trace in $([x_G^\circ], x_T^\circ) \overset{s}{\Rightarrow}^{\mathcal{S}} ([y_G], y_T)$ that there exists $y'_G \in [y_G]$ so that $(x_G^\circ, x_T^\circ) \overset{s}{\Rightarrow}^{\mathcal{S}} (y'_G, y_T)$ in $\mathcal{S}(G \parallel T)$, i.e. $\mathcal{S}(G \parallel T) \overset{s}{\Rightarrow}^{\mathcal{S}} (y'_G, y_T)$. Moreover, since $\mathcal{S}(G \parallel T)$ is non-blocking, $(y'_G, y_T) \overset{t\omega}{\Rightarrow}^{\mathcal{S}}$ in $\mathcal{S}(G \parallel T)$ for some $t \in \Sigma^*$ must hold. Again from Proposition 10.(C2), we can conclude through induction that $([y'_G], y_T) = ([y_G], y_T) \overset{t\omega}{\Rightarrow}^{\mathcal{S}}$ in $\mathcal{S}(G/\approx \parallel T)$. The proof is indeed closed since $y_G$ is arbitrarily picked. □

Note that PWB is defined such that if at some state a regular event $\sigma \in \Sigma$ can be executed, an equivalent state must be able to execute $\sigma$ as well, either immediately or after a number of silent steps with priority *not lower then* $\mathsf{prio}(\sigma)$. The importance of this restriction for conflict equivalence can be seen from the following example. For the brevity in figures, we directly write the priority of each event in the subscript of transition labels. For simplicity, we always assume that $\mathsf{prio}(\omega) = 1$ in the current and subsequent section.

Consider the automaton $G$ as given in Figure 1 again. The failure of the abstraction can be seen as being caused by the *reachable* state $(\mathrm{I}, \mathrm{i})$ in $\mathcal{S}(G \parallel T)$ as state i has the chance to execute $\tau$ whose priority is lower then $\sigma$ since $\sigma \notin G_{\mathrm{rglr}}(\mathrm{I})$. Interestingly, adding further restriction on the automaton can render such "bad" states to be unreachable. As for $G$ in Figure 1, we could switch the initial state to a new state IV and add a new transition $\mathrm{IV} \xrightarrow{\tau_{(3)}} \mathrm{I}$. For such an automaton $G'$ as given in Figure 2, merging I and II yields a conflict preserving abstraction. The intuition behind this modification is that due to the new transition, $(\mathrm{I}, \mathrm{i})$ becomes unreachable. In this case, we say $\mathrm{I} \xrightarrow{\tau} \mathrm{II}$ is a *redundant silent step*.

**Definition 12.** Let $G = \langle Q, A, \rightarrow, Q^\circ \rangle$ be an $\Upsilon$-shaped automaton. A transition $x \xrightarrow{\tau} y$ with $x, y \in Q$ and $\tau \in \Upsilon$ is a *redundant silent step* if this is the only

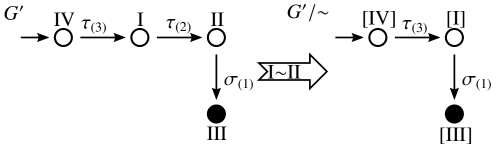

*Figure 2: redundant silent step rule*

transition outgoing from $x$, $x \notin Q^\circ$ and $z \xrightarrow{\alpha} x$ for any $z \in Q$ implies $\alpha \in \Upsilon$ and $\mathsf{prio}(\alpha) > \mathsf{prio}(\tau)$. An equivalence $\sim \subseteq Q \times Q$ on $G$ is *induced by the transition* $x \xrightarrow{\alpha} y$ if $x \sim y$ and for all $z \notin \{x, y\}$, $[z]$ is a singleton class. □

Before proceeding, we first note that the definition of a redundant silent step does not specifically handle the existence of live-locks. The key point is that the active event set of the target state of a redundant silent step can be completely preserved in the quotient automaton. This is stated by the following lemma.

**Lemma 13.** Let $G = \langle Q, A, \rightarrow, Q^\circ \rangle$ be a $\Upsilon$-shaped automaton and the equivalence $\sim \subseteq Q \times Q$ is induced by the redundant silent step $x \xrightarrow{\tau} y$. Then $G(y) = G/{\sim}([y])$.

*Proof.* It suffices to consider the case that $[x] \xrightarrow{\tau'}_{\sim} [x]$ in $G/{\sim}$ for some $\tau' \in \Upsilon$. In this case, $[x]$ contains a $\tau'$-live-lock from $G$ which is formed either by $\{x, y\}$ or solely by $\{y\}$ (solely by $\{x\}$ is clearly impossible). The case of solely by $\{y\}$ is rather trivial, while when $\{x, y\}$ forms a $\tau'$-live-lock, we must have $y \xrightarrow{\tau'} x$ since from the definition of a redundant silent step, $\mathsf{prio}(\tau') > \mathsf{prio}(\tau)$ must hold. □

Given a redundant silent step $x_G \xrightarrow{\tau} y_G$ in $G$ with some non-silent event $\sigma$ active in $y_G$ whose priority is higher than $\tau$, $x$ and $y$ are never prioritised weak bisimilar. Intuitively, this invalidates the property given in Proposition 10 if it is assumed that the resulting quotient automaton and the original one are "equivalent". More precisely, for some state $x_T$ in a test automaton $T$, if $x_T \xrightarrow{\tau'}$ for some $\mathsf{prio}(\tau') \leq \mathsf{prio}(\tau)$, we must have $(x_G, x_T) \xrightarrow{\tau'}^{\mathcal{S}}$ in $\mathcal{S}(G \parallel T)$, while $([x_G], x_T) \xrightarrow{\tau'}^{\mathcal{S}}$ may *not* hold in $\mathcal{S}(G/{\sim} \parallel T)$ when $\mathsf{prio}(\sigma) < \mathsf{prio}(\tau')$. Interestingly, such $(x_G, x_T)$ is never reachable in $\mathcal{S}(G \parallel T)$.

**Proposition 14.** Let $G = \langle Q_G, A, \rightarrow_G, Q_G^\circ \rangle$ be a $\Upsilon$-shaped automaton and the equivalence $\sim \subseteq Q_G \times Q_G$ is induced by the redundant silent step $\bar{x}_G \xrightarrow{\tau} \bar{x}'_G$. Let $n = \mathsf{prio}(\tau) + 1$ and $T = \langle Q_T, A, \rightarrow_T, Q_T^\circ \rangle$ be any automaton. Then for all $\bar{x}_T \in Q_T$ so that $T_{\mathrm{slnt}}^{<n}(\bar{x}_T) \neq \emptyset$, $(\bar{x}_G, \bar{x}_T)$ is not reachable in $\mathcal{S}(G \parallel T)$.

*Proof.* We prove through contradiction: Pick any $\bar{x}_T \in Q_T$ so that $T_{\mathrm{slnt}}^{<n}(\bar{x}_T) \neq \emptyset$. To reach $(\bar{x}_G, \bar{x}_T)$, one shall first reach some $(y_G, y_T)$ where $y_G \in Q_G$, $y_T \in Q_T$ so

that $y_G \xrightarrow{\tau'} \bar{x}_G$ with some $\tau' \in \Upsilon$. From Definition 12, it is clear that $\mathsf{prio}(\tau') \geq n$. This implies that $(y_G, \bar{x}_T) \overset{\tau'}{\not\to}{}^{\mathcal{S}} (\bar{x}_G, \bar{x}_T)$. With this observation, we continue the proof by attempting to construct a trace from $(y_G, y_T)$ to $(\bar{x}_G, \bar{x}_T)$, which must fail. Consider the following cases:

(Case 1) $T^{<n}_{\mathrm{slnt}}(y_T) \neq \emptyset$. Let $y_T \xrightarrow{\tau''} \bar{y}_T$ for some $\bar{y}_T \in Q_T$ and $\tau'' \in T^{<n}_{\mathrm{slnt}}(y_T)$. Clearly, $\mathsf{prio}(\tau'') < \mathsf{prio}(\tau')$, and we concatenate $(y_G, y_T) \xrightarrow{\tau''}{}^{\mathcal{S}} (y_G, \bar{y}_T)$. If $T^{<n}_{\mathrm{slnt}}(\bar{y}_T) \neq \emptyset$ always holds for such concatenation, then the construction is trapped in Case 1 and $\bar{x}_G$ can never be visited. Otherwise, let $T^{<n}_{\mathrm{slnt}}(\bar{y}_T) = \emptyset$, which leads to a Case 2 situation.

(Case 2) $T^{<n}_{\mathrm{slnt}}(y_T) = \emptyset$. From $(y_G, y_T)$, since only private and silent events can be executed, consider the possibility of concatenating $(y_G, y_T) \xrightarrow{\tau'}{}^{\mathcal{S}} (\bar{x}_G, y_T)$ in $\mathcal{S}(G \parallel T)$, since executing a private transition in $T$ indeed rolls the construction back to the beginning of either Case 1 or 2. However, if $(y_G, y_T) \xrightarrow{\tau'}{}^{\mathcal{S}} (\bar{x}_G, y_T)$, it implies that the next transition which can be concatenated must be $(\bar{x}_G, y_T) \xrightarrow{\tau}{}^{\mathcal{S}} (\bar{x}'_G, y_T)$ since $\mathsf{prio}(\tau) < \mathsf{prio}(\tau')$ and executing any shared event with priority higher than $\tau$ in $(\bar{x}_G, y_T)$ is not possible. Recall that $y_T \neq \bar{x}_T$ due to $T^{<n}_{\mathrm{slnt}}(\bar{x}_T) \neq \emptyset$, i.e. for any $z_T \in Q_T$ so that $(\bar{x}_G, z_T)$ is reachable in $\mathcal{S}(G \parallel T)$, $T^{<n}_{\mathrm{slnt}}(z_T) = \emptyset$ must hold. This indeed closes the proof. □

When merging a redundant silent step, states characterised in Proposition 14 are exactly those "bad" states which potentially invalidate conflict preservation. This implies the following proposition which in turn implies conflict preservation.

**Proposition 15.** Let $G = \langle Q_G, A, \to_G, Q^\circ_G \rangle$ be an $\Upsilon$-shaped automaton. The equivalence $\sim \subseteq Q_G \times Q_G$ is induced by the redundant silent step $\bar{x}_G \xrightarrow{\tau} \bar{x}'_G$. The following two statements hold for any $T = \langle Q_T, A, \to_T, Q^\circ_T \rangle$:

(R1) If $([x_G], x_T) \xrightarrow{\alpha}{}^{\mathcal{S}} ([y_G], y_T)$ in $\mathcal{S}(G/\!\sim \parallel T)$, then for all $x'_G \in [x_G]$, at least one of the following two statements is true:

(i) There exists some $y'_G \in [y_G]$ so that $(x'_G, x_T) \overset{\mathsf{p}(\alpha)}{\Longrightarrow}{}^{\mathcal{S}} (y'_G, y_T)$ in $\mathcal{S}(G \parallel T)$.

(ii) $(x'_G, x_T)$ is not reachable in $\mathcal{S}(G \parallel T)$.

(R2) If $(x_G, x_T) \xrightarrow{\alpha}{}^{\mathcal{S}} (y_G, y_T)$ in $\mathcal{S}(G \parallel T)$, then at least one of the following two statements is true:

(i) $([x_G], x_T) \xrightarrow{\mathsf{p}(\alpha)}{}^{\mathcal{S}} ([y_G], y_T)$ in $\mathcal{S}(G/\!\sim \parallel T)$.

(ii) $(x_G, x_T)$ is not reachable in $\mathcal{S}(G \parallel T)$.

*Proof.* (R1): We assume that $[x_G] = [\bar{x}_G]$, since if not, $[x_G]$ is a singleton and statement (i) must hold due to Lemma 7.(i). In this case, note that if $([x_G], x_T) \xrightarrow{\alpha}{}^{\mathcal{S}} ([y_G], y_T)$ is not driven by $G$, then statement (i) must be true as well since either $(\bar{x}_G, x_T) \xrightarrow{\alpha}{}^{\mathcal{S}} (\bar{x}_G, y_T)$ or $(\bar{x}_G, x_T) \xrightarrow{\tau}{}^{\mathcal{S}} (\bar{x}'_G, x_T) \xrightarrow{\alpha}{}^{\mathcal{S}} (\bar{x}'_G, y_T)$ holds in $\mathcal{S}(G \parallel T)$.

Thus, let $([x_G], x_T) \xrightarrow{\alpha}^{\mathcal{S}} ([y_G], y_T)$ be driven by $G$, implying $\alpha \in G(\bar{x}'_G)$. There are two cases:

(Case 1) $x'_G = \bar{x}'_G$. We shall note that $G(\bar{x}'_G) = G/{\sim}([\bar{x}'_G])$ from Lemma 13. Thus, in this case, statement (i) must hold.

(Case 2) $x'_G = \bar{x}_G$. We directly suppose that statement (i) is not true, i.e. $(\bar{x}_G, x_T) \overset{\mathsf{p}(\alpha)}{\not\Longrightarrow}^{\mathcal{S}} (y'_G, y_T)$ in $\mathcal{S}(G \parallel T)$ for any $y'_G \in [y_G]$. This implies that $T^{<\tau}_{\mathrm{slnt}}(x_T) \neq \emptyset$, since otherwise, we must be able to execute $(\bar{x}_G, x_T) \xrightarrow{\tau}^{\mathcal{S}} (\bar{x}'_G, x_T)$, which is indeed Case 1. Thus, in this case, statement (ii) must hold from Proposition 14.

(R2): Note that statement (i) must hold if $[x_G]$ is a singleton. In addition, statement (i) holds for $x_G = \bar{x}'_G$ as well from Lemma 13. Let $x_G = \bar{x}_G$. If $(x_G, x_T) \xrightarrow{\alpha}^{\mathcal{S}} (y_G, y_T)$ is driven by $G$, then statement (i) holds from a trivial transition $([x_G], x_T) \xrightarrow{\epsilon} ([y_G], x_T)$. Let $(x_G, x_T) \xrightarrow{\alpha}^{\mathcal{S}} (y_G, y_T)$ be not driven by $G$. In this case, statement (ii) must hold from Proposition 14 since $\mathsf{prio}(\alpha) \leq \mathsf{prio}(\tau)$, i.e. $\alpha \in T^{<n}_{\mathrm{slnt}}(x_T)$ with $n = \mathsf{prio}(\tau) + 1$. □

From this we obtain the following abstraction rule.

**Theorem 16** (redundant silent step rule)**.** Given an $\Upsilon$-shaped automaton $G = \langle Q_G, A, \rightarrow_G, Q^\circ_G \rangle$ and the equivalence $\sim \subseteq Q_G \times Q_G$ induced by a redundant silent step. It then holds that $G \simeq_{\mathcal{S}} (G/{\sim})$.

*Proof.* The proof is indeed the same as that of Theorem 11. □

## 4 Abstraction rules based on incoming equivalence

For ordinary conflict-preserving abstraction without event priorities, Flordal and Malik (2009) introduce the *active events rule* and the *silent continuation rule* which are based on a pre-partition through *incoming equivalence*. The key property of incoming equivalence in the ordinary set-up is that, if there is a trace beginning with a non-silent event in the composition after abstraction, then a trace with the same non-silent events can be constructed in the original automaton as well. To achieve this property with event priorities, we use the following notion *string preservation*.

**Definition 17.** Let $G = \langle Q_G, A, \rightarrow_G, Q^\circ_G \rangle$ be an $\Upsilon$-shaped automaton. An equivalence $\sim \subseteq Q \times Q$ on $G$ is *string-preserving* if for any arbitrary automaton $T = \langle Q_T, A, \rightarrow_T, Q^\circ_T \rangle$ and any trace

$$([x_{G0}], x_{T0}) \xrightarrow{\alpha_1}^{\mathcal{S}} ([x_{G1}], x_{T1}) \xrightarrow{\alpha_2}^{\mathcal{S}} \cdots \xrightarrow{\alpha_k}^{\mathcal{S}} ([x_{Gk}], x_{Tk})$$

in $\mathcal{S}(G/\!\sim \parallel T)$ where $k \geq 1$, $\alpha_1 \in \Sigma$ and $\alpha_i \in A$ for all $i \in \{2, \cdots, k\}$, there exist $x'_{G0} \in [x_{G0}]$ and $x'_{Gk} \in [x_{Gk}]$ so that $(x'_{G0}, x_{T0}) \xRightarrow{\mathsf{p}(\alpha_1 \cdots \alpha_k)}_{\mathcal{S}} (x'_{Gk}, x_{Tk})$ in $\mathcal{S}(G \parallel T)$. □

To achieve string preservation, we first conveniently define some new notations for transitions.

**Definition 18.** Given an $\Upsilon$-shaped automaton $G = \langle Q, A, \rightarrow, Q^\circ \rangle$, define the following extended transition relations:

(T3) $\underset{!}{\rightarrow} \subseteq Q \times \Upsilon \times Q$: $x \xrightarrow[!]{\tau} y$ if $x \xrightarrow{\tau} y$ and $G^{<\tau}_{\mathrm{rglr}}(x) = \emptyset$.
(T4) $\underset{!n}{\hookrightarrow} \subseteq Q \times \{\epsilon\} \times Q$: $x \overset{\epsilon}{\underset{!n}{\hookrightarrow}} y$ if
  (i) either $n = 1$ and $x \xRightarrow[\Sigma:1]{\epsilon} y$,
  (ii) or $n \geq 2$, $x \xrightarrow[!]{\tau_1}\xrightarrow[!]{\tau_2} \cdots \xrightarrow[!]{\tau_k} y$, $k \geq 1$ and $\mathsf{lo}(\{\tau_1, \cdots, \tau_k\}) = n$. □

Transition relations introduced in Definition 18 are generally more restrictive than those in Definition 8 in that all non-final states are individually in shaped form. Note that we intentionally use the new transition symbol "$\hookrightarrow$" since when $n \geq 1$, we do *not* have $x \overset{\epsilon}{\underset{!n}{\hookrightarrow}} x$ for all $x$ as at least one $\tau$ transition with $\mathsf{prio}(\tau) = n$ must happen during $\overset{\epsilon}{\underset{!n}{\hookrightarrow}}$. Based on Definition 18, we introduce the modified incoming equivalence when considering priority:

**Definition 19.** Let $G = \langle Q, A, \rightarrow, Q^\circ \rangle$ be an $\Upsilon$-shaped automaton. An equivalence $\sim_{\mathrm{inc}} \subseteq Q \times Q$ on $G$ is an *incoming equivalence* if for any $x, x' \in Q$ so that $x \sim_{\mathrm{inc}} x'$, it holds that

(I1) for all $\sigma \in \Sigma$, $n \geq 0$ and $y \in Q$, $y \xRightarrow[\Delta:\sigma]{\epsilon}\xrightarrow[\Delta:\sigma]{\sigma}\overset{\epsilon}{\underset{!n}{\hookrightarrow}} x \;\Leftrightarrow\; y \xRightarrow[\Delta:\sigma]{\epsilon}\xrightarrow[\Delta:\sigma]{\sigma}\overset{\epsilon}{\underset{!n}{\hookrightarrow}} x'$ where $\Delta = G^{<\sigma}_{\mathrm{rglr}}(y)$;
(I2) for any $n \geq 0$, $Q^\circ \overset{\epsilon}{\underset{!n}{\hookrightarrow}} x \;\Leftrightarrow\; Q^\circ \overset{\epsilon}{\underset{!n}{\hookrightarrow}} x'$ ;
(I3) for any $y \in Q$ and $\tau \in \Upsilon$, $y \overset{\tau}{\rightarrow}\overset{\epsilon}{\Rightarrow} x$ or $y \overset{\tau}{\rightarrow}\overset{\epsilon}{\Rightarrow} x'$ implies $G^{<\tau}_{\mathrm{rglr}}(y) = \emptyset$. □

Similar to the original version in Flordal and Malik (2009), Definition 19 attempts to equalise states which can be reached in the same way, i.e. we only care about the past of a state and ignore its future behaviour. However, such intuition is inadequate when event priorities are considered since string preservation requires the same state $x_{T0}$ and $x_{Tk}$ from some test $T$ to be connected before and after abstraction. If no restriction over the future behaviour of incoming equivalent states is given, one may fail to achieve string preservation in that two equivalent states may have different preemption power. In this regard, we first introduce our definition of active-event equivalence and silent-continuation equivalence.

**Definition 20.** Let $G = \langle Q, A, \rightarrow, Q^\circ \rangle$ be an $\Upsilon$-shaped automaton. An equivalence $\sim_{\mathrm{ae}} \subseteq Q \times Q$ on $G$ is an *active-event equivalence* if for any $x, x' \in Q$ so that $x \sim_{\mathrm{ae}} x'$, either $x = x'$ or:

(AE1) $G_{\mathrm{slnt}}(x) = G_{\mathrm{slnt}}(x') = \emptyset$;

(AE2) $G_{\mathrm{rglr}}(x) = G_{\mathrm{rglr}}(x')$. □

**Definition 21.** Let $G = \langle Q, A, \rightarrow, Q^\circ \rangle$ be an $\Upsilon$-shaped automaton. An equivalence $\sim_{\mathrm{sc}} \subseteq Q \times Q$ on $G$ is a *silent-continuation equivalence* if for any $x, x' \in Q$ so that $x \sim_{\mathrm{sc}} x'$, either $x = x'$, or there exists some $\tau \in \Upsilon$ so that:

(SC1) $\tau \in G_{\mathrm{slnt}}(x) \cap G_{\mathrm{slnt}}(x')$;

(SC2) $G^{<\tau}_{\mathrm{rglr}}(x) = G^{<\tau}_{\mathrm{rglr}}(x') = \emptyset$;

(SC3) Neither $x$ nor $x'$ is in any live-lock. □

Before proceeding to prove Proposition 25, note that $\sim_{\mathrm{ae}}$ imposes a relatively strong restriction on equivalent states that silent events are never active on any state in a non-singleton class. Readers may be curious about the possibility of relaxing Definition 20 to equate states with non-silent active events delayed by $\xrightarrow{\tau^*_{(1)}}$, i.e., by defining $\Delta_{\mathrm{ae}}(x) := \{\sigma \in \Sigma \mid x \xrightarrow{\tau^*_{(1)}\sigma}\}$, one may expect that $x \sim x'$ when $\Delta_{\mathrm{ae}}(x) = \Delta_{\mathrm{ae}}(x')$, since $\tau_{(1)}$ transitions is will not be preempted by any events. More attractively, relaxing in this manner also preserves the $\Upsilon$-shapedness of a given automaton. Consider the following example:

**Example 22.** Let $G = \langle Q_G, A, \rightarrow_G, Q^\circ_G \rangle$ and $T = \langle Q_T, A, \rightarrow_T, Q^\circ_T \rangle$ be such as given in Figure 3 with $\Sigma = \{\sigma, \omega\}$. Note that $G$ is $\Upsilon$-shaped and it clearly holds that I $\sim_{\mathrm{inc}}$ III since state III can be reached from the initial state through $\tau^*_{(0)}$. Furthermore, from $\Delta_{\mathrm{ae}}(x) = \Delta_{\mathrm{ae}}(x')$, we are able to equate states I and III to construct $G/\!\sim$. However, this is not the case of $\sim_{\mathrm{ae}}$ as defined in Definition 20 which causes the invalidation of string preservation. Although ([II], ii) is reachable in $\mathcal{S}(G/\!\sim \parallel T)$, (II, ii) is not reachable in $\mathcal{S}(G \parallel T)$ since i $\xrightarrow{\tau'}$ ii cannot happen before I $\xrightarrow{\tau}$ II and the transition I $\xrightarrow{\sigma}$ II is labelled by a shared event $\sigma$. One observe that in this example, I $\xrightarrow{\tau}$ III somewhat "disables" I $\xrightarrow{\sigma}$ II although both events are with the same priority. In this case, equating I and III is unacceptable, especially when both states have different future behaviour, e.g. one leads to a marking state while another only has blocking future behaviour. □

From the above example, the importance for two equivalent states allowing the same set of silent events from other modules to happen is revealed. This property can be guaranteed by both $\sim_{\mathrm{ae}}$ and $\sim_{\mathrm{sc}}$, which is summarised by the following lemma.

**Lemma 23.** Let $G = \langle Q_G, A, \rightarrow_G, Q^\circ_G \rangle$ be a well-formed automaton. Let $\sim \subseteq Q \times Q$ be an equivalence on $G$ so that either $\sim \subseteq \sim_{\mathrm{ae}}$ or $\sim \subseteq \sim_{\mathrm{sc}}$. For any automaton $T$ and any silent trace

$$(x_G, x_{T0}) \xrightarrow{\tau_1}^{\mathcal{S}} (x_G, x_{T1}) \xrightarrow{\tau_2}^{\mathcal{S}} \cdots \xrightarrow{\tau_k}^{\mathcal{S}} (x_G, x_{Tk})$$

in $\mathcal{S}(G \parallel T)$ where $k \geq 0$ and all transitions are driven by $T$. It holds that for any $x'$ so that $x'_G \sim x_G$, a trace

$$(x'_G, x_{T0}) \xrightarrow{\tau_1}^{\mathcal{S}} (x'_G, x_{T1}) \xrightarrow{\tau_2}^{\mathcal{S}} \cdots \xrightarrow{\tau_k}^{\mathcal{S}} (x'_G, x_{Tk})$$

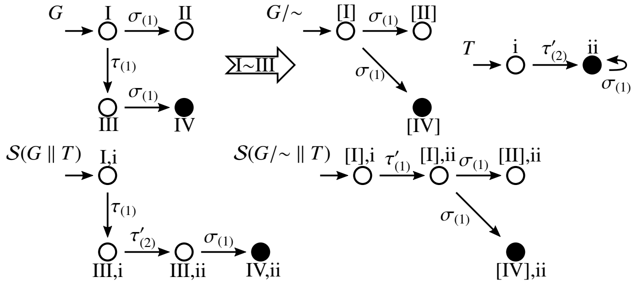

*Figure 3: counterexample of equating incoming equivalent states with the same set of delayed active events*

must exist in $\mathcal{S}(G \parallel T)$ as well. □

We now consider the properties of $\overset{\epsilon}{\underset{!n}{\hookrightarrow}}$, which is utilised in (I1) and (I2). In fact, when the target states of two $\overset{\epsilon}{\underset{!n}{\hookrightarrow}}$ transitions are $\sim_{\mathrm{ae}}$ or $\sim_{\mathrm{sc}}$ equivalent, then both or neither of them can be synchronised with a same silent trace from another automaton. This property is formalised by the statement (ii) of the following proposition.

**Proposition 24.** Let $G = \langle Q_G, A, \rightarrow_G, Q_G^\circ \rangle$ be an $\Upsilon$-shaped automaton with an equivalence $\sim \subseteq Q \times Q$ on $G$ so that either $\sim \subseteq \sim_{\mathrm{ae}}$ or $\sim \subseteq \sim_{\mathrm{sc}}$. Let

$$(x_{G0}, x_{T0}) \xrightarrow{\tau_1}{}^{\mathcal{S}} (x_{G1}, x_{T1}) \xrightarrow{\tau_2}{}^{\mathcal{S}} \cdots \xrightarrow{\tau_k}{}^{\mathcal{S}} (x_{Gk}, x_{Tk}) \tag{6}$$

be a silent trace in $\mathcal{S}(G \parallel T)$ with $k \geq 0$. Let $n = \mathsf{lo}(\{\tau_i \,|\, (x_{Gi-1}, x_{Ti-1}) \xrightarrow{\tau_i} (x_{Gi}, x_{Ti})$ is driven by $G\})$ and

$$x'_{G0} \xrightarrow{\tau'_1} x'_{G1} \xrightarrow{\tau'_2} \cdots \xrightarrow{\tau'_{k'}} x'_{Gk'} \tag{7}$$

be an arbitrary silent trace in $G$ where $k' \geq 0$ so that $\mathsf{lo}(\{\tau'_1, \cdots, \tau'_{k'}\}) = n$ and for all $i' \in \{1, \cdots, k'-1\}$, $G^{<\tau'_{i'}}_{\mathrm{rglr}}(x'_{Gi'}) = \emptyset$. The following two statements hold:

(i) For the trace given in (6), if $k \geq 1$ and the last transition $(x_{Gk-1}, x_{Tk-1}) \xrightarrow{\tau_k}{}^{\mathcal{S}} (x_{Gk}, x_{Tk})$ is driven by $G$, then $(x'_{G0}, x_{T0}) \overset{\epsilon}{\Rightarrow}{}^{\mathcal{S}} (x'_{Gk'}, x_{Tk})$ in $\mathcal{S}(G \parallel T)$ where the last transition is driven by $G$.

(ii) If $x_{Gk} \sim x'_{Gk'}$, then $(x'_{G0}, x_{T0}) \overset{\epsilon}{\Rightarrow}{}^{\mathcal{S}} (x'_{Gk'}, x_{Tk})$ in $\mathcal{S}(G \parallel T)$. □

The correctness of Proposition 24.(ii) can be seen from its weaker version, namely Proposition 24.(i). We consider a silent trace as given in "grid" as depicted in Figure 4, where points on the horizontal axis correspond to states in $Q_G$, while those on the vertical axis correspond to states in $Q_T$. Recall that within $\hookrightarrow$, pre-emption through shared events is impossible. We first notice that each time when the "transition-driving" automaton alternates (i.e. in those states where the trace

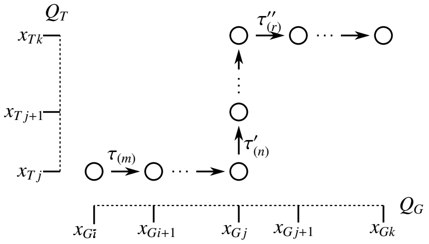

*Figure 4: A silent trace in shaped synchronisation*

"turns"), the priority of the silent event on the next transition cannot elevate. This immediately implies that $m \leq n \leq r$ for the trace in Figure 24. More importantly, if the trace ends with a transition driven by $G$ (as shown in Figure 4), we can immediately conclude that

(O1) the last "$T$-state" of the last state ($x_{Tk}$ in Figure 4) cannot execute any silent event whose priority is higher than any transition in the trace and

(O2) the lowest priority of all transitions driven by $G$ cannot be higher then the lowest priority of all transition driven by $T$.

With both observations, the correctness of Proposition 24.(i) is clear in that for a silent trace as given in (7), we can simply construct a silent trace from $(x'_{G0}, x_{T0})$ to reach $(x'_{Gk'}, x_{Tk})$ in $\mathcal{S}(G \parallel T)$. Due to the observation (O2), we must be able to reach $x_{Tk}$ before reaching $x'_{Gk'}$ and due to (O1), we can fill the rest part from $T$ to reach $x_{Tk}$. Afterwards, combining either $\sim_{\text{ae}}$ or $\sim_{\text{sc}}$ of the last state, Proposition 24.(ii) can be verified easily due to Lemma 23. The proof is formally given as follows:

*Proof of Proposition 24.* Note that the possibility of preemption through shared prioritised events at any $(x'_{Gi'}, x_{Tj})$ where $i' \in \{0, \cdots, k'-1\}$ and $j \in \{0, \cdots, k\}$ is excluded. For convenience, let $n' = \mathsf{lo}(\{\tau'_1, \cdots, \tau'_{k'}\})$.

(i) It suffices to construct a silent trace from $(x'_{G0}, x_{T0})$ to $(x'_{Gk'}, x_{Tk})$ which will not be influenced by shaping and the last transition is driven by $G$. Let $i' = j = 0$ and we start the construction from the first state $(x'_{Gi'}, x_{Tj}) = (x'_{G0}, x_{T0})$. Note that due to Case 2 of Step 2 in the following, it is not possible to reach $x'_{Gk'}$ before $x_{Tk}$ is reached.

(Step 1) Consider two possible cases:

(Case 1) Only $j = k$ holds, i.e. $x_{Tk}$ is reached. Consider the trace given in (6) and from Observation (O1), it follows that $T^{<n}_{\text{slnt}}(x_{Tk}) = \emptyset$. Since $n = n'$ is required, we are able to directly complete the construction by concatenating the remaining transitions driven by $G$ to reach $x'_{Gk'}$, i.e. we must have

$(x'_{Gi'}, x_{Tk}) \overset{\epsilon}{\Rightarrow}^{\mathcal{S}} (x'_{Gk'}, x_{Tk})$ where all transitions are driven by $G$ in $\mathcal{S}(G \parallel T)$, since priority of all remaining transitions driven by $G$ cannot be lower then any $\tau \in T_{\text{slnt}}(x_{Tk})$ and preemption through shared events is impossible. This terminates the construction.

(Case 2) Neither $i' = k'$ nor $j = k$ holds. Go to Step 2.

(Step 2) Since preemption through shared prioritised events is not possible, we can proceed from $(x'_{Gi'}, x_{Tj})$ with either one transition driven by $G$ or one driven by $T$, or both. Consider the two possible cases:

(Case 1) $\mathsf{prio}(\tau'_{Gi'+1}) \neq n'$. Then concatenate either $(x'_{Gi'}, x_{Tj}) \xrightarrow{\tau'_{i'+1}}^{\mathcal{S}} (x'_{Gi'+1}, x_{Tj})$ or $(x'_{Gi'}, x_{Tj}) \xrightarrow{\tau_{j+1}}^{\mathcal{S}} (x'_{Gi'}, x_{Tj+1})$ according to their priority and update either $i' := i' + 1$ or $j := j + 1$, respectively. Go to Step 2.

(Case 2) $\mathsf{prio}(\tau'_{Gi'+1}) = n'$. Since $n = n'$ was required, from Observation (O2), it follows that $\mathsf{prio}(\tau'_{Gi'+1}) = n \geq \mathsf{lo}(\{\tau_i \mid (x_{Gi-1}, x_{Ti-1}) \xrightarrow{\tau_i}^{\mathcal{S}} (x_{Gi}, x_{Ti}) \text{ is driven by } T\})$. Thus, we are able to concatenate the remaining transitions driven by $T$ to reach $x_{Tk}$, i.e. we have $(x'_{Gi'}, x_{Tj}) \overset{\epsilon}{\Rightarrow}^{\mathcal{S}} (x'_{Gi'}, x_{Tk})$ where all transitions are driven by $T$ in $\mathcal{S}(G \parallel T)$. Go to Step 1 and we will be in Case 1 of Step 1.

(ii) The statement is trivially true if all transitions are driven by $T$ due to Lemma 23. Otherwise, for the trace given in (6), consider the trace fragment $(x_{Gi}, x_{Ti}) \xrightarrow{\tau_{i+1}}^{\mathcal{S}} \cdots \xrightarrow{\tau_k}^{\mathcal{S}} (x_{Gk}, x_{Tk})$ where $i \in \{1, \cdots, k-1\}$ so that all transitions are driven by $T$ and $(x_{Gi-1}, x_{Ti-1}) \xrightarrow{\tau_i}^{\mathcal{S}} (x_{Gi}, x_{Ti})$ is driven by $G$ (i.e. $x_{Gi} = x_{Gi+1} = \cdots = x_{Gk}$). From statement (i), $(x'_{G0}, x_{T0}) \overset{\epsilon}{\Rightarrow}^{\mathcal{S}} (x'_{Gk'}, x_{Ti})$ in $\mathcal{S}(G \parallel T)$ holds. Furthermore, due to Lemma 23, we must be able to concatenate the remaining transitions driven by $T$ to reach $x_{Tk}$, i.e. $(x'_{Gk'}, x_{Ti}) \overset{\epsilon}{\Rightarrow}^{\mathcal{S}} (x'_{Gk'}, x_{Tk})$. □

With Proposition 24, we are well prepared to prove that $\sim_{\text{inc}}$ conjuncted with either $\sim_{\text{ae}}$ or $\sim_{\text{sc}}$ is string preserving.

**Proposition 25.** Let $G = \langle Q_G, A, \rightarrow_G, Q^\circ_G \rangle$ be an $\Upsilon$-shaped automaton with an equivalence $\sim \subseteq Q \times Q$ on $G$ be such that either $\sim \subseteq \sim_{\text{inc}} \cap \sim_{\text{ae}}$ or $\sim \subseteq \sim_{\text{inc}} \cap \sim_{\text{sc}}$. It holds that $\sim$ is string-preserving.

*Proof.* Let $T = \langle Q_T, A, \rightarrow_T, Q^\circ_T \rangle$ be any automaton. We complete the proof through induction:

(*Base step*) The statement is trivially true when $k = 0$. For $k = 1$, it holds immediately that there exists $x'_{G0} \in [x_{G0}]$ and $x'_{G1} \in [x_{G1}]$ so that $(x'_{G0}, x_{T0}) \xrightarrow{\alpha_1}^{\mathcal{S}} (x'_{G1}, x_{T1})$ in $\mathcal{S}(G \parallel T)$ due to Lemma 7.

(*Inductive step*) Suppose the proposition holds for some $k \geq 1$, then it shall also hold for $k + 1$. From this hypothesis, for some trace

$$([x_{G0}], x_{T0}) \xrightarrow{\alpha_1}^{\mathcal{S}} \cdots \xrightarrow{\alpha_k}^{\mathcal{S}} ([x_{Gk}], x_{Tk}) \tag{8}$$

in $\mathcal{S}(G/{\sim} \parallel T)$ where $\alpha_1 \in \Sigma$ and $\alpha_i \in A$ for $i \neq 1$, there exist some $x'_{G0} \in [x_{G0}]$ and $x'_{Gk} \in [x_{Gk}]$ so that

$$(x'_{G0}, x_{T0}) \xRightarrow{\mathsf{p}(\alpha_1 \cdots \alpha_k)}{}^{\mathcal{S}} (x'_{Gk}, x_{Tk}) \tag{9}$$

in $\mathcal{S}(G \parallel T)$. Consider any successive transition $([x_{Gk}], x_{Tk}) \xrightarrow{\alpha_{k+1}}{}^{\mathcal{S}} ([x_{Gk+1}], x_{Tk+1})$ of trace (8), which shall imply the existence of $x''_{Gk} \in [x_{Gk}]$ and $x'_{Gk+1} \in [x_{Gk+1}]$ so that $(x''_{Gk}, x_{Tk}) \xrightarrow{\alpha_{k+1}}{}^{\mathcal{S}} (x'_{Gk+1}, x_{Tk+1})$ in $\mathcal{S}(G \parallel T)$ due to Lemma 7. Now if $[x_{Gk}]$ is singleton, the proof closes directly since $x'_{Gk} = x''_{Gk}$. Otherwise, from trace (8), we shall find the last non-silent transition driven by $G$, i.e. find the largest $i$ for $1 \leq i \leq k$ so that $\alpha_i \in \Sigma$. We consider the trace fragment $([x_{Gi-1}], x_{Ti-1}) \xrightarrow{\alpha_i}{}^{\mathcal{S}} ([x_{Gi}], x_{Ti}) \xrightarrow{\alpha_{i+1} \cdots \alpha_k}{}^{\mathcal{S}} ([x_{Gk}], x_{Tk})$ from (8) where $\alpha_{i+1} \cdots \alpha_k \in \Sigma^*_{\Upsilon}$. Due to the inductive hypothesis, we can extract the fragment

$$(\bar{x}_G, \bar{x}_T) \xrightarrow{\alpha_i} (\bar{x}'_G, \bar{x}'_T) \overset{\epsilon}{\Rightarrow}{}^{\mathcal{S}} (x'_{Gk}, x_{Tk}) \tag{10}$$

from some trace expressible by (9) for some $\bar{x}_G, \bar{x}'_G \in Q_G$ and $\bar{x}_T, \bar{x}'_T \in Q_T$. Since $x'_{Gk} \sim x''_{Gk}$, we have $\bar{x}_G \xRightarrow[\Delta:\alpha_i]{\epsilon} \bar{y}_G \xrightarrow[\Delta:\alpha_i]{\alpha_i} \bar{y}'_G \overset{\epsilon}{\underset{!n}{\hookrightarrow}} x''_{Gk}$ where $\Delta = G^{<\alpha_i}_{\mathrm{rglr}}(\bar{x}_G)$ in $G$ for some $\bar{y}_G, \bar{y}'_G \in Q_G$ and $n \geq 1$. With the help of Proposition 24.(ii), we have

$$(\bar{x}_G, \bar{x}_T) \overset{\epsilon}{\Rightarrow}{}^{\mathcal{S}} (\bar{y}_G, \bar{x}_T) \xrightarrow{\alpha_i}{}^{\mathcal{S}} (\bar{y}'_G, \bar{x}'_T) \overset{\epsilon}{\Rightarrow}{}^{\mathcal{S}} (x''_{Gk}, x_{Tk}) \tag{11}$$

in $\mathcal{S}(G \parallel T)$ which can be concatenated by transition $([x_{Gk}], x_{Tk}) \xrightarrow{\alpha_{k+1}}{}^{\mathcal{S}} ([x_{Gk+1}], x_{Tk+1})$. □

When the first state in the trace as described in Proposition 25 is a initial state, string preservation can be further relaxed as follows:

**Proposition 26.** Let $G$ be a well-formed automaton with an equivalence relation $\sim \subseteq Q_G \times Q_G$ on $G$ being such that either $\sim \subseteq \sim_{\mathrm{inc}} \cap \sim_{\mathrm{ae}}$ or $\sim \subseteq \sim_{\mathrm{inc}} \cap \sim_{\mathrm{sc}}$ holds. Then for any arbitrary automaton $T = \langle Q_T, A, \rightarrow_T, Q^\circ_T \rangle$, if $\mathcal{S}(G/{\sim} \parallel T) \overset{s}{\Rightarrow}{}^{\mathcal{S}} ([x_G], x_T)$ for some $s \in \Sigma^*$, then there exists $x'_G \in [x_G]$ so that $\mathcal{S}(G \parallel T) \overset{s}{\Rightarrow}{}^{\mathcal{S}} (x'_G, x_T)$.

*Proof.* We separate the proof into two cases:

(Case 1) $s = \epsilon$. Note that if $G/{\sim} \overset{\epsilon}{\underset{!n}{\hookrightarrow}}_{\sim} [x_G]$ for some $n \geq 1$, then for all $x'_G \in [x_G]$, $G \overset{\epsilon}{\underset{!n}{\hookrightarrow}} x'_G$ must hold, which can be proven by a simple induction based on (I2). Also note that $G/{\sim}$ is $\Upsilon$-shaped. Thus, this case holds directly from Proposition 24.(ii). Note that we have proven an even more general version of the statement, i.e. for all states in $[x_G]$ instead of the existence of some state in $[x_G]$, which will be utilised in the proof for the next case.

(Case 2) Let $s \neq \epsilon$. Then let

$$\mathcal{S}(G/{\sim} \parallel T) \overset{\epsilon}{\Rightarrow}{}^{\mathcal{S}} ([y_G], y_T) \xrightarrow{\sigma}{}^{\mathcal{S}} ([z_G], z_T) \overset{t}{\Rightarrow}{}^{\mathcal{S}} ([x_G], x_T) \tag{12}$$

where $\sigma \in \Sigma$ and $t \in \Sigma^*$ so that $\sigma t = s$. From Case 1, for all $y'_G \in [y_G]$, $\mathcal{S}(G \parallel T) \overset{\epsilon}{\Rightarrow}^{\mathcal{S}} (y'_G, y_T)$. From Proposition 25, there exists $y''_G \in [y_G]$ and $x'_G \in [x_G]$ so that $(y''_G, y_T) \overset{\sigma t}{\Rightarrow}^{\mathcal{S}} (x'_G, x_T)$, which indeed closes the proof. □

Note that generally, string preservation guarantees the existence of a string before abstraction from a string after abstraction. We shall also prove a similar statement in the reversed direction, i.e. we shall ensure that a transition before abstraction shall also consistently exists after abstraction.

**Proposition 27.** Let $G = \langle Q_G, A, \rightarrow_G, Q_G^\circ \rangle$ be an $\Upsilon$-shaped automaton with an equivalence $\sim \subseteq Q_G \times Q_G$ on $G$ so that either $\sim \subseteq \sim_{\mathrm{inc}} \cap \sim_{\mathrm{ae}}$ or $\sim \subseteq \sim_{\mathrm{inc}} \cap \sim_{\mathrm{sc}}$ holds. For any arbitrary automaton $T = \langle Q_T, A, \rightarrow_T, Q_T^\circ \rangle$, if $(x_G, x_T) \overset{\alpha}{\rightarrow}^{\mathcal{S}} (y_G, y_T)$ in $\mathcal{S}(G \parallel T)$, then $([x_G], x_T) \xrightarrow{\mathsf{p}(\alpha)}^{\mathcal{S}} ([y_G], y_T)$ in $\mathcal{S}(G/{\sim} \parallel T)$.

*Proof.* Let $\mathsf{prio}(\alpha) = m$. If $x_G \sim y_G$ and $x_G \overset{\alpha}{\rightarrow} y_G$ for $\alpha \in \Upsilon$ being driven by $G$, we will have a trivial transition $([x_G], x_T) \overset{\epsilon}{\rightarrow}^{\mathcal{S}} ([y_G], y_T) = ([x_G], x_T)$ in $\mathcal{S}(G/{\sim} \parallel T)$, regardless $\alpha$-selfloop being preserved by the quotient or not. Otherwise, $([x_G], x_T) \overset{\alpha}{\rightarrow} ([y_G], y_T)$ in $G/{\sim} \parallel T$. This transition will clearly not be shaped due to the definition of $\sim_{\mathrm{ae}}$ and $\sim_{\mathrm{sc}}$. □

With Propositions 25 and 27, we declare the active events rule and silent continuation rule under event preemption. Flordal and Malik (2009). For the active events rule, the following lemma is helproofful to clarify the proof.

**Lemma 28.** Let $G = \langle Q_G, A, \rightarrow_G, Q_G^\circ \rangle$ be a well-formed automaton with an equivalence $\sim \subseteq Q_G \times Q_G$ on $G$ be such that $\sim \subseteq \sim_{\mathrm{ae}}$. Then for any automaton $T = \langle Q_T, A, \rightarrow_T, Q_T^\circ \rangle$, if $([x_G], x_T) \overset{\sigma}{\Rightarrow}^{\mathcal{S}}$ in $\mathcal{S}(G/{\sim} \parallel T)$ for some $\sigma \in \Sigma$, then for all $x'_G \in [x_G]$, $(x'_G, x_T) \overset{\sigma}{\Rightarrow}^{\mathcal{S}}$ in $\mathcal{S}(G \parallel T)$.

*Proof.* Let $([x_G], x_T) \overset{\epsilon}{\Rightarrow}^{\mathcal{S}} ([\bar{x}_G], \bar{x}_T) \overset{\sigma}{\rightarrow}^{\mathcal{S}}$ in $\mathcal{S}(G/{\sim} \parallel T)$ for some $\bar{x}_G \in Q_G$ and $\bar{x}_T \in Q_T$. Note that all states on $[x_G] \overset{\epsilon}{\Rightarrow}_{\sim} [\bar{x}_G]$ before $[\bar{x}_G]$ in $G/{\sim}$ are singletons. The statement is thus directly true from Lemma 7 and Definition 20. □

We are now in the position to state two more conflict equivalent abstraction rules.

**Theorem 29** (active events rule)**.** Let $G = \langle Q, A, \rightarrow, Q^\circ \rangle$ be an $\Upsilon$-shaped automaton with an equivalence $\sim \subseteq Q \times Q$ on $G$ so that $\sim \subseteq \sim_{\mathrm{inc}} \cap \sim_{\mathrm{ae}}$. It holds that $G \simeq_{\mathcal{S}} (G/{\sim})$.

*Proof.* Let $T = \langle Q_T, A, \rightarrow_T, Q_T^\circ \rangle$ be any automaton:

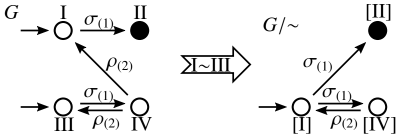

*Figure 5: active events rule*

($\Rightarrow$) Suppose $\mathcal{S}(G \parallel T)$ is non-blocking. Pick $([x_G], x_T)$ so that $\mathcal{S}(G/\!\sim \parallel T) \overset{s}{\Rightarrow}{}^{\mathcal{S}} ([x_G], x_T)$ for some $s \in \Sigma^*$. By Proposition 25, there exists $x'_G \in [x_G]$ so that $\mathcal{S}(G \parallel T) \overset{s}{\Rightarrow}{}^{\mathcal{S}} (x'_G, x_T)$ and due to the non-blockingness of $\mathcal{S}(G \parallel T)$, $(x'_G, x_T) \overset{t\omega}{\Longrightarrow}{}^{\mathcal{S}}$ in $\mathcal{S}(G \parallel T)$ for some $t \in \Sigma^*$. By Proposition 27, it holds that $([x_G], x_T) \overset{t\omega}{\Longrightarrow}{}^{\mathcal{S}}$.

($\Leftarrow$) Suppose $\mathcal{S}(G/\!\sim \parallel T)$ is non-blocking. Pick $(x_G, x_T)$ so that $\mathcal{S}(G \parallel T) \overset{s}{\Rightarrow}{}^{\mathcal{S}} (x_G, x_T)$ for some $s \in \Sigma^*$. From Proposition 27 and the non-blockingness of $\mathcal{S}(G/\!\sim \parallel T)$, there exists $t \in \Sigma^*$ so that $\mathcal{S}(G/\!\sim \parallel T) \overset{s}{\Rightarrow}{}^{\mathcal{S}} ([x_G], x_T) \overset{t\omega}{\Longrightarrow}{}^{\mathcal{S}}$. If $t = \epsilon$, then $(x_G, x_T) \overset{t\omega}{\Longrightarrow}{}^{\mathcal{S}}$ in $\mathcal{S}(G \parallel T)$ follows direct from Lemma 28. Otherwise, we must first have $([x_G], x_T) \overset{\epsilon}{\Rightarrow}{}^{\mathcal{S}} \overset{\sigma}{\rightarrow}{}^{\mathcal{S}}$ in $\mathcal{S}(G/\!\sim \parallel T)$ for some $\sigma \in \Sigma - \{\omega\}$. By applying Lemma 28, we have $(x_G, x_T) \overset{\epsilon}{\Rightarrow}{}^{\mathcal{S}} (\bar{x}_G, \bar{x}_T) \overset{\sigma}{\rightarrow}{}^{\mathcal{S}}$ in $\mathcal{S}(G \parallel T)$ for some $(\bar{x}_G, \bar{x}_T)$ where $\bar{x}_G \in Q_G$ and $\bar{x}_T \in Q_T$. By applying Proposition 27 and then Lemma 28 again, we have altogether $(x_G, x_T) \overset{\epsilon}{\Rightarrow}{}^{\mathcal{S}} (\bar{x}_G, \bar{x}_T) \overset{\sigma}{\rightarrow}{}^{\mathcal{S}} \overset{\epsilon}{\Rightarrow}{}^{\mathcal{S}} (y_G, y_T) \overset{\sigma'}{\longrightarrow}{}^{\mathcal{S}}$ in $\mathcal{S}(G \parallel T)$ for some $\sigma' \in \Sigma$ and $(y_G, y_T)$ where $y_G \in Q_G$ and $y_T \in Q_T$. From Proposition 26 and the non-blockingness of $\mathcal{S}(G/\!\sim \parallel T)$, there exists $y'_G \in [\bar{y}_G]$ so that $(y'_G, y_T) \overset{u\omega}{\Longrightarrow}{}^{\mathcal{S}}$ for some $u \in \Sigma^*$. If $[y_G]$ is singleton, the proof closes directly. Otherwise, from Definition 19, we have $\bar{x}_G \overset{\sigma}{\rightarrow} \underset{!n}{\overset{\epsilon}{\hookrightarrow}} y_G$ for some $n \geq 0$. Since $y'_G \sim y_G$, we have $\bar{x}_G \underset{\Delta:\sigma}{\overset{\epsilon}{\Longrightarrow}} \underset{\Delta:\sigma}{\overset{\sigma}{\longrightarrow}} \underset{!n}{\overset{\epsilon}{\hookrightarrow}} y'_G$ where $\Delta = G^{<\sigma}_{\mathrm{rglr}}(\bar{x}_G)$, implying $(\bar{x}_G, \bar{x}_T) \overset{\sigma}{\Rightarrow}{}^{\mathcal{S}} (y'_G, y_T)$ in $\mathcal{S}(G \parallel T)$ due to Proposition 24.(ii). This indeed closes the proof. □

**Theorem 30** (silent continuation rule)**.** Let $G = \langle Q_G, A, \rightarrow_G, Q^\circ_G \rangle$ be an $\Upsilon$-shaped automaton with an equivalence $\sim \subseteq Q \times Q$ on $G$ so that $\sim \subseteq \sim_{\mathrm{inc}} \cap \sim_{\mathrm{sc}}$. It holds that $G \simeq_{\mathcal{S}} (G/\!\sim)$.

*Proof.* For any $T = \langle Q_T, A, \rightarrow_T, Q^\circ_T \rangle$:

($\Rightarrow$) Same as the proof for Theorem 29

($\Leftarrow$) Suppose $\mathcal{S}(G/\!\sim \parallel T)$ is non-blocking. Pick $(x_G, x_T)$ so that $\mathcal{S}(G \parallel T) \overset{s}{\Rightarrow}{}^{\mathcal{S}} (x_G, x_T)$ for some $s \in \Sigma^*$. From Proposition 27 and the non-blockingness of $\mathcal{S}(G/\!\sim \parallel T)$, there exists $t \in \Sigma^*$ so that $\mathcal{S}(G/\!\sim \parallel T) \overset{s}{\Rightarrow}{}^{\mathcal{S}} ([x_G], x_T) \overset{t\omega}{\Longrightarrow}{}^{\mathcal{S}}$. Consider three cases:

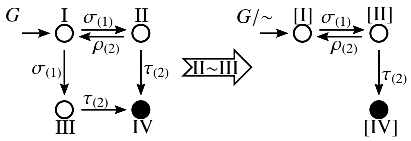

*Figure 6: active events rule*

(Case 1) $[x_G]$ is singleton and $([x_G], x_T) \overset{\sigma}{\rightarrow}{}^{\mathcal{S}}$ for some $\sigma \in \Sigma$. This case is directly true from Proposition 25 and 26.

(Case 2) $[x_G]$ is not singleton. Since $x_G$ is not in any live-lock but there exists some $\tau \in G_{\text{slnt}}(x_G)$, there must exist some $y_G \in Q_G$ so that $x_G \overset{\epsilon}{\Rightarrow} y_G$ and $G_{\text{slnt}}(y_G) = \emptyset$ in $G$. There are two further possibilities:

(i) There exists some $y_T \in Q_T$ and $\sigma \in \Sigma$ so that $(x_G, x_T) \overset{\epsilon}{\Rightarrow}{}^{\mathcal{S}} (y_G, y_T) \overset{\sigma}{\rightarrow}{}^{\mathcal{S}}$ in $\mathcal{S}(G \parallel T)$. Note that since $([y_G], y_T)$ is reachable in $\mathcal{S}(G/\!\sim \parallel T)$ which is nonblocking, it must be co-reachable as well. In addition, since $G_{\text{slnt}}(y_G) = \emptyset$, $[y_G]$ must be a singleton. Thus we have met a Case 1 situation.

(ii) If (i) does not hold, then there must exist some $z_G \in Q_G$ and $z_T \in Q_T$ so that

a) $x_G \overset{\epsilon}{\Rightarrow}{}^{\mathcal{S}} z_G \overset{\epsilon}{\Rightarrow}{}^{\mathcal{S}} y_G$ and $z_G \neq y_G$,

b) there exists some $\tau' \in G_{\text{slnt}}(z_G)$ and

c) $(x_G, x_T) \overset{\epsilon}{\Rightarrow}{}^{\mathcal{S}} (z_G, z_T) \overset{\sigma}{\rightarrow}{}^{\mathcal{S}}$ in $\mathcal{S}(G \parallel T)$ for some $\sigma \in \Sigma$ so that $\mathsf{prio}(\sigma) < \mathsf{prio}(\tau')$.

This again implies that $[z_G]$ is a singleton state from (SC1) and (SC2), i.e. a Case 1 situation is met.

(Case 3) $[x_G]$ is a singleton but $([x_G], x_T) \overset{\alpha}{\rightarrow}{}^{\mathcal{S}}$ for any $\alpha \in A$ implies $\alpha \in \Upsilon$. If in $([x_G], x_T) \overset{t\omega}{\Rightarrow}{}^{\mathcal{S}}$, each state consists of a singleton state from $Q_G/\!\sim$, the statement is trivially true. Else, let $([\bar{x}_G], \bar{x}_T) \overset{\alpha}{\rightarrow}{}^{\mathcal{S}} ([y_G], y_T)$ be the first transition in $([x_G], x_T) \overset{t\omega}{\Rightarrow}{}^{\mathcal{S}}$ so that $\alpha \in A$ and $[y_G]$ is not a singleton. Clearly, this transition is driven by $G$ since $[\bar{x}_G]$ is singleton and from Lemma 7, since $[\bar{x}_G]$ is singleton, there exists $y'_G \in [y_G]$ so that $(\bar{x}_G, \bar{x}_T) \overset{\alpha}{\rightarrow}{}^{\mathcal{S}} (y'_G, y_T)$ in $\mathcal{S}(G \parallel T)$. This indicates that Case 3 always reaches a Case 2 situation if at least one non-singleton state in $G/\!\sim$ can be reached. □

While the two previous abstraction rules requiured a careful adaptation to event priorities, the following two carry over immediately from Flordal and Malik (2009).

**Theorem 31** (only silent incoming rule)**.** Let $G = \langle Q, A, \rightarrow, Q^\circ \rangle$ be an $\Upsilon$-automaton and let $\bar{x} \in Q$ be such that $\bar{x}$ is not in any live-lock, $\bar{x} \xrightarrow{\tau_{(1)}}$ and $y \overset{\alpha}{\rightarrow} \bar{x}$ implies $\alpha = \tau_{(1)}$. Then for the automaton $H = \langle Q, \Sigma, \rightarrow_H, Q^\circ \rangle$ with

$$\rightarrow_H = \{(x, \alpha, y) \mid x \overset{\alpha}{\rightarrow} y \text{ and } y \neq \bar{x}\} \cup \{(x, \alpha, y) \mid x \xrightarrow{\tau_{(1)}} \bar{x} \overset{\alpha}{\rightarrow} y\} \qquad (13)$$

it holds that $G \simeq_{\mathcal{S}} H$. □

**Theorem 32** (only silent outgoing rule)**.** Let $G = \langle Q, A, \to, Q^\circ \rangle$ be a $\Upsilon$-shaped automaton and let $\bar{x} \in Q$ be such that $\bar{x} \xrightarrow{\alpha} y$ implies $\alpha = \tau_{(1)}$ and $y$ is not in any live-lock, while $z \xrightarrow{\alpha'} \bar{x}$ implies $\alpha' \notin \Upsilon$. Let $\bar{Q} := \{y \in Q \,|\, \bar{x} \xrightarrow{\alpha} y \text{ for any } \alpha \in A\}$, then for the automaton $H = \langle Q \backslash \{\bar{x}\}, \Sigma, \to_H, Q^\circ_H \rangle$ with

$$Q^\circ_H = \begin{cases} Q^\circ & \text{if } \bar{x} \notin Q^\circ \\ (Q^\circ \backslash \{\bar{x}\}) \cup \bar{Q} & \text{if } \bar{x} \in Q^\circ \end{cases} \tag{14}$$

$$\begin{aligned} \to_H =& \{(x, \alpha, y) \,|\, x \xrightarrow{\alpha} y \text{ and } x \neq \bar{x} \text{ and } y \neq \bar{x}\} \\ & \cup \{(x, \alpha, y) \,|\, x \xrightarrow{\alpha} \bar{x} \text{ and } y \in \bar{Q}\} \end{aligned} \tag{15}$$

it holds that $G \simeq_{\mathcal{S}} H$. □

## Conclusion

Considering a class of modular discrete-event systems with event priorities, we have presented a number of conflict equivalent abstractions to be used for compositional verification of non-blockingness. Technically, our study builds on Flordal and Malik (2009) and we inspect the abstraction rules presented there in order to derive adaptations that address priorities. This approach is closely related to the development of CCS$^{\text{ch}}$ by Lüttgen (1998) which introduces priorities to process algebra CCS (Milner (1989)).

## References

Cassandras, C.G. and Lafortune, S. (2008). *Introduction to Discrete Event Systems*. Springer, second edition.

Flordal, H. and Malik, R. (2009). Compositional verification in supervisory control. *SIAM J. Control and Optimization*, 48, 1914–1938.

Lüttgen, G. (1998). Pre-emptive modeling of concurrent and distributed systems.

Malik, R., Streader, D., and Reeves, S. (2004). Fair testing revisited: A process-algebraic characterisation of conflicts. In F. Wang (ed.), *Automated Technology for Verification and Analysis*, 120–134. Springer.

Milner, R. (1989). *Communication and Concurrency*. Prentice-Hall, Inc., USA.

Mohajerani, S., Malik, R., and Fabian, M. (2014). A framework for compositional synthesis of modular nonblocking supervisors. *IEEE Transactions on Automatic Control*, 59(1), 150–162.

Nicola, R.D. and Hennessy, M. (1984). Testing equivalences for processes. *Theoretical Computer Science*, 34(1), 83 – 133.

Pilbrow, C. and Malik, R. (2015). An algorithm for compositional nonblocking verification using special events. *Science of Computer Programming*, 113.