Ready Simulation for Concurrency: It's Logical! [⋆]

Gerald Lüttgen [1]

Software Technologies Research Group, University of Bamberg, D-96045 Bamberg, Germany

Walter Vogler [2]

Institut für Informatik, Universität Augsburg, D-86135 Augsburg, Germany

Abstract

This article provides new insight into the connection between the trace-based lower part of van Glabbeek's linear-time, branching-time spectrum and its simulation-based upper part. We establish that ready simulation is fully abstract with respect to failure inclusion, when adding the conjunction operator that was proposed by the authors in [TCS 373(1–2):19–40] to the standard setting of labelled transition systems with (CSP-style) parallel composition. More precisely, we actually prove a stronger result by considering a coarser relation than failure inclusion, namely a preorder that relates processes with respect to inconsistencies that may arise under conjunctive composition.

Ready simulation is also shown to satisfy standard logic properties. In addition, our semantic formalism proves itself robust when adding disjunction, external choice and hiding operators, and is thus suited for studying mixed operational and logic languages. Finally, the utility of our formalism is demonstrated by means of a small example that deals with specifying and reasoning about mode logics within aircraft control systems.

Key words: Logic labelled transition system, conjunction, parallel composition, consistency preorder, ready simulation, full abstraction.

[⋆] An extended abstract appeared in C. Cachin, T. Jurdziński and A. Tarlecki, eds., 34th Intl. Coll. on Automata, Languages and Programming (ICALP 2007), vol. 4596 of Lecture Notes in Computer Science, pp. 752–763, 2007. Springer.

[1] Email: gerald.luettgen@swt-bamberg.de. The first author was partially supported by the EPSRC under grant no. EP/E034853/1.

[2] Corresponding author. Email: walter.vogler@informatik.uni-augsburg.de; phone: +49 821 598-2120; fax: +49 821 598-2175.

1 Introduction

Basic research in concurrency theory over the past 25 years has resulted in a wealth of process algebras [1–3] and temporal logics [4] for specifying and reasoning about concurrent processes. However, little research has been conducted on mixing process-algebraic and logic styles of specification in a single formalism. This is surprising since many popular software-engineering languages, including UML, permit such mixed specifications.

In [5,6] we proposed an approach to defining and reasoning about conjunction on labelled transition systems. Our setting consisted of τ-pure labelled transition systems — i.e., LTSs where each state encodes either an external or internal (disjunctive) choice between its outgoing transitions — and augmented by an *inconsistency predicate* (Logic LTS, cf. Sec. 2). While our conjunction operator is in essence a synchronous product on visible actions and an interleaving product on internal actions, the challenge was in dealing with inconsistencies. Inconsistencies may either arise when conjunctively composing two processes with different initial action sets (i.e., *ready sets*), or when a process has no other choice for some action than entering an inconsistent state. Our framework was equipped with *ready-tree semantics*, which is a variant of van Glabbeek's path-based possible-worlds semantics [7] that was inspired by Veglioni and De Nicola [8]. The resulting ready-tree preorder turned out to be coarser than ready simulation and finer than failure inclusion (for divergence-free systems) and ready-trace inclusion, which implies that ready-tree semantics is sensitive to deadlock. We proved in [6] that the ready-tree preorder is fully abstract under conjunction with respect to a naive *inconsistency preorder*,[3] which allows an inconsistent specification only to be implemented by an inconsistent implementation.

This article first shows that the ready-tree preorder is inadequate in the presence of concurrency, as it fails to be compositional for standard parallel composition, such as the parallel operator of CSP [9] and LOTOS [10],while it *is* a precongruence for parallel composition in the special case that all actions are synchronised [6]. A different compositionality problem for the parallel composition of SCCS was already noted in [7]. We then establish our main result (cf. Sec. 3), namely that *ready simulation* [11], which adds to ordinary simulation the requirement that related processes must have identical ready sets, is fully abstract with respect to conjunction *and* parallel composition, for labelled transition systems with inconsistencies. The proof of this result uses our earlier full-abstraction result involving ready trees. Along the way, we adapt ready simulation to dealing with internal actions and inconsistencies.

[3] I.e., the ready-tree preorder is the *coarsest precongruence* for conjunction which refines the inconsistency preorder.

We also conduct several sanity checks on our framework; in particular, we verify that our conjunction operator indeed formalises *conjunction* regarding ready simulation.

In addition to including all proofs and more explanatory text when compared to the conference version on which this article is based, we firstly apply our framework to a small example and secondly extend it by further logical and process-algebraic operators. The example involves specifying and reasoning about a simple mode logic for an aircraft control system, and highlights the practical utility of Logic LTS and ready simulation for system design. The additional operators we introduce are disjunction, external choice and hiding, for which ready simulation is shown to be compositional, too. The inclusion of disjunction also allows us to establish several standard logic properties desired of ready simulation, including the distributivity laws between conjunction and disjunction. Hiding proves to be challenging to define, since the usual, straightforward definition of hiding violates τ-purity.

Our full-abstraction result provides an interesting insight into van Glabbeek's linear time–branching time spectrum [7], namely that conjunction on processes is a tool, via full abstraction, for relating the trace-based lower part of the spectrum to the simulation-based upper part. In addition, our results and our example testify to the robustness of our technical framework and to the adequacy of ready simulation as the semantic basis for mixed process-algebraic and logic languages. Indeed, ready simulation eliminates the necessity for restrictions on the nesting of process-algebraic and logic constructs, such as the one employed by Olderog when embedding trace formulas into CSP [12].

The remainder of this article is organised as follows. The next section introduces our setting of Logic LTS, as well as our conjunction and parallel composition operators. Sec. 3 establishes the aforementioned full abstraction result and thus is the key section of this article. Our theory is applied to an example in Sec. 4, and extended by disjunction, external choice and hiding operators in Sec. 5. Finally, Secs. 6 and 7 discuss some related work and present our conclusions, respectively.

2 Logic LTS, conjunction & parallel composition

This section recalls the definitions of *Logic Labelled Transition Systems*, or Logic LTS for short, and the conjunction operator on Logic LTS which were introduced in [6]. It also lifts the parallel composition operator in the style of CSP [9] and LOTOS [10] to Logic LTS.

Key to our setting is the consideration of *inconsistencies* that may arise under conjunctive composition. The intuitive idea behind inconsistency is twofold:

(1) Processes p and q are *consistent* if and only if they have a common implementation. This principle was also adopted by Steen et al [13] for consistency in the context of partial process specifications.
(2) A *stable* process r, where r cannot perform the internal, unobservable action τ, cannot be an implementation of a stable process p if one offers an action that the other cannot perform, i.e., if r and p have different *ready sets* [14]. This property is satisfied, e.g., in failure semantics [9].

The reason for the second item is that the absence of deadlock is an important feature of a concurrent system, which we want to preserve when replacing p inside a parallel composition by its implementation r; this is also the justification behind failure semantics. If p offers action a while r does not, and the system environment only offers a, then replacing p by r leads to an immediate deadlock. Vice versa, if r offers an action a while p does not, then replacing p by r can allow new behaviour of the environment and new deadlocks.

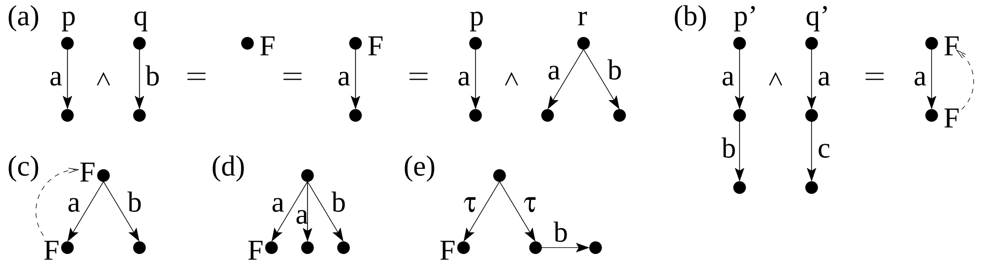

Fig. 1. Basic intuition behind conjunctive composition.

Hence, it is immediately clear that the conjunction of stable processes p and q with different ready sets is inconsistent, and we will mark $p \wedge q$ as such. As an example consider the processes p, q and r of Fig. 1(a). Process p and q specify that exactly action a and b, respectively, is offered initially, i.e., their ready sets are $\{a\}$ and $\{b\}$, respectively. Similarly, process r specifies that a and b are offered initially and thus has ready set $\{a, b\}$. Therefore, $p \wedge q$ as well as $p \wedge r$ are *inconsistent* (or *false*) and tagged accordingly. Formally, our variant of LTS will be augmented by an *inconsistency predicate*, or false-predicate, F, so that $p \wedge q$, $p \wedge r \in F$ in our example.

Inconsistency is more tricky, however, as it can propagate backwards along transitions. For example, in the conjunction $p' \wedge q'$ shown in Fig. 1(b), both conjuncts require action a to be performed, whence $p' \wedge q'$ should have an a-transition. But this transition does lead to an inconsistent state and, in the absence of any alternative a-transition leading to a consistent state, $p' \wedge q'$ must itself be considered as inconsistent. Therefore, inconsistency propagates backwards for the processes in Fig. 1(b) and (c), whereas it does not for the processes in Figs. 1(d) and 1(e), as they can engage in an a- and τ-transition, respectively, leading to a consistent state. As an aside, the process in Fig. 1(e) may be interpreted as a disjunction between the inconsistent process marked F which has empty behaviour, and the consistent process offering a b-transition.

2.1 Logic Labelled Transition Systems

Let $\mathcal{A}$ be an alphabet with representatives a and b, and let $\mathcal{A}_\tau$ denote $\mathcal{A} \cup \{\tau\}$ with representatives α and β. An LTS is a triple $\langle P, \longrightarrow, F\rangle$,[4] where P is the set of *processes* (states), $\longrightarrow \subseteq P \times \mathcal{A}_\tau \times P$ is the *transition relation*, and $F \subseteq P$ is the *inconsistency predicate*. We write (i) $p \xrightarrow{\alpha} p'$ instead of $\langle p, \alpha, p'\rangle \in \longrightarrow$, (ii) $p \xrightarrow{\alpha}$ instead of $\exists p' \in P.\, p \xrightarrow{\alpha} p'$ and (iii) $p \longrightarrow$ instead of $\exists p' \in P,\, \alpha \in \mathcal{A}_\tau.\, p \xrightarrow{\alpha} p'$. When $p \xrightarrow{\alpha} p'$, we say that process p can perform an α-step to p', and we call p' an α-derivative. $\mathcal{I}(p)$ stands for the *ready set* $\{\alpha \in \mathcal{A}_\tau \,|\, p \xrightarrow{\alpha}\}$ of process p. A process p that cannot engage in a τ-transition, i.e., $p \not\xrightarrow{\tau}$, is called *stable*. The *sort* $\mathcal{A}_P$ of the LTS (and its processes) is the set of actions occurring in $\longrightarrow$.

We also require an LTS to satisfy the following *τ-purity* condition: $p \xrightarrow{\tau} \implies \nexists a \in \mathcal{A}.\, p \xrightarrow{a}$, for all $p \in P$. Hence, each process represents either an external or internal (disjunctive) choice between its outgoing transitions. This restriction reflects the fact that ready sets can only be observed at stable states, so that visible transitions leaving instable states are outside our observation. We discuss the need for τ-purity later; see Fig. 3 and the paragraphs thereafter.

The LTSs of interest to us need to satisfy two further properties. Before we can present them, we first need to define our notion of when a process *can stabilise*. To do so, we introduce several variants of weak transition relations which will prove useful in the sequel. We write $p \overset{\epsilon}{\Longrightarrow} p'$ if $p \xrightarrow{\tau}{}^{*} p'$ and $p \overset{\alpha}{\Longrightarrow} p'$ if $\exists p'', p'''.\, p \overset{\epsilon}{\Longrightarrow} p'' \xrightarrow{\alpha} p''' \overset{\epsilon}{\Longrightarrow} p'$.[5]. If all processes along a computation $p \overset{\epsilon}{\Longrightarrow} p'$ or $p \overset{a}{\Longrightarrow} p'$, including p and p', are consistent, then we write $p \overset{\epsilon}{\Longrightarrow}_{\mathrm{F}} p'$ and $p \overset{a}{\Longrightarrow}_{\mathrm{F}} p'$, respectively. If in addition, p' is stable, we write $p \overset{\epsilon}{\Longrightarrow}\!| \, p'$ and $p \overset{a}{\Longrightarrow}\!| \, p'$, respectively. These notions were first introduced in [5], and the subscript F is intended to remind the reader that the transition relation takes (in-)consistency into account.

We may now define that a process p *can stabilise* if $\exists p'.\, p \overset{\epsilon}{\Longrightarrow}\!| \, p'$.

Definition 1 (Logic LTS [6]) *An LTS $\langle P, \longrightarrow, F\rangle$ is a* Logic LTS *if:*

(LTS1) $p \in F$ *if* $\exists \alpha \in \mathcal{I}(p)\, \forall p' \in P.\, p \xrightarrow{\alpha} p' \implies p' \in F$;
(LTS2) p *cannot stabilise* $\implies p \in F$.

[4] The additional, less relevant *true predicate* of [6] is omitted here for clarity.
[5] A double arrow without action label still stands for mathematical implication

The first condition formalises the backward propagation of inconsistencies as discussed above. The second condition relates to *divergence*, i.e., infinite sequences of τ-transitions, where divergence is viewed as catastrophic if a process cannot *stabilise*, as is discussed in [6, Fig. 3].

As notational conventions, we will denote a transition $p \xrightarrow{\alpha} p'$ with $p, p' \notin F$ by $p \xrightarrow{\alpha}_{\mathrm{F}} p'$. Moreover, whenever we mention a process p without stating a respective Logic LTS explicitly, we assume implicitly that such a Logic LTS $\langle P, \longrightarrow, F\rangle$ is given. Finally, we let *ff* stand for the only process of the LTS $\langle\{\mathit{ff}\}, \emptyset, \{\mathit{ff}\}\rangle$; *ff* represents the boolean constant *false*. Intuitively, any given process is either inconsistent, in which case it is equivalent to *ff*, or it is equivalent to a process from which no inconsistent process can be reached; the latter can simply be achieved by omitting inconsistent processes in LTSs and all transitions leading to them.

2.2 Conjunction & parallel composition

Our conjunction operator is a synchronous product for visible transitions and an asynchronous product for τ-transitions, analogous to $\|_{\mathcal{A}}$ defined below. However, we need to take care of inconsistencies. This is because, otherwise, $p \wedge q$, with p and q defined as in Fig. 1(a), would erroneously be the “deadlock” process, i.e., ‘the’ consistent process without any transitions.

Definition 2 (Conjunction operator [6]) *The conjunction of Logic LTSs* $\langle P, \longrightarrow_P, F_P\rangle$ *and* $\langle Q, \longrightarrow_Q, F_Q\rangle$ *is the Logic LTS* $\langle P \wedge Q, \longrightarrow_{P\wedge Q}, F_{P\wedge Q}\rangle$*:*

- $P \wedge Q =_{df} \{p \wedge q \,|\, p \in P,\ q \in Q\}$
- $\longrightarrow_{P\wedge Q}$ *is determined by the following operational rules:*

$$\begin{array}{rcl}
p \xrightarrow{\tau}_P p' & \Longrightarrow & p \wedge q \xrightarrow{\tau}_{P\wedge Q} p' \wedge q \\
q \xrightarrow{\tau}_Q q' & \Longrightarrow & p \wedge q \xrightarrow{\tau}_{P\wedge Q} p \wedge q' \\
p \xrightarrow{a}_P p',\ q \xrightarrow{a}_Q q' & \Longrightarrow & p \wedge q \xrightarrow{a}_{P\wedge Q} p' \wedge q'
\end{array}$$

- $F_{P\wedge Q}$ *is the least set containing each* $p \wedge q$ *that satisfies at least one of the following conditions:*
  (C1) $p \in F_P$ *or* $q \in F_Q$*;*
  (C2) $p \wedge q \not\xrightarrow{\tau}_{P\wedge Q}$ *and* $\mathcal{I}(p) \neq \mathcal{I}(q)$*;*
  (C3) $\exists \alpha \in \mathcal{I}(p \wedge q)\, \forall p' \wedge q'.\ p \wedge q \xrightarrow{\alpha}_{P\wedge Q} p' \wedge q' \implies p' \wedge q' \in F_{P\wedge Q}$*;*
  (C4) $p \wedge q$ *cannot stabilise.*

We are left with explaining Conds. (C1)–(C4). Firstly, a conjunction is inconsistent if a conjunct is inconsistent. Conds. (C2) and (C3) reflect our intuition of inconsistency and backward propagation. Cond. (C4) is added to enforce (LTS2); note that this condition is not automatically enforced since it is *not* true that $p \wedge q$ cannot stabilise if both p and q cannot stabilise. To see this, consider the processes p and q of Fig. 2 and the resulting Logic LTS for $p \wedge q$; it is only because of the inclusion of Cond. (C4) that state $p \wedge q$ of this Logic LTS is tagged as inconsistent.

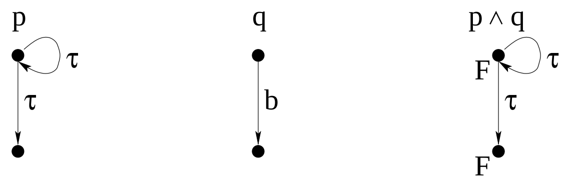

Fig. 2. Discussing the need for Cond. (C4).

It is easy to check that conjunction is well-defined, i.e., that the conjunctive composition of two Logic LTSs is indeed a Logic LTS; see [6] for details. In the sequel, we leave out indices of relations and predicates whenever the context is clear.

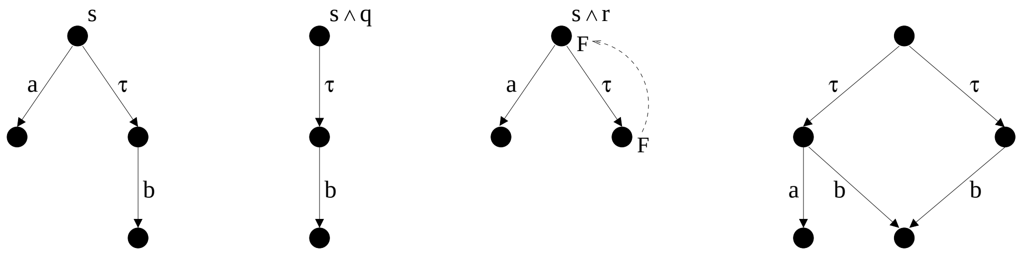

Fig. 3. Discussing the need for τ-purity.

Given this definition of conjunction we can briefly discuss our adoption of τ-purity. To see what happens if we allow *impure* processes, consider process s of Fig. 3, which violates τ-purity, and processes q and r of Fig. 1(a). The question is how we should possibly modify Cond. (C2) of Def. 2 so as to define immediate inconsistency when dealing with s. Since our justification for Cond. (C2) at the beginning of this section only applies to stable processes and since we only want to tag a process as inconsistent if necessary, the most generous possibility would be to leave Cond. (C2) unchanged, i.e., never to tag processes like $s \wedge q$ or $s \wedge r$ as immediately inconsistent.

With this approach, $s \wedge q$ and $s \wedge r$ are the processes shown in Fig. 3: $s \wedge q$ is consistent — in accordance with failure semantics where q is an implementation of s — but $s \wedge r$ is inconsistent by backward propagation. The latter effect occurs with any stable process r offering a, whence this approach treats s almost as if there was no a-transition. Although this might be technically feasible, we think the effect is counter-intuitive; indeed, r is also an implementation of s according to failure semantics.

We therefore insist on τ-purity here and will show that this leads to a setting that satisfies all desirable properties: full abstraction, the fact that our conjunction operator indeed captures conjunction, and logic properties expected of conjunction. Later, in Sec. 5.3 where we discuss hiding, impure processes like s could naturally occur; we will come back to the above discussion then. Finally, it should already be noted here that we regard s as the disjunction of q and r (cf. Sec. 5.1). Hence, we can express s in our setting as the process on the right-hand side of Fig. 3, and similarly for any other impure process. Thus, we regard the restriction to pure processes as semantically irrelevant. Moreover, τ-purity is equivalent to giving τ-transitions precedence over visible transitions, as we will discuss on the next page after defining parallel composition.

We now turn to introducing the proof tool of *witness* which will turn out to be convenient when reasoning about inconsistency within conjunction:

Definition 3 ($\wedge$-Witness) *An* $\wedge$-witness *is a set $W \subseteq P \wedge Q$ such that, for all $p \wedge q \in W$, the following conditions hold:*

(AW1) *$p, q \notin F$;*
(AW2) *$p \xrightarrow{\tau}$ or $q \xrightarrow{\tau}$ or $\mathcal{I}(p) = \mathcal{I}(q)$;*
(AW3) *$\forall \alpha \in \mathcal{I}(p \wedge q)\, \exists p' \wedge q' \in W.\ p \wedge q \xrightarrow{\alpha} p' \wedge q'$;*
(AW4) *$p \wedge q$ can stabilise in W, i.e., $p \wedge q \xrightarrow{\tau} p_1 \wedge q_1 \xrightarrow{\tau} \cdots \xrightarrow{\tau} p_n \wedge q_n \not\xrightarrow{\tau}$ with all $p_i \wedge q_i \in W$.*

The following straightforward property of $\wedge$-witnesses gives us a useful tool for proving that a conjunction of processes is consistent:

Proposition 4 *$p \wedge q \notin F_{P \wedge Q}$ if and only if $\exists \wedge$-witness $W.\ p \wedge q \in W$.*

PROOF. Direction "$\Longrightarrow$" follows from the fact that $\overline{F_{P \wedge Q}}$, i.e., the complement of $F_{P \wedge Q}$, is an $\wedge$-witness. For direction "$\Longleftarrow$" we note that $\overline{W}$ satisfies the conditions of $F_{P \wedge Q}$, whence $F_{P \wedge Q} \subseteq \overline{W}$. $\Box$

For example, the concept of $\wedge$-witness may be employed to prove the second statement of the following lemma:

Lemma 5 *(1) If $p \wedge q \xrightarrow{\tau} p' \wedge q' \notin F$ and $p, q \notin F$, then $p \wedge q \notin F$.*
(2) Let $p \overset{\epsilon}{\Longrightarrow}\!| p'$, $q \overset{\epsilon}{\Longrightarrow}\!| q'$ and $p' \wedge q' \notin F$. Then, $p \wedge q \overset{\epsilon}{\Longrightarrow}\!| p' \wedge q'$.

PROOF. As Part (1) is not difficult, we focus on proving Part (2). Obviously, we can combine the given computations to get $p \wedge q \overset{\epsilon}{\Longrightarrow} p' \wedge q'$ with $p' \wedge q'$ stable. It remains to be shown that no process along this combined computation is inconsistent. To do so, we define W'' as the set of processes along the combined computation, except the last one, and prove that $W'' \cup \overline{F}$ is an $\wedge$-witness (with

$F = F_{P \wedge Q}$). Since $\overline{F}$ is an $\wedge$-witness, it is sufficient to check (AW1)–(AW4) for the elements of W''. Cond. (AW1) holds due to $p \overset{\epsilon}{\Longrightarrow}| \, p'$ and $q \overset{\epsilon}{\Longrightarrow}| \, q'$. To see the validity of (AW2) and (AW3), observe that all processes in W'' are instable and can perform a τ-transition to reach a process in $W'' \cup \{p' \wedge q'\} \subseteq W'' \cup \overline{F}$. Finally, the computation $p \wedge q \overset{\epsilon}{\Longrightarrow} p' \wedge q'$ shows that the processes in W'' can stabilise in $W'' \cup \overline{F}$, whence (AW4) holds. $\square$

Finally, we adapt the parallel operator $\|_A$ of CSP [2] to our setting, where $A \subseteq \mathcal{A}$ denotes the *synchronisation alphabet*. Naturally, the parallel composition of two processes is inconsistent if either process is inconsistent.

Definition 6 (Parallel operator) *The parallel composition of Logic LTS $\langle P, \longrightarrow_P, F_P \rangle$ and $\langle Q, \longrightarrow_Q, F_Q \rangle$ for the synchronisation set $A \subseteq \mathcal{A}$, is the Logic LTS $\langle P \|_A Q, \longrightarrow_{P\|_A Q}, F_{P\|_A Q} \rangle$:*

- *$P \|_A Q =_{df} \{p \|_A q \,|\, p \in P,\, q \in Q\}$*
- *$\longrightarrow_{P\|_A Q}$ is determined by the following operational rules:*

$$\begin{array}{rcl}
p \xrightarrow{\alpha}_P p',\ \alpha \notin A,\ (\alpha = \tau \text{ or } q \not\xrightarrow{\tau}_Q) & \Longrightarrow & p \|_A q \xrightarrow{\alpha}_{P\|_A Q} p' \|_A q \\
q \xrightarrow{\alpha}_Q q',\ \alpha \notin A,\ (\alpha = \tau \text{ or } p \not\xrightarrow{\tau}_P) & \Longrightarrow & p \|_A q \xrightarrow{\alpha}_{P\|_A Q} p \|_A q' \\
p \xrightarrow{a}_P p',\ q \xrightarrow{a}_Q q',\ a \in A & \Longrightarrow & p \|_A q \xrightarrow{a}_{P\|_A Q} p' \|_A q'
\end{array}$$

- *$p \|_A q \in F_{P\|_A Q}$ if $p \in F_P$ or $q \in F_Q$.*

Also parallel composition is well-defined, i.e., the parallel composition of two Logic LTSs satisfies the conditions of Def. 1. To preserve τ-purity, the first two of the above operational rules include the premise ($\alpha = \tau$ or $q \not\xrightarrow{\tau}_Q$) and ($\alpha = \tau$ or $p \not\xrightarrow{\tau}_P$), respectively, which gives τ-transitions precedence over visible transitions.

As an aside, rather than demanding τ-purity in the definition of LTS, one could equivalently permit τ-impurity but enforce priority of τ by defining $\mathcal{I}(p)$ as $\{\alpha \in \mathcal{A}_\tau \,|\, p \xrightarrow{\alpha}$ and $(\alpha = \tau$ or $p \not\xrightarrow{\tau})\}$, modifying $\overset{\alpha}{\Longrightarrow}$ accordingly, and possibly changing the use of $\xrightarrow{\alpha}$. Although this version would allow one to drop the two premises mentioned above, we prefer to require τ-purity explicitly since we consider this to be more transparent and technically easier.

2.3 Ready-tree semantics

This section defines our reference preorder, the *inconsistency preorder*, and also recalls further definitions from an earlier paper [6], including *ready trees*. Our previous work [6] focused only on studying conjunction on Logic LTSs. It characterised the largest precongruence contained in the inconsistency preorder, which states that a consistent implementation p never refines an inconsistent specification q.[6] On this, our new full-abstraction result in Sec. 3 is based.

Definition 7 (Inconsistency preorder [6]) *The* inconsistency preorder $\sqsubseteq_F$ *on processes is defined by* $p \sqsubseteq_F q$ *if* $p \notin F \implies q \notin F$.

This definition directly encodes the standard verification question whether an implementation refines its specification. When reading 'refines' logically as 'implies', it is clear that an inconsistent (i.e., 'false') specification can only be met by an inconsistent implementation.

Obviously, the inconsistency preorder is not compositional with respect to conjunction. Our characterisation of the fully-abstract preorder contained in $\sqsubseteq_F$ and presented in [6] is founded on a variant of the path-based possible-worlds semantics of [7,8], to which we refer as *ready-tree semantics*. This semantics employs the notion of *observation tree*. An observation tree is a Logic LTS $\langle V, \longrightarrow, \emptyset \rangle$ whose processes and transitions form a *deterministic tree* and whose processes (vertices) are stable; we refer to the tree's root as v_0. We may now formalise our desired observations of a process p, called *ready trees*:

Definition 8 (Ready tree [6]) *An observation tree* v_0 *is a* ready tree of p, *if there is a labelling* $h : V \longrightarrow P$ *satisfying the following conditions:*

(RT1) $\forall v \in V.\ h(v)$ *stable and* $h(v) \notin F_P$*;*
(RT2) $p \overset{\epsilon}{\Longrightarrow}\!| \, h(v_0)$*;*
(RT3) $\forall v \in V,\ a \in \mathcal{A}.\ v \xrightarrow{a} v' \implies h(v) \overset{a}{\Longrightarrow}\!| \, h(v')$*;*
(RT4) $\forall v \in V. \mathcal{I}(v) = \mathcal{I}(h(v))$.

Intuitively, nodes v in a ready tree represent stable states $h(v)$ of p (cf. the first part of Cond. (RT1)) and transitions represent stable, consistent computations (cf. Cond. (RT3)). Since such computations do not contain inconsistent states, no represented state is allowed to be in F (cf. the second part of Cond. (RT1)). Since p might not be stable, the root v_0 of a ready tree represents a stable process reachable from p via some internal computation (cf. Cond. (RT2)). Moreover, v must mimic the ready set of $h(v)$ (cf. Cond. (RT4)). In the following, we write RT(p) for the set of all ready trees of p; note that *ff* has no ready tree.

Definition 9 (Ready-tree preorder [6]) *The* ready-tree preorder $\subseteq_{RT}$ *on processes is defined as ready-tree inclusion, i.e.,* $p \subseteq_{RT} q$ *if* $RT(p) \subseteq RT(q)$.

[6] The reader familiar with [6] should note that we now write the implementation to the left and the specification to the right of the preorder symbol, in order to be consistent with the notational conventions of simulation-based preorders.

Theorem 10 (Full-abstraction wrt. conjunction [6]) *$\subseteq_{RT}$ is the largest precongruence in $\sqsubseteq_F$, when considering conjunction as the only operator.*

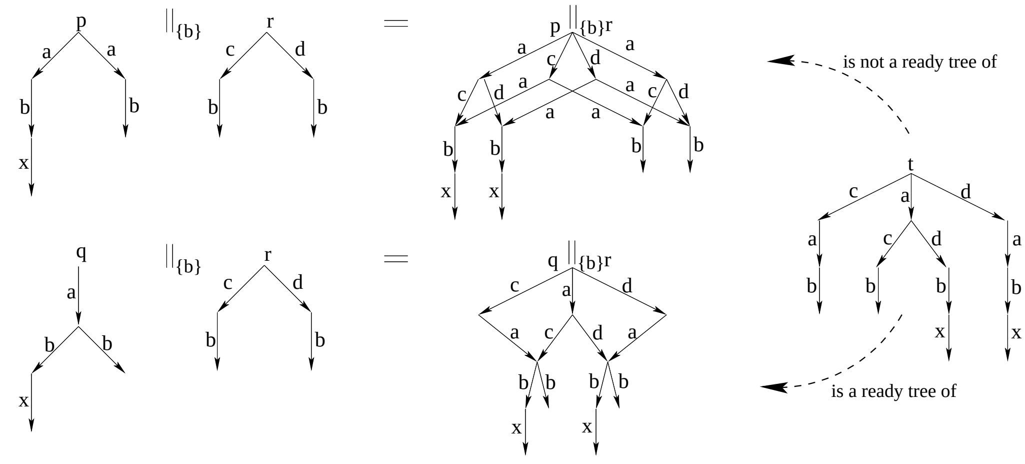

Fig. 4. Ready-tree semantics is not compositional for parallel composition.

Unfortunately, $\subseteq_{\mathrm{RT}}$ is *not* a precongruence for parallel composition $\|_A$, which makes the preorder unsuitable for reasoning about concurrency. To see this, consider the Logic LTSs p, q and r of Fig. 4. Here, p and q have the same ready trees, but t is a ready tree of $q \;\|_{\{b\}}\; r$ but not of $p \;\|_{\{b\}}\; r$. This non-compositionality result was initially a bit of a surprise to us, given that we had proved in [6] that the ready-tree preorder is compositional for the fully-synchronous product, which is a special case of $\|_A$ when taking $A = \mathcal{A}$.

3 Full abstraction via ready simulation

We now establish our full-abstraction result of ready simulation wrt. the inconsistency preorder, when considering both conjunction and parallel composition. In addition, we prove that our conjunction operator $\wedge$ possesses the desired compositionality and logic properties.

3.1 Ready simulation and compositionality

We start off with a definition of ready simulation [11] for Logic LTS:

Definition 11 (Ready simulation on Logic LTS) *Let $\langle P, \longrightarrow_P, F_P\rangle$ and $\langle Q, \longrightarrow_Q, F_Q\rangle$ be two Logic LTS. A relation $\mathcal{R} \subseteq P \times Q$ is a* stable ready simulation relation*, if the following conditions hold, for any $\langle p, q\rangle \in \mathcal{R}$ and $a \in \mathcal{A}$:*

(RS1) *p, q stable;*

(RS2) $p \notin F_P \implies q \notin F_Q$;
(RS3) $p \overset{a}{\Longrightarrow}| p' \implies \exists q'.\, q \overset{a}{\Longrightarrow}| q'$ *and* $\langle p', q' \rangle \in \mathcal{R}$;
(RS4) $p \notin F_P \implies \mathcal{I}(p) = \mathcal{I}(q)$.

We say that p *is* stable ready simulated *by* q*, in symbols* $p \underset{\sim}{\sqsubset}_{RS} q$*, if there exists a stable ready simulation relation* $\mathcal{R}$ *with* $\langle p, q \rangle \in \mathcal{R}$*. Further,* p *is* ready simulated *by* q*, written* $p \sqsubseteq_{RS} q$*, if* $\forall p'.\, p \overset{\epsilon}{\Longrightarrow}| p' \implies \exists q'.\ q \overset{\epsilon}{\Longrightarrow}| q'$ *and* $p' \underset{\sim}{\sqsubset}_{RS} q'$ *(*root condition*). We write* $\approx_{RS}$ *and* $=_{RS}$ *for the kernel of* $\underset{\sim}{\sqsubset}_{RS}$ *and* $\sqsubseteq_{RS}$*, respectively.*

It is easy to see that $\underset{\sim}{\sqsubset}_{\mathrm{RS}}$ and $\sqsubseteq_{\mathrm{RS}}$ are preorders, and that $p \sqsubseteq_{\mathrm{RS}} q$ trivially holds if $p \in F$. Note that $\underset{\sim}{\sqsubset}_{\mathrm{RS}}$ is itself a stable ready simulation relation and the largest such relation. In addition, ready simulation $\sqsubseteq_{\mathrm{RS}}$ is contained in the ready-tree preorder $\subseteq_{\mathrm{RT}}$, as essentially stated in [7], and conjunction and parallel composition are associative and commutative with respect to $=_{\mathrm{RS}}$. Note that there are several ways how to define ready simulation [11,7] for settings with internal actions [15]. Our variant is an adaptation of van Glabbeek's *stability respecting ready simulation may preorder* to Logic LTS.

Van Glabbeek has pointed out to us the following shorter definition of $\sqsubseteq_{\mathrm{RS}}$, which essentially integrates the treatment of unstable processes into the definition of ready simulation relation:

Definition 12 (Alternative definition of ready simulation)
Let $\langle P, \longrightarrow_P, F_P \rangle$ *and* $\langle Q, \longrightarrow_Q, F_Q \rangle$ *be two Logic LTS. A relation* $\mathcal{R} \subseteq P \times Q$ *is an* alternative ready simulation relation*, if the following three conditions hold, for any* $\langle p, q \rangle \in \mathcal{R}$ *and* $a \in \mathcal{A}$*:*

(RSi) $p \overset{\epsilon}{\Longrightarrow}| p' \implies \exists q'.\, q \overset{\epsilon}{\Longrightarrow}| q'$ *and* $\langle p', q' \rangle \in \mathcal{R}$;
(RSiii) $p \overset{a}{\Longrightarrow}| p'$ *and* p, q *stable* $\implies \exists q'.\, q \overset{a}{\Longrightarrow}| q'$ *and* $\langle p', q' \rangle \in \mathcal{R}$;
(RSiv) $p \notin F_P$ *and* p, q *stable* $\implies \mathcal{I}(p) = \mathcal{I}(q)$.

We write $p \sqsubseteq_{alt} q$*, if there exists an alternative ready simulation relation* $\mathcal{R}$ *with* $\langle p, q \rangle \in \mathcal{R}$.

It can easily be checked that $\sqsubseteq_{\mathrm{alt}}$ is the largest alternative ready simulation relation. The following proposition establishes that it is indeed an alternative definition of $\sqsubseteq_{\mathrm{RS}}$:

Proposition 13 (Coincidence) $\sqsubseteq_{RS} = \sqsubseteq_{alt}$.

PROOF. For proving inclusion "$\subseteq$", let $p \sqsubseteq_{\mathrm{RS}} q$. It is sufficient to establish that $\{\langle p, q \rangle\} \cup \underset{\sim}{\sqsubset}_{\mathrm{RS}}$ is an alternative ready simulation relation. First consider the pair $\langle p, q \rangle$:

(RSi) Straightforward.

(RSiii) Let $p \overset{a}{\Longrightarrow}| \, p'$ and p, q be stable, i.e., $p \notin F_P$. Hence, $p \overset{\epsilon}{\Longrightarrow}| \, p$ and, thus, $q \overset{\epsilon}{\Longrightarrow}| \, q$ as well as $p \underset{\sim}{\sqsubset}_{\mathrm{RS}} q$ by the root condition of $\sqsubseteq_{\mathrm{RS}}$. Now, Cond. (RSiii) immediately follows from Cond. (RS3).

(RSiv) Let $p \notin F_P$ and p, q be stable. We obtain $p \underset{\sim}{\sqsubset}_{\mathrm{RS}} q$ as above and are now done with Cond. (RS4).

If $\overline{p} \underset{\sim}{\sqsubset}_{\mathrm{RS}} \overline{q}$, then Conds. (RSiii) and (RSiv) are obviously satisfied due to the assumed Conds. (RS3) and (RS4). Moreover, Cond. (RSi) degrades to (RS2), since $\overline{p}$ and $\overline{q}$ are stable.

For establishing the reverse inclusion "$\supseteq$", we prove that $\sqsubseteq_{\mathrm{alt}}$, when restricted to stable processes, is a stable ready simulation relation. Let $p \sqsubseteq_{\mathrm{alt}} q$ for stable p, q. Conds. (RS1), (RS3) and (RS4) are straightforward. To prove Cond. (RS2), assume $p \notin F_P$ and p stable, i.e., $p \overset{\epsilon}{\Longrightarrow}| \, p$. By Cond. (RSi) we get $q \overset{\epsilon}{\Longrightarrow}| \, q$, i.e., $q \notin F_Q$. It remains for us to establish the root condition of $\sqsubseteq_{\mathrm{RS}}$, for which we let $p \sqsubseteq_{\mathrm{alt}} q$ and $p \overset{\epsilon}{\Longrightarrow}| \, p'$ for some stable p'. Then, by Cond. (RSi), there exists some stable q' with $q \overset{\epsilon}{\Longrightarrow}| \, q'$ and $p' \sqsubseteq_{\mathrm{alt}} q'$, i.e., $p' \underset{\sim}{\sqsubset}_{\mathrm{RS}} q'$. □

The main concern when studying ready simulation is with stable processes, which is more explicit in our definition rather than van Glabbeek's one. For this conceptional reason we will use $\sqsubseteq_{\mathrm{RS}}$ as defined in Def. 11. In addition, we note that the proofs of our statements appear to be equally concise when employing either definition of ready simulation; this is because the nature of ready simulation requires a separate treatment of instable processes in any case. We can now state and prove the desired compositionality result:

Theorem 14 (Compositionality)

(1) Let $p \underset{\sim}{\sqsubset}_{RS} q$, r be stable and $A \subseteq \mathcal{A}$. Then, (a) $p \parallel_A r \underset{\sim}{\sqsubset}_{RS} q \parallel_A r$ as well as (b) $p \wedge r \underset{\sim}{\sqsubset}_{RS} q \wedge r$.
(2) Let $p \sqsubseteq_{RS} q$, r be an arbitrary process and $A \subseteq \mathcal{A}$. Then, (a) $p \parallel_A r \sqsubseteq_{RS} q \parallel_A r$ and (b) $p \wedge r \sqsubseteq_{RS} q \wedge r$.

Proving this theorem requires us to reason about the consistency of conjunctively composed processes. To do so, it is convenient to employ the proof tool of $\wedge$-witness. The following $\wedge$-witness turns out to be adequate for our purpose:

Lemma 15 *The set $W =_{df} \{q \wedge r \,|\, \exists p', q', r'.\ p' \wedge r' \notin F,\ p' \underset{\sim}{\sqsubset}_{RS} q',\ q \overset{\epsilon}{\Longrightarrow}| \, q'$ and $r \overset{\epsilon}{\Longrightarrow}| \, r'\}$ is a $\wedge$-witness.*

PROOF. *(Lemma 15)* It will be advantageous to read W as the union of two sets W_1 and W_2, where W_1 collects the cases with $p = p'$, $q = q'$ and $r = r'$:

$$
\begin{aligned}
W_1 =_{\mathrm{df}} & \{ q \wedge r \,|\, \exists p.\ p \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q,\ r \text{ stable and } p \wedge r \notin F \}; \\
W_2 =_{\mathrm{df}} & \{ q \wedge r \,|\, \exists p', r'.\ p' \wedge r' \notin F,\ p' \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q',\ q \overset{\epsilon 1}{\Longrightarrow}\!|\, q',\ r \overset{\epsilon 2}{\Longrightarrow}\!|\, r', \text{ with} \\
& \qquad \{\epsilon 1, \epsilon 2\} = \{\epsilon, \tau\} \}\,.
\end{aligned}
$$

We need to check Conds. (AW1)–(AW4) of $\wedge$-witness.

(AW1) If $q \wedge r \in W_1$, then $p \wedge r \notin F$ implies $p \notin F$ and $r \notin F$. Moreover, $q \notin F$ by $p \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q$ and (RS2).

If $q \wedge r \in W_2$, then $q, r \notin F$ since $q \overset{\epsilon 1}{\Longrightarrow}\!|$ and $r \overset{\epsilon 2}{\Longrightarrow}\!|$.

(AW2) Let $q \wedge r \in W_1$, $q \overset{\tau}{\not\longrightarrow}$ and $r \overset{\tau}{\not\longrightarrow}$. Since $p \notin F$ (see above) and $p \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q$, we have $\mathcal{I}(p) = \mathcal{I}(q)$ by (RS4). In addition, $p \wedge r \notin F$, r stable and p stable by Def. 11, and thus $\mathcal{I}(p) = \mathcal{I}(r)$ by (C2). Hence, $\mathcal{I}(q) = \mathcal{I}(r)$.

If $q \wedge r \in W_2$, then $q \overset{\tau}{\longrightarrow}$ or $r \overset{\tau}{\longrightarrow}$ and we are done.

(AW3) If $q \wedge r \in W_1$ and $q \wedge r \overset{\alpha}{\longrightarrow}$, then we consider the following cases:

- $\underline{\alpha = \tau}$: This is case is impossible since q stable by $p \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q$, and r stable by assumption.
- $\underline{\alpha = a}$: Hence $q \overset{a}{\longrightarrow}$ and, with $p \notin F$ and (RS4), we get $p \overset{a}{\longrightarrow}$. Since $r \overset{a}{\longrightarrow}$ we have $p \wedge r \overset{a}{\longrightarrow}$. Because of $p \wedge r \notin F$ we obtain $p \wedge r \overset{a}{\longrightarrow}_{\mathrm{F}} \overline{p} \wedge \overline{r}$ for some $\overline{p} \wedge \overline{r} \notin F$ by (LTS1), and the latter process can stabilise by (LTS2), i.e., $\exists p', r'.\, p \wedge r \overset{a}{\longrightarrow}_{\mathrm{F}} \overline{p} \wedge \overline{r} \overset{\epsilon}{\Longrightarrow}\!|\, p' \wedge r'$. This implies $p \overset{a}{\Longrightarrow}\!|\, p'$, $q \overset{a}{\Longrightarrow}\!|\, q'$ and $p' \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q'$ for some q' by (RS3). We detail transition $q \overset{a}{\Longrightarrow}\!|\, q'$ by naming the first intermediate state $\overline{q}$, i.e., $q \overset{a}{\longrightarrow}_{\mathrm{F}} \overline{q} \overset{\epsilon}{\Longrightarrow}\!|\, q'$. Similarly, $r \overset{a}{\longrightarrow}_{\mathrm{F}} \overline{r} \overset{\epsilon}{\Longrightarrow}\!|\, r'$. If $\overline{q} \overset{\tau}{\Longrightarrow}\!|\, q'$ or $\overline{r} \overset{\tau}{\Longrightarrow}\!|\, r'$, then $q \wedge r \overset{a}{\longrightarrow} \overline{q} \wedge \overline{r} \in W_2$. Otherwise, $\overline{q} = q'$, $\overline{r} = r'$ and $q \wedge r \overset{a}{\longrightarrow} q' \wedge r' \in W_1$ due to p'; note $p' \wedge r' \notin F$ by the definition of $\overset{a}{\Longrightarrow}\!|$.

If $q \wedge r \in W_2$, then we only have to consider $q \wedge r \overset{\tau}{\longrightarrow}$, since $q \overset{\tau}{\longrightarrow}$ or $r \overset{\tau}{\longrightarrow}$. We only treat the first of these cases, i.e., $q \overset{\tau}{\longrightarrow}$, since the second is analogous. In this first case, $q \overset{\tau}{\longrightarrow}_{\mathrm{F}} \overline{q} \overset{\epsilon}{\Longrightarrow}\!|\, q'$ and $q \wedge r \overset{\tau}{\longrightarrow} \overline{q} \wedge r$ for some suitable $\overline{q}$. If $\overline{q} \overset{\tau}{\Longrightarrow}\!|\, q'$ or $r \overset{\tau}{\Longrightarrow}\!|\, r'$, then $\overline{q} \wedge r \in W_2$; otherwise, $\overline{q} = q'$, $r = r'$ and r stable, and $q' \wedge r' \in W_1$ due to p'.

(AW4) Let $q \wedge r \in W_1$ due to p. By the definition of W_1, q is stable (by $p \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q$) and r is stable, i.e., $q \wedge r$ is stable, too.

Let $q \wedge r \in W_2$. From the assumptions, we derive a computation $q \wedge r \overset{\tau}{\Longrightarrow} q' \wedge r'$ with $q' \wedge r'$ stable, i.e., q', r' stable. Consider a process $\overline{q} \wedge \overline{r}$ on this computation so that $q \wedge r \overset{\tau}{\Longrightarrow} \overline{q} \wedge \overline{r} \overset{\epsilon}{\Longrightarrow} q' \wedge r'$. If $\overline{q} \overset{\tau}{\Longrightarrow} q'$ or $\overline{r} \overset{\tau}{\Longrightarrow} r'$, then $\overline{q} \wedge \overline{r} \in W_2$; otherwise $\overline{q} = q'$, $\overline{r} = r'$ and $q' \wedge r' \in W_1$ due to p'. $\square$

Using this lemma, we can now prove Thm. 14:

PROOF. *(Thm. 14)* The proof of Parts (1a) and Part (2a) is straightforward. In particular, Part (1a) proceeds in the usual fashion, i.e., by verifying that

$$\mathcal{R}_{\|} =_{\mathrm{df}} \{\langle p \parallel_A r, q \parallel_A r\rangle \mid r \text{ stable and } p \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q\}$$

is a stable ready simulation relation.

The proof of Part (1b) is quite challenging since we need to take care of inconsistencies that may arise when composing processes conjunctively. In analogy to the above we prove that

$$\mathcal{R}_{\wedge} =_{\mathrm{df}} \{\langle p \wedge r, q \wedge r\rangle \mid r \text{ stable and } p \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q\}$$

is a stable ready simulation relation. Let $\langle p \wedge r, q \wedge r\rangle \in \mathcal{R}_{\wedge}$, i.e., r stable and $p \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q$.

(RS1) This property is straightforward since the conjunction of two processes is stable exactly when both processes are stable.

(RS2) This property follows immediately from the fact that $p \wedge r \notin F$ implies that $q \wedge r$ is contained in the $\wedge$-witness W of Lemma 15.

(RS3) Assume $p \wedge r \overset{a}{\Longrightarrow}\!| \, p' \wedge r'$. Hence, $p \overset{a}{\Longrightarrow}\!| \, p'$ and $r \overset{a}{\longrightarrow}_{\mathrm{F}} r'' \overset{\epsilon}{\Longrightarrow}\!| \, r'$, and by (RS3) also $q \overset{a}{\longrightarrow}_{\mathrm{F}} q'' \overset{\epsilon}{\Longrightarrow}\!| \, q'$ with $p' \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q'$. We combine the latter two computations to obtain the computation $q \wedge r \overset{a}{\longrightarrow} q'' \wedge r'' \overset{\epsilon}{\Longrightarrow} q' \wedge r'$; further note that $q' \wedge r'$ is stable.

Consider $\overline{q} \wedge \overline{r}$ with $q'' \wedge r'' \overset{\epsilon}{\Longrightarrow} \overline{q} \wedge \overline{r} \overset{\epsilon}{\Longrightarrow} q' \wedge r'$. Then, recall $p' \wedge r' \notin F$ (due to $p \wedge r \overset{a}{\Longrightarrow}\!| \, p' \wedge r'$) and $p' \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q'$. Thus, $\overline{q} \wedge \overline{r} \in W$.

Thus, no process along $q'' \wedge r'' \overset{\epsilon}{\Longrightarrow} q' \wedge r'$ is in F by Prop. 4. Since $p \wedge r \notin F$ due to $p \wedge r \overset{a}{\Longrightarrow}\!| \, p' \wedge r'$ and r stable, we have $q \wedge r \in W$, i.e., $q \wedge r \notin F$ by Prop. 4. We conclude $q \wedge r \overset{a}{\longrightarrow}_{\mathrm{F}} q'' \wedge r'' \overset{\epsilon}{\Longrightarrow}_{\mathrm{F}} q' \wedge r'$ and $q' \wedge r'$ stable, whence $q \wedge r \overset{a}{\Longrightarrow}\!| \, q' \wedge r'$. Since r' is stable, we also have $\langle p' \wedge r', q' \wedge r'\rangle \in \mathcal{R}_{\wedge}$.

(RS4) Assume $p \wedge r \notin F$. Hence, we have $p \notin F$ and $\mathcal{I}(p) = \mathcal{I}(r)$. Further, $\mathcal{I}(p) = \mathcal{I}(q)$ by (RS4) for $p \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q$, and thus $\mathcal{I}(q) = \mathcal{I}(r)$. Hence, $\mathcal{I}(p \wedge r) = \mathcal{I}(r) = \mathcal{I}(q \wedge r)$, as desired.

To prove Part (2b), let $p \wedge r \overset{\epsilon}{\Longrightarrow}\!| \, p' \wedge r'$. Hence, $p \overset{\epsilon}{\Longrightarrow}\!| \, p'$ and $r \overset{\epsilon}{\Longrightarrow}\!| \, r'$. Due to $p \sqsubseteq_{\mathrm{RS}} q$, the former implies the existence of some q' such that $q \overset{\epsilon}{\Longrightarrow}\!| \, q'$ and $p' \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q'$. Therefore, $p' \wedge r' \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q' \wedge r'$ by Part (1b). Further we apply Lemma 15 and Prop. 4 to obtain $q' \wedge r' \notin F$, as $q' \wedge r' \in W_1$ due to p'. Thus, Lemma 5(2) proves $q \wedge r \overset{\epsilon}{\Longrightarrow}\!| \, q' \wedge r'$. $\square$

Before presenting our full-abstraction result, we first highlight some logic properties of ready simulation.

Theorem 16 ($\wedge$ is And) *(1) $r \mathrel{\underset{\sim}{\sqsubset}}_{RS} p \wedge q$ if and only if $r \mathrel{\underset{\sim}{\sqsubset}}_{RS} p$ and $r \mathrel{\underset{\sim}{\sqsubset}}_{RS} q$; (2) $r \sqsubseteq_{RS} p \wedge q$ if and only if $r \sqsubseteq_{RS} p$ and $r \sqsubseteq_{RS} q$.*

As for the compositionality proof of ready simulation wrt. conjunction, the proof of this theorem uses the concept of $\wedge$-witness for reasoning about inconsistencies:

Lemma 17 *The set $W' =_{df} \{p \wedge q \,|\, \exists r', p', q'.\, r' \notin F,\, p \overset{\epsilon}{\Longrightarrow}| \, p',\, q \overset{\epsilon}{\Longrightarrow}| \, q',\, r' \mathrel{\underset{\sim}{\sqsubset}}_{RS} p'$ and $r' \mathrel{\underset{\sim}{\sqsubset}}_{RS} q'\}$ is a $\wedge$-witness.*

PROOF. It will be advantageous to read W' as the union of two sets W'_1 and W'_2:

$$\begin{aligned} W'_1 =_{\mathrm{df}} & \{p \wedge q \,|\, \exists r.\, r \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} p,\ r \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q \text{ and } r \notin F\} \\ W'_2 =_{\mathrm{df}} & \{p \wedge q \,|\, \exists r', p', q'.\, r' \notin F,\ p \overset{\epsilon 1}{\Longrightarrow}| \, p' \text{ and } q \overset{\epsilon 2}{\Longrightarrow}| \, q' \\ & \qquad \text{with } \{\epsilon 1, \epsilon 2\} = \{\epsilon, \tau\},\ r' \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} p' \text{ and } r' \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q'\}\,. \end{aligned}$$

We check the four conditions of $\wedge$-witness for W':

(AW1) If $p \wedge q \in W'_1$ due to $r \notin F$, then $p, q \notin F$ by (RS2). If $p \wedge q \in W'_2$, then we are done by the definition of $\overset{\epsilon 1}{\Longrightarrow}|$ and $\overset{\epsilon 2}{\Longrightarrow}|$.

(AW2) If $p \wedge q \in W'_1$ due to $r \notin F$, then $\mathcal{I}(p) = \mathcal{I}(r) = \mathcal{I}(q)$. If $p \wedge q \in W'_2$, we are done since $p \xrightarrow{\tau}$ or $q \xrightarrow{\tau}$ by $\{\epsilon 1, \epsilon 2\} = \{\epsilon, \tau\}$.

(AW3) If $p \wedge q \in W'_1$, then $\alpha \neq \tau$, and $p \wedge q \xrightarrow{\alpha}$ implies $r \xrightarrow{\alpha}$ by (AW2) above. Since $r \notin F$, we have $r \overset{\alpha}{\Longrightarrow}| \, r'$ for some $r' \notin F$. Hence, $\exists p', q'.\, p \xrightarrow{\alpha}_{\mathrm{F}} \overline{p} \overset{\epsilon}{\Longrightarrow} p'$, $q \xrightarrow{\alpha}_{\mathrm{F}} \overline{q} \overset{\epsilon}{\Longrightarrow} q'$, $r' \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} p'$ and $r' \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q'$. Thus, $p \wedge q \xrightarrow{\alpha} \overline{p} \wedge \overline{q}$. If $\overline{p} \neq p'$ or $\overline{q} \neq q'$, then $\overline{p} \wedge \overline{q} \in W'_2$. If $\overline{p} = p'$ and $\overline{q} = q'$, then $\overline{p} \wedge \overline{q} \in W'_1$ due to r'.
If $p \wedge q \in W'_2$, then $\alpha = \tau$ since $p \xrightarrow{\tau}$ or $q \xrightarrow{\tau}$. Consider, w.l.o.g., $p \wedge q \xrightarrow{\tau} \overline{p} \wedge q$ for some $\overline{p}$. If $\overline{p} \neq p'$ or $q \neq q'$, we have $\overline{p} \wedge q \in W'_2$. Otherwise, $\overline{p} \wedge q = p' \wedge q' \in W'_1$ due to r'.

(AW4) If $p \wedge q \in W'_1$, then $p \wedge q$ is stable since both p and q are stable according to (RS1). If $p \wedge q \in W'_2$, then $p \wedge q$ can also stabilise in W', cf. (AW3). □

The proof of Thm. 16 is now quite straightforward:

PROOF. *(Thm. 16)* We start with Part (1), direction "$\Longrightarrow$", and show that

$$\mathcal{R} =_{\mathrm{df}} \{\langle r, p\rangle \mid \exists q.\, r \underset{\sim}{\sqsubset}_{\mathrm{RS}} p \wedge q\}$$

is a stable ready simulation relation. Let $\langle r, p\rangle \in \mathcal{R}$.

(RS1) Since r and $p \wedge q$ are stable due to $r \underset{\sim}{\sqsubset}_{\mathrm{RS}} p \wedge q$, we have that p is stable, too.

(RS2) $p \in F$ implies $p \wedge q \in F$, which in turn implies $r \in F$ due to $r \underset{\sim}{\sqsubset}_{\mathrm{RS}} p \wedge q$.

(RS3) $r \overset{a}{\Longrightarrow}\!| \, r'$ implies, since $r \underset{\sim}{\sqsubset}_{\mathrm{RS}} p \wedge q$, the existence of p' and q' such that $p \wedge q \overset{a}{\Longrightarrow}\!| \, p' \wedge q'$ and $r' \underset{\sim}{\sqsubset}_{\mathrm{RS}} p' \wedge q'$. Hence, $p \overset{a}{\Longrightarrow}\!| \, p'$ and $\langle r', p'\rangle \in \mathcal{R}$.

(RS4) By $r \underset{\sim}{\sqsubset}_{\mathrm{RS}} p \wedge q$, we have that $r \notin F$ implies $\mathcal{I}(r) = \mathcal{I}(p \wedge q)$. Moreover, since $p \wedge q \notin F$ (cf. (RS2)) and $p \wedge q$ stable, we get $\mathcal{I}(p) = \mathcal{I}(q)$ by (C2). Hence, $\mathcal{I}(r) = \mathcal{I}(p \wedge q) = \mathcal{I}(p)$.

Hence, $r \underset{\sim}{\sqsubset}_{\mathrm{RS}} p$. Analogously, we can establish $r \underset{\sim}{\sqsubset}_{\mathrm{RS}} q$.

Now, let $r \sqsubseteq_{\mathrm{RS}} p \wedge q$. Then, $r \overset{\epsilon}{\Longrightarrow}\!| \, r'$ implies $p \wedge q \overset{\epsilon}{\Longrightarrow}\!| \, p' \wedge q'$ for some p', q' with $r' \underset{\sim}{\sqsubset}_{\mathrm{RS}} p' \wedge q'$. Thus, $p \overset{\epsilon}{\Longrightarrow}\!| \, p'$ and, by the above, $r' \underset{\sim}{\sqsubset}_{\mathrm{RS}} p'$. This proves $r \sqsubseteq_{\mathrm{RS}} p$. Again, we can prove $r \sqsubseteq_{\mathrm{RS}} q$ analogously, which finishes the proof of Part (2), direction "$\Longrightarrow$".

To show direction "$\Longleftarrow$" of Part (1), it is sufficient to prove that

$$\mathcal{R} =_{\mathrm{df}} \{\langle r, p \wedge q\rangle \mid r \underset{\sim}{\sqsubset}_{\mathrm{RS}} p \text{ and } r \underset{\sim}{\sqsubset}_{\mathrm{RS}} q\}$$

is a stable ready simulation relation. Let $\langle r, p \wedge q\rangle \in \mathcal{R}$.

(RS1) This is trivial since p, q, r are stable due to $r \underset{\sim}{\sqsubset}_{\mathrm{RS}} p$ and $r \underset{\sim}{\sqsubset}_{\mathrm{RS}} q$.

(RS2) If $r \notin F$, then $p \wedge q \in W'$ of Lemma 17, i.e., $p \wedge q \notin F$ by Prop. 4.

(RS3) If $r \overset{a}{\Longrightarrow}\!| \, r'$, then $p \overset{a}{\Longrightarrow}\!| \, p'$ and $q \overset{a}{\Longrightarrow}\!| \, q'$, as well as $r' \underset{\sim}{\sqsubset}_{\mathrm{RS}} p'$ and $r' \underset{\sim}{\sqsubset}_{\mathrm{RS}} q'$ by (RS3) for $r \underset{\sim}{\sqsubset}_{\mathrm{RS}} p$ and $r \underset{\sim}{\sqsubset}_{\mathrm{RS}} q$. Moreover, $p' \wedge q' \in W'$ since $r' \notin F$ and thus $p' \wedge q' \notin F$ by Prop. 4. Now we combine the latter two computations and consider intermediate states $\overline{p} \wedge \overline{q}$ such that $p \wedge q \overset{a}{\Longrightarrow} \overline{p} \wedge \overline{q} \overset{\tau}{\Longrightarrow} p' \wedge q'$, i.e., $p \overset{a}{\Longrightarrow}_{\mathrm{F}} \overline{p} \overset{\epsilon 1}{\Longrightarrow}\!| \, p'$ and $q \overset{a}{\Longrightarrow}_{\mathrm{F}} \overline{q} \overset{\epsilon 2}{\Longrightarrow}\!| \, q'$ with $\{\epsilon 1, \epsilon 2\} = \{\epsilon, \tau\}$; this implies $\overline{p} \wedge \overline{q} \in W'_2 \subseteq W'$. Hence, $\overline{p} \wedge \overline{q} \notin F$ by Prop. 4. Moreover, because of $p \wedge q \notin F$ by (RS2), we have $p \wedge q \overset{a}{\Longrightarrow}\!| \, p' \wedge q'$.

(RS4) If $r \notin F$, then $\mathcal{I}(p) = \mathcal{I}(r) = \mathcal{I}(q)$ lets us conclude $\mathcal{I}(r) = \mathcal{I}(p \wedge q)$.

Finally, let $r \sqsubseteq_{\mathrm{RS}} p$ and $r \sqsubseteq_{\mathrm{RS}} q$. For $r \overset{\epsilon}{\Longrightarrow}\!| \, r'$, take p' and q' with $p \overset{\epsilon}{\Longrightarrow}\!| \, p'$, $q \overset{\epsilon}{\Longrightarrow}\!| \, q'$, $r' \underset{\sim}{\sqsubset}_{\mathrm{RS}} p'$ and $r' \underset{\sim}{\sqsubset}_{\mathrm{RS}} q'$, and observe $r' \notin F$. By the above, $r' \underset{\sim}{\sqsubset}_{\mathrm{RS}} p' \wedge q'$ and $p \wedge q \overset{\epsilon}{\Longrightarrow} p' \wedge q'$. Referring to Lemma 5(2), and noting $p' \wedge q' \notin F$ by (RS2), we obtain $p \wedge q \overset{\epsilon}{\Longrightarrow}\!| \, p' \wedge q'$, which proves direction "$\Longleftarrow$" of Part (2). $\square$

Conjunction also satisfies further standard logic properties:

Proposition 18 (Logic properties of ready simulation)

(1) (a) $p \wedge ff =_{RS} ff$, and (b) $p \wedge ff \approx_{RS} ff$ if p stable;
(2) (a) $p \wedge q \sqsubseteq_{RS} p$, and (b) $p \wedge q \underset{\sim}{\sqsubset}_{RS} p$ if p, q stable;
(3) $p \wedge p =_{RS} p$;
(4) $p \wedge q =_{RS} p$ if and only if $p \sqsubseteq_{RS} q$.

PROOF.

(1) The validity of part (b), for stable p, is a consequence of the fact that $\{\langle ff, p \wedge ff\rangle \mid p \text{ stable}\}$ and $\{\langle p \wedge ff, ff\rangle \mid p \text{ stable}\}$ are stable ready simulation relations. Part (a) holds trivially since neither $p \wedge ff \overset{\epsilon}{\Longrightarrow}\!|$ nor $ff \overset{\epsilon}{\Longrightarrow}\!|$.

(2) To prove Part (b) for stable p, q, it is sufficient to verify that the relation $\mathcal{R} =_{\text{df}} \{\langle p \wedge q, p\rangle \mid p, q \text{ stable}\}$ is a stable ready simulation relation:

(RS1) Trivial.

(RS2) $p \in F$ implies $p \wedge q \in F$.

(RS3) $p \wedge q \overset{a}{\Longrightarrow}\!| \, p' \wedge q'$ implies $p \overset{a}{\Longrightarrow}\!| \, p'$ and $\langle p' \wedge q', p'\rangle \in \mathcal{R}$.

(RS4) $p \wedge q \notin F$ and $p \wedge q$ stable implies $\mathcal{I}(p \wedge q) = \mathcal{I}(p)$.

For proving Part (a) for arbitrary p, q, let $p \wedge q \overset{\epsilon}{\Longrightarrow}\!| \, p' \wedge q'$. Then, $p \overset{\epsilon}{\Longrightarrow}\!| \, p'$ and, by the above, $p' \wedge q' \underset{\sim}{\sqsubset}_{\text{RS}} p'$.

(3) The inclusion "$\sqsubseteq_{\text{RS}}$" is a consequence of Part (2). The inclusion "$\sqsupseteq_{\text{RS}}$" follows by Thm. 16(2).

(4) Part "$\sqsubseteq_{\text{RS}}$" of the "if" direction is the statement of Part (2). Part "$\sqsupseteq_{\text{RS}}$" of the "if" direction is a consequence of the compositionality and idempotence of $\wedge$ (Thm. 14(2) and Part (3), respectively). The "only if" direction follows directly from Part (2) and commutativity. □

3.3 Full-abstraction result

To prove our main result, we encode the full behaviour of a stable process p into a single ready tree. The idea is to unfold p to a tree and to eliminate any nondeterminism by placing fresh actions into the tree. As an illustration of this, consider a part of a process p — or rather its unfolding — as shown on the left-hand side of Fig. 5.

A ready tree for state 1 would represent the a-transition, the b-transition and one of the two c-transitions; in terms of Def. 19 below, there are two relevant selection sets. In order to reflect both possibilities of c-transitions in a deterministic way, the characteristic ready tree of p chooses between them with fresh actions x and y, as shown in the centre of Fig. 5. Of course, due

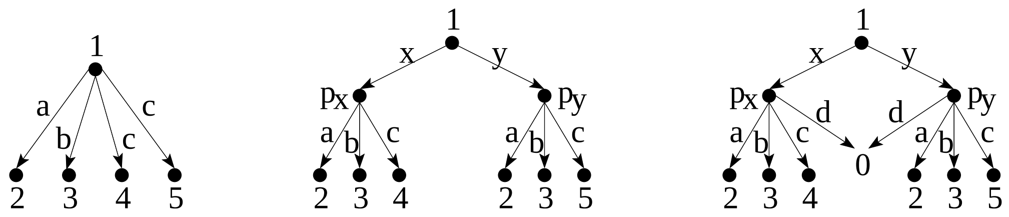

Fig. 5. Basic intuition behind conjunctive composition.

to the presence of x and y, this tree is not a ready tree of p, but it is a ready tree of a suitable parallel composition. For this composition, we need a context K which we take to be the characteristic ready tree augmented by some additional transitions. These depend on the sort of the process q we want to compare with p: if q has sort $\{a, b, c, d\}$, the respective part of K looks as shown on the right-hand side of Fig. 5, where 0 denotes the deadlock process that has no initial actions.

We now define the *characteristic ready tree* and *context* formally:

Definition 19 (Characteristic ready tree & context) *Let p be a process with Logic LTS $\langle P, \longrightarrow, F\rangle$ having sort $\mathcal{A}_P$, let q be a process with sort $\mathcal{A}_Q$, and let $p \overset{\epsilon}{\Longrightarrow\!|}\, p_0$.*

(1) The characteristic ready tree P_0 *of p with respect to p_0 and q is a Logic LTS* [7] *whose states are paths $\pi \in P \times (\mathcal{A}_P \times P)^*$ of p originating in p_0, as well as such paths concatenated with* selection sets D *which are subsets of $\mathcal{A}_P \times P$. Formally, the state set P_0 and transition relation $\longrightarrow_{P_0}$ are inductively defined as follows, where $last(\pi)$ denotes the last process on path π and the $x_D \notin \mathcal{A}_P \cup \mathcal{A}_Q$ are fresh actions with respect to p and q:*
- *$p_0 \in P_0$;*
- *$\pi \xrightarrow{x_D}_{P_0} \pi D$ and $\pi D \in P_0$, if $\pi \in P_0$, $\forall \langle a, p\rangle \in D.\ last(\pi) \overset{a}{\Longrightarrow\!|}\, p$ in P and $\forall a \in \mathcal{I}(last(\pi))\, \exists_1 \langle a, p\rangle \in D$;*
- *$\pi D \xrightarrow{a}_{P_0} \pi a p$ and $\pi a p \in P_0$, if $\pi D \in P_0$ and $\langle a, p\rangle \in D$.*

We will write $\langle p_0\rangle$ instead of p_0 whenever we wish to highlight that not the process p_0 is meant, but the path consisting only of p_0.

(2) The characteristic context K *of p with respect to p_0 and q is defined as the Logic LTS P_0 augmented with the fresh process 0 and transitions*
- *$\pi D \xrightarrow{a}_K 0$, if $\pi D \in P_0$, $a \in \mathcal{A}_Q$ and $\nexists p. \langle a, p\rangle \in D$.*

Also K does not have any inconsistent states.

Next, we prove that P_0, as defined in Def. 19, is indeed a ready tree. Note that in the parallel composition, the additional transitions of the context — i.e., the d-transitions in Fig. 5 — are eliminated; they will be needed in the proof of full abstraction (cf. Thm. 21).

[7] As an aside, note that only the sort of process q is needed here, not q itself. Further recall that a ready tree does not have any inconsistent states.

Proposition 20 *Let P_0 be the characteristic ready tree of process p wrt. some p_0 and q, and let K be the respective characteristic context of p. Then, P_0 is a ready tree of $p \parallel_A \langle p_0 \rangle$, where $A =_{df} \mathcal{A}_P \cup \mathcal{A}_Q$ and $\langle p_0 \rangle$ is the root of K.*

PROOF. P_0 is an observation tree by construction, since it is a deterministic tree and since all its vertices are stable processes. We define a mapping h_0 from the vertices in P_0 to processes in $P \parallel_A K$ by $h_0(\pi) =_{\text{df}} \text{last}(\pi) \parallel_A \pi$ and $h_0(\pi D) =_{\text{df}} \text{last}(\pi) \parallel_A \pi D$, and verify Conds. (RT1)–(RT4) of Def. 8:

(RT1) This is trivial since last(π), π and πD are all stable and not in F.

(RT2) We have $p \parallel_A \langle p_0 \rangle \overset{\epsilon}{\Longrightarrow}\!| \, p_0 \parallel_A \langle p_0 \rangle$ by construction.

(RT3) If $\pi \xrightarrow{x_D}_{P_0} \pi D$, then $\pi \xrightarrow{x_D}_K \pi D$ by construction of K. Since x_D is a "fresh" action, $h_0(\pi) = \text{last}(\pi) \parallel_A \pi \xrightarrow{x_D}_{\text{F}} \text{last}(\pi) \parallel_A \pi D = h_0(\pi D)$. If $\pi D \xrightarrow{a}_{P_0} \pi a p$, then $\text{last}(\pi) \overset{a}{\Longrightarrow}\!| \, p$ and $\pi D \xrightarrow{a}_K \pi a p$ by the construction of K. As $a \in A$, we have $h_0(\pi D) = \text{last}(\pi) \parallel_A \pi D \overset{a}{\Longrightarrow}\!| \, p \parallel_A \pi a p = h_0(\pi a p)$.

(RT4) Observe that the ready set of πD in K is the ready set $\mathcal{I}_P(\text{last}(\pi))$ of the last process of path π in P plus all actions in $\mathcal{A}_Q$, whereas the same state in P_0 has only ready set $\mathcal{I}_P(\text{last}(\pi))$. By the operational rules for parallel composition we obtain:
- $\mathcal{I}_{P \parallel_A K}(\text{last}(\pi) \parallel_A \pi) = (\mathcal{I}_P(\text{last}(\pi)) \cap \mathcal{I}_K(\pi) \cap A) \cup (\mathcal{I}_P(\text{last}(\pi)) \setminus A) \cup (\mathcal{I}_K(\pi) \setminus A) = \emptyset \cup \emptyset \cup \mathcal{I}_K(\pi) = \mathcal{I}_{P_0}(\pi)$.
- $\mathcal{I}_{P \parallel_A K}(\text{last}(\pi) \parallel_A \pi D) = (\mathcal{I}_P(\text{last}(\pi)) \cap \mathcal{I}_K(\pi D) \cap A) \cup (\mathcal{I}_P(\text{last}(\pi)) \setminus A) \cup (\mathcal{I}_K(\pi D) \setminus A) = (\mathcal{I}_P(\text{last}(\pi)) \cap (\mathcal{I}_P(\text{last}(\pi)) \cup \mathcal{A}_Q) \cap A) \cup \emptyset \cup \emptyset = \mathcal{I}_P(\text{last}(\pi)) = \mathcal{I}_{P_0}(\pi D)$; note that the last equality is due to the construction of P_0 from P. □

Together, characteristic ready trees and Prop. 20 are the key for proving our main result:

Theorem 21 (Full abstraction) *The largest precongruence within $\sqsubseteq_F$, with respect to parallel composition and conjunction, equals $\sqsubseteq_{RS}$.*

PROOF. Because of Thm. 14 and Thm. 10 [6], as well as the fact that ready simulation is contained in the ready-tree preorder $\subseteq_{\text{RT}}$ and thus in $\sqsubseteq_F$ [6], it is sufficient to prove that $\sqsubseteq_{\text{RS}}$ subsumes the largest precongruence $\subseteq^+_{\text{RT}}$ contained in $\subseteq_{\text{RT}}$. Consider processes p and q with Logic LTSs P and Q and sorts $\mathcal{A}_P$ and $\mathcal{A}_Q$. We let $\mathcal{A}_{PQ}$ stand for $\mathcal{A}_P \cup \mathcal{A}_Q$, and abbreviate $\parallel_{\mathcal{A}_{PQ}}$ by $\parallel$.

Now assume $p \subseteq^+_{\text{RT}} q$, and consider some p_0 such that $p \overset{\epsilon}{\Longrightarrow}\!| \, p_0$. Because of $p \subseteq^+_{\text{RT}} q$ and Prop. 20, we have $P_0 \in \text{RT}(q \parallel \langle p_0 \rangle)$ due to some mapping h; in particular, $q \notin F$. Here, P_0 is the characteristic ready tree of p with respect to p_0 and q. To prove our claim, it is sufficient to establish that

$$\mathcal{R}_0 =_{\mathrm{df}} \{\langle p', q' \rangle \mid \exists \pi.\, \mathrm{last}(\pi) = p' \text{ and } h(\pi) = q' \parallel \pi\}$$

is a stable ready simulation relation. Thus, let $\langle p', q' \rangle \in \mathcal{R}_0$ due to π.

(RS1) $h(\pi)$ is stable, whence q' is. Moreover, $\mathrm{last}(\pi)$ is stable by construction.

(RS2) $h(\pi) \notin F$ implies $q' \notin F$.

(RS3) Let $p' \stackrel{a}{\Longrightarrow}\!| \, p''$ and $\pi \xrightarrow{x_D} \pi D$ with $\langle a, p'' \rangle \in D$ for some p''; the latter transition always exists by construction. Then, $\pi D \xrightarrow{a} \pi a p''$. Moreover, $h(\pi D) = q' \parallel \pi D$, whence $q' \parallel \pi D \stackrel{a}{\Longrightarrow}\!| \, h(\pi a p'') = q'' \parallel \pi a p''$ for some q'' by (RT3), as well as $q' \stackrel{a}{\Longrightarrow}\!| \, q''$ and $\langle p'', q'' \rangle \in \mathcal{R}_0$ due to $\pi a p''$.

(RS4) We have $p' \notin F$ by construction. Choose some D with $\pi \xrightarrow{x_D} \pi D$ which always exists by construction, whence $h(\pi D) = q' \| \pi D$. Now, $\mathcal{I}(p') = \mathcal{I}(\pi D)$ in P_0 by construction of P_0. The latter equals $\mathcal{I}(q' \| \pi D)$ by (RT4), which in turn equals the set $\mathcal{I}(q')$ since $\mathcal{A}_Q \subseteq \mathcal{I}(\pi D) \subseteq \mathcal{A}_{PQ}$, for $\mathcal{I}(\pi D)$ in the characteristic context. Hence, $\mathcal{I}(p') = \mathcal{I}(q')$.

Observe that the additional transitions of the context ensure that $\mathcal{A}_Q \subseteq \mathcal{I}(\pi D)$. To see what goes wrong without these transitions, consider Fig. 5 with p_x corresponding to πD, and 1 to p'. Without the d-transitions we could have $\mathcal{I}(q' \parallel \pi D) = \{a, b, c\}$, but nevertheless $\mathcal{I}(q') = \{a, b, c, d\} \neq \mathcal{I}(p')$.

Thus, $\mathcal{R}_0$ is a stable ready simulation relation. Finally observe $h(p_0) = q_0 \parallel \langle p_0 \rangle$ for some q_0 such that $q \| \langle p_0 \rangle \stackrel{\epsilon}{\Longrightarrow}\!| \, q_0 \| \langle p_0 \rangle$ (by (RT2)); therefore, $q \stackrel{\epsilon}{\Longrightarrow}\!| \, q_0$ and $\langle p_0, q_0 \rangle \in \mathcal{R}_0$ due to $\langle p_0 \rangle$.

Summarising, we have shown that, for each p_0 with $p \stackrel{\epsilon}{\Longrightarrow}\!| \, p_0$, there exists some q_0 satisfying $q \stackrel{\epsilon}{\Longrightarrow}\!| \, q_0$ and $p_0 \underset{\sim}{\sqsubset}_{\mathrm{RS}} q_0$. Hence, $p \sqsubseteq_{\mathrm{RS}} q$. □

We wish to point out that there are several ways how to guarantee the existence of the fresh actions required in the construction of characteristic ready trees. One way is to assume an uncountable alphabet $\mathcal{A}$ and to restrict ourselves to those processes that are finitely branching with respect to $\bigcup\{ \stackrel{a}{\Longrightarrow}\!| \mid a \in \mathcal{A}\}$ and have a countable sort. Then, context K and the characteristic ready trees are also finitely branching and have countable sorts.

Alternatively, we may assume an infinite alphabet $\mathcal{A}$ and restrict ourselves to processes that have finite sort and are bounded branching, for some bound $c \in \mathbb{N}$. This is sufficient since a careful inspection of the full-abstraction proof reveals that actually only c fresh actions are needed for constructing context K and the required characteristic ready trees. Moreover, K and the characteristic ready trees have then only c-bounded branching as well as finite sorts.

Note that the need for fresh actions is standard in full-abstraction proofs (see, e.g., Milner [3, p. 153]) and is discussed extensively in the process-algebra literature.

4 Example

To testify to the utility of our framework for mixed operational and logic styles of reasoning and specification we conduct a small but realistic example. The example deals with the specification and refinement of a simple mode logic, such as used in flight control systems in avionics and in other embedded control applications. Note that the avionics term "mode" refers to a state, or a set of states, as used in formal methods and concurrency theory.

The brief for the mode logic is as follows: The mode logic shall consist of two modes, which can be thought of as on-/off-switches and whose interaction shall be governed by some scheduler. This is a standard architecture for mode logics. However, our particular mode logic is required to satisfy the constraint that not both modes can be on at any given time.

This informal requirements description directly lends itself to a heterogeneous specification of the mode logic in a mixed operational and logic style, using both parallel composition and conjunction. Before formally presenting this specification, we introduce some conventions to simplify notation in this section. We use $P \parallel Q$ to denote the parallel composition of P and Q with the joint alphabet of P and Q being the synchronisation alphabet. Here, the alphabet of a Logic LTS is simply the set of actions occurring as labels of the transitions of the LTS. The main purpose of this convention, other than relieving us from explicitly annotating each parallel operator by a synchronisation alphabet, is that $\parallel$ is commutative and associative provided the alphabet of $P \parallel Q$ is the union of the alphabets of P and Q; this condition will be satisfied for the parallel compositions in this section.

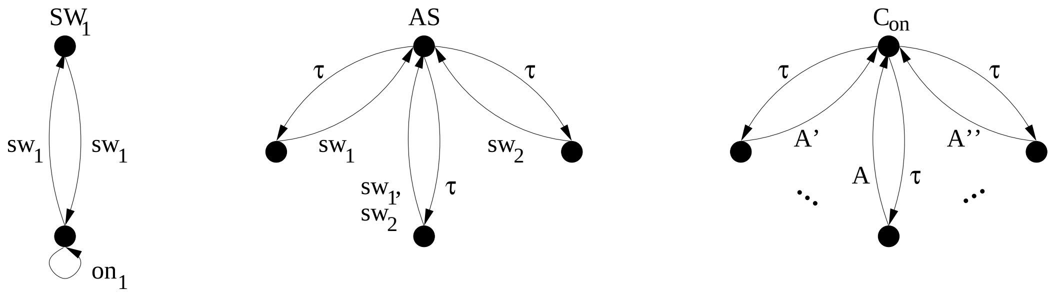

Fig. 6. Abstract scheduler $\mathtt{AS}$ and constraint $\mathtt{C_{on}}$.

We can now phrase the specification of our mode logic as

$$\mathtt{Spec} =_{\mathrm{df}} (\mathtt{SW}_1 \parallel \mathtt{SW}_2 \parallel \mathtt{AS}) \wedge \mathtt{C}_{\mathtt{on}} \,,$$

whose component Logic LTSs are sketched in Fig. 6, with $\mathtt{SW}_2$ being defined as $\mathtt{SW}_1$ but replacing all indices 1 with 2. The LTS $\mathtt{SW}_1$, and analogously $\mathtt{SW}_2$, specifies a simple switch with two states: the initial, upper off-state and the lower on-state. One can toggle between these states using the switching action $\mathtt{sw}_1$, with the $\mathtt{on}_1$ self-loop of the on-state indicating that the switch is on. Component $\mathtt{AS}$ is the most abstract scheduler for the two switches: it allows at least one switching action at any state. Here, the transition labelled "$\mathtt{sw}_1, \mathtt{sw}_2$" stands for two transitions, one labelled with $\mathtt{sw}_1$ and the other with $\mathtt{sw}_2$. Intuitively, $\mathtt{AS}$ can be understood as the temporal logic formula "*always*$(\mathtt{sw}_1 \vee \mathtt{sw}_2 \vee (\mathtt{sw}_1 \Box \mathtt{sw}_2))$", where operator $\Box$ denotes external choice (cf. Def. 26 below). Note that our scheduler cannot control the actions $\mathtt{on}_1$ and $\mathtt{on}_2$, which are just probes as suggested above.

Finally, Logic LTS $\mathtt{C}_{\mathtt{on}}$ formalises our desired constraint that only one mode can be in its on-state at any given time. This LTS has a τ-branch for each action set $A \subseteq \{\mathtt{on}_1, \mathtt{on}_2, \mathtt{sw}_1, \mathtt{sw}_2\}$ such that not both $\mathtt{on}_1 \in A$ and $\mathtt{on}_2 \in A$; for illustration purposes, Fig. 6 shows several such sets A, denoted A, A' and A''. Each τ-branch returns to its initial state via a bundle of transitions, one for each action $a \in A$. In Fig. 6, this bundle of transitions is simply depicted as a single transition labelled A. One may think of this LTS as being automatically generated from the temporal logic formula "*always*$\,\neg(\mathtt{on}_1 \wedge \mathtt{on}_2)$." Hence, our specification is in principle a conjunction of a parallel composition and a temporal logic formula, where the parallel composition contains a temporal logic formula as component, and is thus truly heterogeneous.

Our aim is to refine the above specification $\mathtt{Spec}$ to a purely operational system $\mathtt{Impl}$ such that the parallel composition is refined and constraint $\mathtt{C}_{\mathtt{on}}$ is satisfied, i.e., $\mathtt{Impl} \sqsubseteq_{\mathrm{RS}} \mathtt{SW}_1 \parallel \mathtt{SW}_2 \parallel \mathtt{AS}$ and $\mathtt{Impl} \sqsubseteq_{\mathrm{RS}} \mathtt{C}_{\mathtt{on}}$ which implies $\mathtt{Impl} \sqsubseteq_{\mathrm{RS}} \mathtt{Spec}$ by Thm. 16.

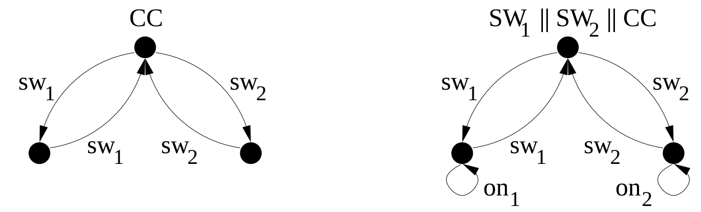

Fig. 7. Central controller $\mathtt{CC}$ and composition with switches $\mathtt{SW}_1$ and $\mathtt{SW}_2$.

The first step is to refine the abstract scheduler $\mathtt{AS}$ by a concrete central controller $\mathtt{CC}$ that guarantees the constraint $\mathtt{C}_{\mathtt{on}}$. This controller is depicted on the left in Fig. 7. Obviously, $\mathtt{CC} \sqsubseteq_{\mathrm{RS}} \mathtt{AS}$ and hence

$$\mathtt{SW}_1 \parallel \mathtt{SW}_2 \parallel \mathtt{CC} \sqsubseteq_{\mathrm{RS}} \mathtt{SW}_1 \parallel \mathtt{SW}_2 \parallel \mathtt{AS}$$

by compositionality of ready simulation for parallel composition (cf. Thm. 14). Furthermore, it is easy to check that

$$\mathtt{SW}_1 \parallel \mathtt{SW}_2 \parallel \mathtt{CC} \sqsubseteq_{\mathrm{RS}} \mathtt{C}_{\mathtt{on}}\,,$$

when considering the Logic LTS of this parallel composition depicted on the right in Fig. 7. By Thm. 16, we conclude that

$$\mathtt{SW}_1 \parallel \mathtt{SW}_2 \parallel \mathtt{CC} \sqsubseteq_{\mathrm{RS}} \mathtt{Spec}\,.$$

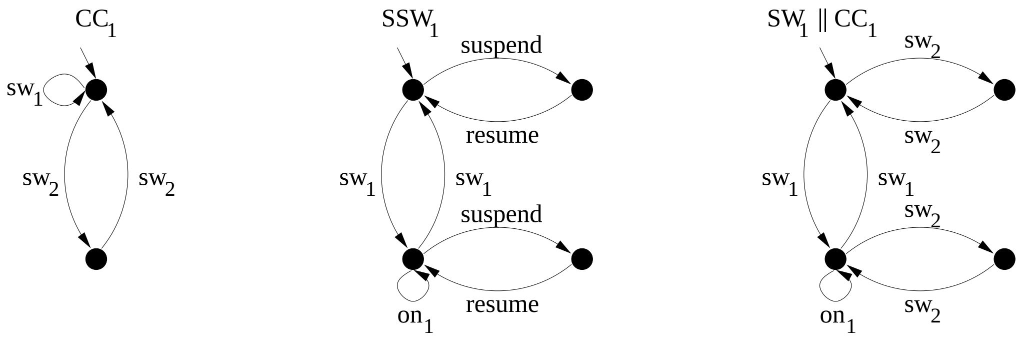

Fig. 8. Basic switch controller $\mathtt{CC}_1$, suspendable switch $\mathtt{SSW}_1$ and $\mathtt{SW}_1 \parallel \mathtt{CC}_1$.

To obtain an implementation with distributed control, we now observe that $\mathtt{CC}$ can be implemented by the parallel composition of two basic switch controllers $\mathtt{CC}_1$ and $\mathtt{CC}_2$, where $\mathtt{CC}_1$ is shown on the left in Fig. 8 and where $\mathtt{CC}_2$ is defined analogous to $\mathtt{CC}_1$ but with indices 1 and 2 interchanged. Observe that $\mathtt{CC} =_{\mathrm{RS}} \mathtt{CC}_1 \parallel \mathtt{CC}_2$. We define

$$\mathtt{Impl} =_{\mathrm{df}} (\mathtt{SW}_1 \parallel \mathtt{CC}_1) \parallel (\mathtt{SW}_2 \parallel \mathtt{CC}_2)$$

and can now infer the desired refinement result by referring to commutativity and associativity, as well as to the already established equivalence and refinement steps:

$$\mathtt{Impl} =_{\mathrm{RS}} (\mathtt{SW}_1 \parallel \mathtt{SW}_2) \parallel (\mathtt{CC}_1 \parallel \mathtt{CC}_2) \;=_{\mathrm{RS}}\; (\mathtt{SW}_1 \parallel \mathtt{SW}_2) \parallel \mathtt{CC} \;\sqsubseteq_{\mathrm{RS}}\; \mathtt{Spec}\,.$$

As an aside, our implementation $\mathtt{Impl}$ can also be understood as being built from two suspendable switches $\mathtt{SSW}_1$ and $\mathtt{SSW}_2$ which might be available off-the-shelf, as shown in the centre of Fig. 8, where $\mathtt{SSW}_2$ is defined as $\mathtt{SSW}_1$ but with indices 1 and 2 interchanged. $\mathtt{SSW}_1$ can be suspended in any state, and has the behaviour of $\mathtt{SW}_1 \parallel \mathtt{CC}_1$ when the `suspend` and `resume` signals are both connected (i.e., renamed) to $\mathtt{sw}_2$.

5 Further operators: disjunction, external choice and hiding

The aim of this section is to show that Logic LTS and ready simulation build a *robust* framework for mixing process-algebraic and logic styles of specification and reasoning. To this end, we extend our framework by the logic operator *disjunction* and the process-algebraic operators *external choice* and *hiding*. We omit the extension by the process-algebraic *action prefix* operator, which is trivial.

While extending our framework is relatively straightforward for disjunction and external choice, defining hiding and proving this operator compositional for ready simulation turns out to be a challenge. This is because hiding a visible action of a Logic LTS, i.e., relabelling that action by the internal action τ, may make the LTS τ-impure, not only on top-level but also deep inside the LTS. Hence, some kind of semantics-preserving transformation (as already indicated in Fig. 3) that re-instates τ-purity is required when applying hiding.

5.1 Disjunction

We start off with defining disjunction on Logic LTS. As suggested in Sec. 2, disjunction intuitively corresponds to internal choice.

Definition 22 (Disjunction operator) *The disjunction of two Logic LTSs $\langle P, \longrightarrow_P, F_P\rangle$ and $\langle Q, \longrightarrow_Q, F_Q\rangle$ satisfying (w.l.o.g.) $P \cap Q = \emptyset$, is the Logic LTS $\langle P \vee Q, \longrightarrow_{P\vee Q}, F_{P\vee Q}\rangle$ defined by:*

- $P \vee Q =_{df} \{p \vee q \,|\, p \in P,\ q \in Q\} \cup P \cup Q$
- $\longrightarrow_{P\vee Q}$ *is determined by the following operational rules:*

$$\begin{array}{rcl}
\textit{always} & \Longrightarrow & p \vee q \xrightarrow{\tau}_{P\vee Q} p \\
\textit{always} & \Longrightarrow & p \vee q \xrightarrow{\tau}_{P\vee Q} q \\
p \xrightarrow{\alpha}_P p' & \Longrightarrow & p \xrightarrow{\alpha}_{P\vee Q} p' \\
q \xrightarrow{\alpha}_Q q' & \Longrightarrow & q \xrightarrow{\alpha}_{P\vee Q} q'
\end{array}$$

- $F_{P\vee Q} = F_P \cup F_Q \cup \{p \vee q \,|\, p \in F_P \textit{ and } q \in F_Q\}$

It is easy to see that disjunction is well-defined, i.e., that the disjunction of Logic LTS is again a Logic LTS. The disjunction operator is also compositional for ready simulation:

Theorem 23 (Compositionality for disjunction) *Let $p \sqsubseteq_{RS} q$ and r be an arbitrary process. Then, $p \vee r \sqsubseteq_{RS} q \vee r$.*

The compositionality proof is straightforward and therefore omitted. As stated in the following proposition, ready simulation also satisfies the desired logic properties for disjunction. Most importantly, the disjunction operator expresses indeed disjunction on Logic LTS and is thus not defined arbitrarily.

Proposition 24 (Logic properties of disjunction)

(1) $p \vee q \sqsubseteq_{RS} r$ *if and only if* $p \sqsubseteq_{RS} r$ *and* $q \sqsubseteq_{RS} r$ *("$\vee$ is or");*
(2) $p \sqsubseteq_{RS} p \vee q$*;*
(3) $p \vee p =_{RS} p$*;*
(4) $p \vee q =_{RS} q$ *if and only if* $p \sqsubseteq_{RS} q$*.*

Again, we omit the straightforward proofs of these properties, which simply refer to our definitions of disjunction and ready simulation. Instead, we state and prove two other properties relating disjunction and conjunction:

Proposition 25 (Distributivity laws)

(1) $p \wedge (q \vee r) =_{RS} (p \wedge q) \vee (p \wedge r)$
(2) $p \vee (q \wedge r) =_{RS} (p \vee q) \wedge (p \vee r)$

PROOF.

(1, $\sqsubseteq_{\mathbf{RS}}$) Let $p \wedge (q \vee r) \stackrel{\epsilon}{\Longrightarrow}\!| \, p' \wedge s$; w.l.o.g., $p \stackrel{\epsilon}{\Longrightarrow}\!| \, p'$ and $q \stackrel{\epsilon}{\Longrightarrow}\!| \, s$. Since $p' \wedge s \notin F$, Lemma 5(2) implies $(p \wedge q) \vee (p \wedge r) \xrightarrow{\tau} p \wedge q \stackrel{\epsilon}{\Longrightarrow}\!| \, p' \wedge s$. Due to the τ-step, $(p \wedge q) \vee (p \wedge r) \notin F$ since $p \wedge q \notin F$. We may therefore conclude $(p \wedge q) \vee (p \wedge r) \stackrel{\epsilon}{\Longrightarrow}\!| \, p' \wedge s$ and, trivially, $p' \wedge s \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} p' \wedge s$.

(1, $\sqsupseteq_{\mathbf{RS}}$) Let $(p \wedge q) \vee (p \wedge r) \stackrel{\epsilon}{\Longrightarrow}\!| \, s$; w.l.o.g., $p \wedge q \stackrel{\epsilon}{\Longrightarrow}\!| \, s$. Then, $p \wedge (q \vee r) \xrightarrow{\tau} p \wedge q \stackrel{\epsilon}{\Longrightarrow}\!| \, s$, as well as $p \wedge (q \vee r) \notin F$ by Lemma 5(1). We may thus conclude $p \wedge (q \vee r) \stackrel{\epsilon}{\Longrightarrow}\!| \, s$ and, trivially, $s \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} s$.

(2, $\sqsubseteq_{\mathbf{RS}}$) If $p \vee (q \wedge r) \stackrel{\epsilon}{\Longrightarrow}\!| \, p'$ due to $p \stackrel{\epsilon}{\Longrightarrow}\!| \, p'$, then $(p \vee q) \wedge (p \vee r) \xrightarrow{\tau} p \wedge (p \vee r) \xrightarrow{\tau} p \wedge p \stackrel{\epsilon}{\Longrightarrow}\!| \, p' \wedge p'$. Now, we obtain $p' \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} p' \wedge p'$ by Thm. 16(1), as well as $(p \vee q) \wedge (p \vee r) \notin F$ and $p \wedge (p \vee r) \notin F$ by Lemma 5(1) due to $p \notin F$.

If $p \vee (q \wedge r) \stackrel{\epsilon}{\Longrightarrow}\!| \, q' \wedge r'$ due to $q \wedge r \stackrel{\epsilon}{\Longrightarrow}\!| \, q' \wedge r'$, then $(p \vee q) \wedge (p \vee r) \xrightarrow{\tau} q \wedge (p \vee r) \xrightarrow{\tau} q \wedge r \stackrel{\epsilon}{\Longrightarrow}\!| \, q' \wedge r'$. Since $q \wedge r \notin F$, we have $q, r \notin F$ and $p \vee q, p \vee r \notin F$, and we are done with Lemma 5(1) since then $(p \vee q) \wedge (p \vee r) \stackrel{\epsilon}{\Longrightarrow}\!| \, q' \wedge r'$ and, trivially, $q' \wedge r' \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q' \wedge r'$.

(2, $\sqsupseteq_{\mathbf{RS}}$) Let $(p \vee q) \wedge (p \vee r) \stackrel{\epsilon}{\Longrightarrow}\!| \, p' \wedge p''$ for some p', p'' due to $p \stackrel{\epsilon}{\Longrightarrow}\!| \, p'$ and $p \stackrel{\epsilon}{\Longrightarrow}\!| \, p''$. Hence, $p \vee (q \wedge r) \xrightarrow{\tau} p \stackrel{\epsilon}{\Longrightarrow}\!| \, p'$, as well as $p \vee (q \wedge r) \notin F$. Finally, observe $p' \wedge p'' \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} p'$ by Prop. 18(2).

The cases $(p \vee q) \wedge (p \vee r) \stackrel{\epsilon}{\Longrightarrow}\!| \, q' \wedge p'$, for some $q \stackrel{\epsilon}{\Longrightarrow}\!| \, q'$ and $p \stackrel{\epsilon}{\Longrightarrow}\!| \, p'$, and $(p \vee q) \wedge (p \vee r) \stackrel{\epsilon}{\Longrightarrow}\!| \, p' \wedge r'$, for some $p \stackrel{\epsilon}{\Longrightarrow}\!| \, p'$ and $r \stackrel{\epsilon}{\Longrightarrow}\!| \, r'$, are analogous.

Finally, let $(p \vee q) \wedge (p \vee r) \overset{\epsilon}{\Longrightarrow}\!| \, q' \wedge r'$ for some q', r' such that $q \overset{\epsilon}{\Longrightarrow}\!| \, q'$ and $r \overset{\epsilon}{\Longrightarrow}\!| \, r'$. Due to $q' \wedge r' \notin F$ we can apply Lemma 5(2) to obtain $p \vee (q \wedge r) \xrightarrow{\tau} q \wedge r \overset{\epsilon}{\Longrightarrow}\!| \, q' \wedge r'$. Since $q \wedge r \notin F$, we have $p \vee (q \wedge r) \notin F$, as required. Hence, $p \vee (q \wedge r) \overset{\epsilon}{\Longrightarrow}\!| \, q' \wedge r'$ and, trivially, $q' \wedge r' \sqsubset\!\!\!\sim_{\mathrm{RS}} q' \wedge r'$. $\Box$

5.2 External choice

We now turn to integrating the process-algebraic external choice operator into our framework. The only subtlety arises when composing an instable process with a stable one, as the usual definition of external choice would offer the initial actions of both processes, thus violating τ-purity. As for parallel composition, we resolve this issue by giving τ-transitions precedence over visible transitions:

Definition 26 (External choice operator) *The external choice on Logic LTSs $\langle P, \longrightarrow_P, F_P \rangle$ and $\langle Q, \longrightarrow_Q, F_Q \rangle$ satisfying (w.l.o.g.) $P \cap Q = \emptyset$, is the Logic LTS $\langle P \Box Q, \longrightarrow_{P \Box Q}, F_{P \Box Q} \rangle$ defined by:*

- *$P \Box Q =_{df} \{p \Box q \,|\, p \in P,\ q \in Q\} \cup P \cup Q$*
- *$\longrightarrow_{P \Box Q}$ is determined by the following operational rules:*

$$\begin{array}{rcl}
p \xrightarrow{\tau}_P p' & \Longrightarrow & p \Box q \xrightarrow{\tau}_{P \Box Q} p' \Box q \\
q \xrightarrow{\tau}_Q q' & \Longrightarrow & p \Box q \xrightarrow{\tau}_{P \Box Q} p \Box q' \\
p \xrightarrow{a}_P p',\ q \not\xrightarrow{\tau}_Q & \Longrightarrow & p \Box q \xrightarrow{a}_{P \Box Q} p' \\
q \xrightarrow{a}_Q q',\ p \not\xrightarrow{\tau}_P & \Longrightarrow & p \Box q \xrightarrow{a}_{P \Box Q} q' \\
p \xrightarrow{\alpha}_P p' & \Longrightarrow & p \xrightarrow{\alpha}_{P \Box Q} p' \\
q \xrightarrow{\alpha}_Q q' & \Longrightarrow & q \xrightarrow{\alpha}_{P \Box Q} q'
\end{array}$$

- *$F_{P \Box Q} = F_P \cup F_Q \cup \{p \Box q \,|\, p \in F_P \text{ or } q \in F_Q\}$*

Note that external choice only resolves choices for external, i.e., visible actions, and not for the internal action τ. Moreover, $p \Box q$ is already inconsistent if one of p or q is inconsistent, whereas the inconsistency of $p \vee q$ requires both p and q to be inconsistent. This is due to the fact that external choice is not a logic concept but an operational one. Our external choice operator is well-defined and, most importantly, it is compositional for ready simulation:

Theorem 27 (Compositionality for external choice)

(1) Let $p \sqsubset\!\!\!\sim_{RS} q$ and r be stable. Then, $p \Box r \sqsubset\!\!\!\sim_{RS} q \Box r$.
(2) Let $p \sqsubseteq_{RS} q$ and r be an arbitrary process. Then, $p \Box r \sqsubseteq_{RS} q \Box r$.

PROOF. In order to establish Part (1), it is sufficient to prove that

$$\mathcal{R}_{\Box} =_{\mathrm{df}} \{\langle p \Box r, q \Box r\rangle \mid r \text{ stable and } p \underset{\sim}{\sqsubset}_{\mathrm{RS}} q\} \;\cup\; \underset{\sim}{\sqsubset}_{\mathrm{RS}}$$

is a stable ready simulation relation. Let $\langle p \Box r, q \Box r\rangle \in \mathcal{R}_{\Box}$ due to $p \underset{\sim}{\sqsubset}_{\mathrm{RS}} q$ and r stable. We have to verify Conds. (RS1)–(RS4) of Def. 11:

(RS1) Trivial, since p, q, r are all stable.

(RS2) $p \Box r \notin F \implies p, r \notin F \implies q, r \notin F$ (by $p \underset{\sim}{\sqsubset}_{\mathrm{RS}} q$ and Def. 11(RS1)) $\implies q \Box r \notin F$.

(RS3) Let $p \Box r \overset{a}{\Longrightarrow}\!| \, p'$ due to $p \overset{a}{\Longrightarrow}\!| \, p'$. According to our definition of the transition relation for external choice, the latter computation has the same states as the former one except for p. Hence, by $p \underset{\sim}{\sqsubset}_{\mathrm{RS}} q$ and Def. 11(RS3), we obtain some q' such that $q \overset{a}{\Longrightarrow}\!| \, q'$ and $p' \underset{\sim}{\sqsubset}_{\mathrm{RS}} q'$, whence $q \Box r \overset{a}{\Longrightarrow}\!| \, q'$ (cf. (RS2)) and $\langle p', q'\rangle \in \mathcal{R}_{\Box}$.

The case of $p \Box r \overset{a}{\Longrightarrow}\!| \, r'$ due to $r \overset{a}{\Longrightarrow}\!| \, r'$ is straightforward.

(RS4) $p \Box r \notin F \implies p \notin F \implies \mathcal{I}(p) = \mathcal{I}(q)$ (by $p \underset{\sim}{\sqsubset}_{\mathrm{RS}} q$ and Def. 11(RS4)) $\implies \mathcal{I}(p \Box r) = \mathcal{I}(p) \cup \mathcal{I}(r) = \mathcal{I}(q) \cup \mathcal{I}(r) = \mathcal{I}(q \Box r)$.

To prove Part (2), consider $p \Box r \overset{\epsilon}{\Longrightarrow}\!| \, p' \Box r'$ due to an interleaving of $p \overset{\epsilon}{\Longrightarrow}\!| \, p'$ and $r \overset{\epsilon}{\Longrightarrow}\!| \, r'$. Because of $p \sqsubseteq_{\mathrm{RS}} q$ we have some q' such that $q \overset{\epsilon}{\Longrightarrow}\!| \, q'$ and $p' \underset{\sim}{\sqsubset}_{\mathrm{RS}} q'$, implying $q \Box r \overset{\epsilon}{\Longrightarrow}\!| \, q' \Box r'$ and, by Part (1), $p' \Box r' \underset{\sim}{\sqsubset}_{\mathrm{RS}} q' \Box r'$. $\Box$

5.3 Hiding

Adding the standard process-algebraic operator $/h$ of hiding into our setting, where h is a visible action, turns out to be much more challenging than, say, integrating the external choice operator. This is because hiding an action, i.e., relabelling a visible action by the internal action τ, typically destroys τ-purity not only at the top-level of a Logic LTS but also at deeper levels. Hence, in order for the hiding operator to be well-defined, it must not only hide the desired action but also re-establish τ-purity. As this is non-trivial, we start off by some simple examples that motivate our subsequent formal definition of hiding.

As the first example, re-consider the LTS shown on the very left in Fig. 3, but think of the τ-labelled transition as the result of hiding the visible action h. Recall our previous discussion on why understanding this transition system is difficult.[8] However, this τ-impure transition system can be expressed 'equivalently' as the τ-pure LTS on the very right in Fig. 3, as explained in Sec. 2. Operationally, this transformation from the LTS on the left to the LTS on the right can be understood as collecting all moves that are possible when "looking through" the hidden action h or pre-empting action a by performing h.

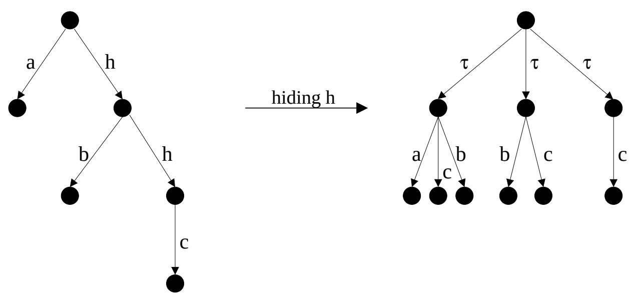

Fig. 9. Second example regarding hiding.

The second example depicted in Fig. 9 requires us to iterate this idea when hiding action h in the LTS on the left, to obtain the LTS on the right. Note that, in the LTS on the right, the target states of the b-transitions are actually the same state, as are the target states of the c-transitions. We have drawn the LTS as a tree for the sole purpose to improve the LTS's layout, which is also the case for the rightmost LTS in Fig. 10.

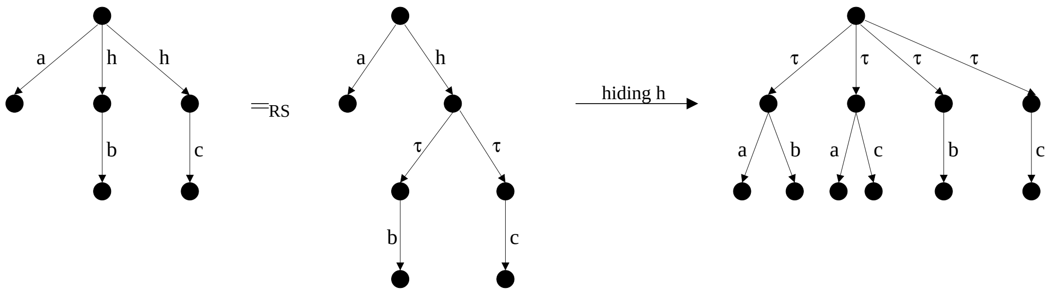

Fig. 10. Third example regarding hiding.

In the final example regarding hiding, depicted in Fig. 10, we first observe the indicated equality, which translates the internal choice on action h to an h-step followed by disjunction. This rewriting makes it clear that actions b and c are exclusive alternatives, and each of these may be combined with action a when hiding h as shown.

For the process algebraist, we briefly indicate the 'equational' rationale behind these examples. The transformation in the first example is based on the following law of failure semantics [9]: $(p \Box h.q)/h = (p \Box q)/h \vee q/h$. This law has been applied three times in our second example above, together with some other obvious laws. However, note that the law of failure semantics does not hold

[8] In particular, giving action τ precedence in this LTS as suggested after Def. 6, essentially removes the a-transition, which we consider to be counter-intuitive.

for ready simulation in case p can engage in an initial h-transition; otherwise, we would have:

$$\begin{aligned}(h.a\Box h.b)/h &= (h.a\Box b)/h \vee b = (a\Box b) \vee a \vee b \\ (h.a\Box h.b)/h &= \ (h.(a \vee b)/h \ = \qquad a \vee b\end{aligned}$$

But $(a\Box b) \vee a \vee b \neq_{\mathrm{RS}} a \vee b$. However, this issue is not a problem since several h-transitions can be merged into one with the help of internal choice (disjunction), as shown in the third example above.

In order to formalise our intuition of hiding and to collect all moves "looking through" h as described above, we have to introduce new processes that are tuples of processes. Formally, given a Logic LTS $\langle P, \longrightarrow_P, F_P\rangle$, we augment P to the set $\hat{P} = P \cup \vec{P}$ by adding all vectors of stable, consistent processes, i.e.,

$$\vec{P} =_{\mathrm{df}} \{\vec{p} = (p_1, p_2, \ldots, p_n) \,|\, n \geq 1,\ \forall 1 \leq i \leq n.\, p_i \in P \text{ stable and } p_i \notin F_P\}\,.$$

We use the conventions that $\hat{p}$ is a generic process in $\hat{P}$, $p \in \hat{P}$ is a process from P, and $\vec{p} \in \hat{P}$ is a process from $\vec{P}$ with components $p_1, p_2, \ldots, p_n$. Moreover, we consider every process $\vec{p} \in \vec{P}$ as consistent and augment the transition relation by transitions according to the following rule:

$$\text{(T)} \qquad p_i \xrightarrow{a} p \qquad \Longrightarrow \qquad \vec{p} \xrightarrow{a} p\,,$$

so that a vector of processes inherits all transitions of its component processes. It is easy to see that the resulting Logic LTS is indeed a Logic LTS. Firstly, τ-purity and (LTS2) are not violated since all $\vec{p} \in \vec{P}$ are stable. Secondly, let $\vec{p} \xrightarrow{a}$, i.e., there exists some i such that $p_i \xrightarrow{a}$. By (LTS1) for $p_i \in P$ we have the existence of some $p \notin F_P$ such that $p_i \xrightarrow{a} p$. Hence, also $\vec{p} \xrightarrow{a} p \notin F$, which establishes (LTS1) for our extension.

Similarly to extending Logic LTS, we can extend the stable ready simulation $\precsim_{\mathrm{RS}} \subseteq P \times Q$ component-wise, to obtain a new relation $\hat{\precsim}_{\mathrm{RS}} \subseteq \hat{P} \times \hat{Q}$. Formally, we add to $\precsim_{\mathrm{RS}}$ all pairs $\langle \vec{p}, \vec{q}\rangle \in \vec{P} \times \vec{Q}$ of tuples of equal length such that $\forall i.\, p_i \precsim_{\mathrm{RS}} q_i$. As expected, $\hat{\precsim}_{\mathrm{RS}}$ is a stable ready simulation relation:

(RS1) Straightforward by Rule (T).

(RS2) If $\vec{p} \,\hat{\precsim}_{\mathrm{RS}}\, \vec{q}$, then both $\vec{p}, \vec{q} \notin F$.

(RS3) If $\vec{p} \,\hat{\precsim}_{\mathrm{RS}}\, \vec{q}$, then $\vec{p} \overset{a}{\Longrightarrow}\!| \, p'$, i.e., $\vec{p} \xrightarrow{a}_{\mathrm{F}} p \overset{\epsilon}{\Longrightarrow}\!| \, p'$ for some p, implies $p_i \xrightarrow{a}_{\mathrm{F}} p \overset{\epsilon}{\Longrightarrow}\!| \, p'$ by Rule (T), i.e., $p_i \overset{a}{\Longrightarrow}\!| \, p'$. Since $p_i \precsim_{\mathrm{RS}} q_i$, we have $q_i \overset{a}{\Longrightarrow}\!| \, q'$ for some q' with $p' \precsim_{\mathrm{RS}} q'$. By Rule (T), also $\vec{q} \overset{a}{\Longrightarrow}\!| \, q'$, as well as $p' \,\hat{\precsim}_{\mathrm{RS}}\, q'$ by the definition of $\hat{\precsim}_{\mathrm{RS}}$.

(RS4) If $\vec{p} \mathrel{\hat{\sqsubseteq}}_{\mathrm{RS}} \vec{q}$, then $p_i \sqsubseteq_{\mathrm{RS}} q_i$ and $p_i \notin F$ for all i; hence, $\mathcal{I}(p_i) = \mathcal{I}(q_i)$ by (RS4) for $p_i \sqsubseteq_{\mathrm{RS}} q_i$. Thus, $\mathcal{I}(\vec{p}) = \bigcup_i \mathcal{I}(p_i) = \bigcup_i \mathcal{I}(q_i) = \mathcal{I}(\vec{q})$.

Hence, our extensions of Logic LTS and stable ready simulation to tuples of stable, consistent processes are well-defined.

We are now in a position to formally define our hiding operator:

Definition 28 (Hiding operator) *The hiding of visible action $h \in \mathcal{A}$ within a Logic LTS $\langle P, \longrightarrow_P, F_P \rangle$ is the Logic LTS $\langle P/h, \longrightarrow_{P/h}, F_{P/h} \rangle$ defined by:*

- $P/h =_{df} \{\hat{p}/h \,|\, \hat{p} \in \hat{P}\}$
- $\longrightarrow_{P/h}$ *is determined by the following operational rules:*

$$\textit{(H1)} \qquad p \xrightarrow{\tau}_P p' \implies p/h \xrightarrow{\tau}_{P/h} p'/h$$

$$\textit{(H2)} \qquad p \xrightarrow{a}_P p',\ p \overset{h}{\not\longrightarrow}_P \implies p/h \xrightarrow{a}_{P/h} p'/h$$

$$\textit{(H3)}\ p \xrightarrow{h}_P,\ p(\overset{h}{\Longrightarrow}\!|_P)^* p_1 \overset{h}{\Longrightarrow}\!|_P\, p_2 \cdots \overset{h}{\Longrightarrow}\!|_P\, p_n \overset{h}{\not\longrightarrow}_P \implies p/h \xrightarrow{\tau}_{P/h} (p_1, p_2, \ldots, p_n)/h$$

$$\textit{(H4)} \qquad p_j \xrightarrow{a}_P p',\ a \neq h,\ (p_1, p_2, \ldots, p_n) \in P/h \implies (p_1, p_2, \ldots, p_n)/h \xrightarrow{a}_{P/h} p'/h$$

- $F_{P/h}$ *is the least set satisfying the following conditions:*

(1) $p/h \in F_{P/h}$ *if* $p \in F_P$*;*
(2) $p/h \in F_{P/h}$ *if* $\nexists p'.\, p \overset{\epsilon}{\Longrightarrow}\!|_P (\overset{h}{\Longrightarrow}\!|_P)^* p' \overset{h}{\not\longrightarrow}_P$*;*
(3) $\hat{p}/h \in F_{P/h}$ *if* $\exists \alpha \in \mathcal{I}_{P/h}(\hat{p}/h)\, \forall \hat{p}'/h.\ \hat{p}/h \xrightarrow{\alpha}_{P/h} \hat{p}'/h \implies \hat{p}'/h \in F_{P/h}$*;*
(4) $\hat{p}/h \in F_{P/h}$ *if* $\hat{p}/h$ *cannot stabilise outside* $F_{P/h}$*.*

Intuitively, the first two of the above conditions for inconsistency of hiding explain where inconsistencies may come from: they may either be inherited from P, or may result from a new inescapable divergence that arises when hiding h. The latter two conditions are needed to ensure Conds. (LTS1) and (LTS2) of Logic LTS. It can easily be checked that hiding is well-defined, i.e., that hiding a visible action in a Logic LTS results again in a Logic LTS.

To be able to reason about the inconsistencies that may arise when hiding actions, we employ again a notion of *witness*, as in our approach to conjunction:

Definition 29 (Hiding-witness) *A* hiding-witness *is a set $W \subseteq P/h$ such that the following conditions hold:*

(HW1) $\forall p/h \in W.\ p \notin F_P$ *and* $\exists p'.\, p \overset{\epsilon}{\Longrightarrow}\!|_P (\overset{h}{\Longrightarrow}\!|_P)^* p' \overset{h}{\not\longrightarrow}_P$*;*
(HW2) $\forall \hat{p}/h \in W.$ *(a)* $\forall \alpha \in \mathcal{I}_{P/h}(\hat{p}/h)\, \exists \hat{p}'/h \in W.\ \hat{p}/h \xrightarrow{\alpha}_{P/h} \hat{p}'/h$*;*
(b) $\hat{p}/h$ *can stabilise in* W*.*

The statement and proof of the following proposition is analogous to the one of Prop. 4 for $\wedge$-witnesses:

Proposition 30 *$\hat{p}/h \notin F_{P/h}$ if and only if $\exists$hiding-witness$W.\ \hat{p}/h \in W$.*

We are now in a position to define a particular hiding-witness that will be needed for proving ready simulation compositional with respect to hiding.

Lemma 31 *Let $\langle P, \longrightarrow_P, F_P \rangle$ and $\langle Q, \longrightarrow_Q, F_Q \rangle$ be Logic LTS and $h \in \mathcal{A}$. Then, the set $W'' =_{df} W_1'' \cup W_2''$ is a hiding-witness for $\langle Q/h, \longrightarrow_{Q/h}, F \rangle$, where*

$$W_1'' =_{df} \{\hat{q}/h \in Q/h \,|\, \exists \hat{p} \in \hat{P}.\ \hat{p} \mathrel{\hat{\underset{\sim}{\sqsubset}}_{RS}} \hat{q} \text{ and } \hat{p}/h \notin F\};$$
$$W_2'' =_{df} \{\hat{q}/h \in Q/h \,|\, \exists \hat{q}''.\ \hat{q} \stackrel{\tau}{\Longrightarrow}\!|\, \hat{q}'' \text{ and } \hat{q}''/h \in W_1''\}\,.$$

Note that $\hat{q}$ in W_2'' must necessarily be of the form q and cannot be the vector $\vec{q}$.

PROOF. To establish Cond. (HW1) of Def. 29, let us first consider $q/h \in W_1''$ due to p. Note that $p/h \notin F$ implies $p \notin F_P$ and $q \notin F_Q$ by $p \mathrel{\underset{\sim}{\sqsubset}_{\mathrm{RS}}} q$ and (RS2). Then, since $p/h \notin F$ implies $\exists p'.\ p(\stackrel{h}{\Longrightarrow}\!|\,)^* p' \not\xrightarrow{h}$ we find, by $p \mathrel{\underset{\sim}{\sqsubset}_{\mathrm{RS}}} q$ and (RS3), some q' such that $q(\stackrel{h}{\Longrightarrow}\!|\,)^* q'$ and $p' \mathrel{\underset{\sim}{\sqsubset}_{\mathrm{RS}}} q'$. Since $p' \notin F_P$, we obtain $q' \not\xrightarrow{h}$ by (RS4).

Now, let us consider the case $q/h \in W_2''$ due to $q \stackrel{\tau}{\Longrightarrow}\!|\, \hat{q}''$ with $\hat{q}''/h \in W_1''$. Obviously, $q \notin F_Q$. Furthermore, $\hat{q}''$ has the form q'', and we have just shown that $\exists q'.\ q''(\stackrel{h}{\Longrightarrow}\!|\,)^* q' \not\xrightarrow{h}$. Hence, $q \stackrel{\epsilon}{\Longrightarrow}\!|\, q''(\stackrel{h}{\Longrightarrow}\!|\,)^* q' \not\xrightarrow{h}$.

To verify Cond. (HW2a), consider some $\alpha \in \mathcal{I}(\hat{q}/h)$ and distinguish the following cases:

<u>$\alpha \neq \tau$:</u> Then $\hat{q}/h \in W_1''$ due to some $\hat{p}$. Note that it cannot be the case that $\hat{q}/h \in W_2''$; assume otherwise, $\hat{q} \xrightarrow{\tau}$ and thus also $\hat{q}/h \xrightarrow{\tau}$ by (H1), contradicting $\hat{q}/h \xrightarrow{\alpha}$ with $\alpha \neq \tau$.

Again, we distinguish two cases, for both of which we establish $\hat{p}/h \xrightarrow{\alpha}$. The first case is $\hat{q}/h \xrightarrow{\alpha}$ due to (H2). Then, $\hat{q} \xrightarrow{\alpha}$ and $\hat{q} \not\xrightarrow{h}$. By $\hat{p} \mathrel{\hat{\underset{\sim}{\sqsubset}}_{\mathrm{RS}}} \hat{q}$ and $\hat{p} \notin F_P$, we have $\hat{p} \xrightarrow{\alpha}$ and $\hat{p} \not\xrightarrow{h}$ by (RS4). Hence, $\hat{p}/h \xrightarrow{\alpha}$.

The second case is $\hat{q}/h \xrightarrow{\alpha}$ due to (H4). Take some i with $q_i \xrightarrow{\alpha}$. Since $p_i \notin F_P$ by $\hat{p} \in \hat{P}$ and since $p_i \mathrel{\underset{\sim}{\sqsubset}_{\mathrm{RS}}} q_i$ by $\hat{p} \mathrel{\hat{\underset{\sim}{\sqsubset}}_{\mathrm{RS}}} \hat{q}$, we have $p_i \xrightarrow{\alpha}$ and $\hat{p}/h \xrightarrow{\alpha}$.

Now we may conclude the proof for both cases uniformly. By (LTS1) and (LTS2), $\hat{p}/h \stackrel{\alpha}{\Longrightarrow}\!|$. Let $\hat{p}''/h \notin F$ be the first state on this computation with $\hat{p}''$ stable, whence $\hat{p} \stackrel{\alpha}{\Longrightarrow}\!|\, \hat{p}''$. By $\hat{p} \mathrel{\hat{\underset{\sim}{\sqsubset}}_{\mathrm{RS}}} \hat{q}$, we also have some $\hat{q}''$ and q'

with $\hat{q} \xrightarrow{\alpha}_{\mathrm{F}} q' \overset{\epsilon}{\Longrightarrow}| \hat{q}''$ and $\hat{p}'' \mathrel{\hat{\underset{\sim}{\sqsubset}}_{\mathrm{RS}}} \hat{q}''$. Since $\hat{p}''/h \notin F$, we have $\hat{q}''/h \in W_1''$. Thus, $\hat{q}/h \xrightarrow{\alpha} q'/h \in W''$.

<u>$\alpha = \tau$:</u> If $\hat{q}/h \in W_1''$ due to $\hat{p}$, then $\hat{q}$, being stable, must be q with $q \xrightarrow{h}$ and $\hat{p} = p$. Since $p/h \notin F$, we have $p \notin F_P$ and, by (RS4), $p \xrightarrow{h}$; furthermore, $p(\overset{h}{\Longrightarrow}|)^* p' \overset{h}{\not\longrightarrow}$ for some p'. Thus, $p/h \xrightarrow{\tau}$ by (H3) and $p/h \xrightarrow{\tau} \vec{p}/h \notin F$ for some suitable $\vec{p}/h$ by (LTS1), with $p(\overset{h}{\Longrightarrow}|)^* p_1 \overset{h}{\Longrightarrow}| p_2 \cdots \overset{h}{\Longrightarrow}| p_n \overset{h}{\not\longrightarrow}$ and $\vec{p} = (p_1, p_2, \dots, p_n)$. From the assumption $p \mathrel{\underset{\sim}{\sqsubset}_{\mathrm{RS}}} q$ we conclude by (RS3) that $q(\overset{h}{\Longrightarrow}|)^* q_1 \overset{h}{\Longrightarrow}| q_2 \cdots \overset{h}{\Longrightarrow}|$, $p_i \mathrel{\underset{\sim}{\sqsubset}_{\mathrm{RS}}} q_i$ for all $1 \leq i \leq n$, and $q_n \overset{h}{\not\longrightarrow}$ by (RS4). Thus, by (H3), $q/h \xrightarrow{\tau} \vec{q}/h$ and $\vec{p} \mathrel{\hat{\underset{\sim}{\sqsubset}}_{\mathrm{RS}}} \vec{q}$, i.e., $\vec{q}/h \in W_1'' \subseteq W''$.

If $\hat{q}/h \in W_2''$, then the state $\hat{q}'$ succeeding $\hat{q}$ on the respective computation $\hat{q} \overset{\tau}{\Longrightarrow}| \hat{q}''$ satisfies $\hat{q} \xrightarrow{\tau} \hat{q}'$ and $\hat{q}'/h \in W_2''$, or $\hat{q}'/h = \hat{q}''/h \in W_1''$.

To establish Cond. (HW2b) we can assume that we are in the case $\alpha = \tau$ above. Thus, either $\hat{q}/h \xrightarrow{\tau} \vec{q}/h \in W''$ and $\vec{q}/h$ is stable; or $\hat{q}/h \overset{\epsilon}{\Longrightarrow} \hat{q}''/h$ with all states in W'' and $\hat{q}''$ stable. If $\hat{q}''/h$ is not stable, it can stabilise in W'' with some $\hat{q}''/h \xrightarrow{\tau} \vec{q}/h$ as in the 'either' case. $\square$

We can now prove the desired compositionality result for ready simulation with respect to hiding:

Theorem 32 (Compositionality for hiding)

(1) Let $p \mathrel{\underset{\sim}{\sqsubset}_{RS}} q$ and $h \in \mathcal{A}$ with $p \overset{h}{\not\longrightarrow}$ and $q \overset{h}{\not\longrightarrow}$. Then, $p/h \mathrel{\underset{\sim}{\sqsubset}_{RS}} q/h$.
(2) Let $p \sqsubseteq_{RS} q$ and $h \in \mathcal{A}$. Then, $p/h \sqsubseteq_{RS} q/h$.

PROOF. For proving Part (1), it is sufficient to establish that $\mathcal{R} =_{\mathrm{df}} \mathcal{R}_1 \cup \mathcal{R}_2$ with

$$\begin{aligned}
\mathcal{R}_1 =_{\mathrm{df}} & \{ \langle p/h, q/h \rangle \mid p \mathrel{\underset{\sim}{\sqsubset}_{\mathrm{RS}}} q,\ p \overset{h}{\not\longrightarrow} \text{ and } q \overset{h}{\not\longrightarrow} \} \\
\mathcal{R}_2 =_{\mathrm{df}} & \{ \langle \vec{p}/h, \vec{q}/h \rangle \mid \vec{p} = (p_1, p_2, \dots, p_n) \in \vec{P},\ \vec{q} = (q_1, q_2, \dots, q_n) \in \vec{Q}, \text{ and} \\
& \quad \vec{p} \mathrel{\hat{\underset{\sim}{\sqsubset}}_{\mathrm{RS}}} \vec{q} \}
\end{aligned}$$

is a stable ready simulation relation. We check the four conditions of Def. 11:

(RS1) This condition is straightforward for all pairs in $\mathcal{R}_1$ and $\mathcal{R}_2$.

(RS2) Let $\langle \hat{p}/h, \hat{q}/h \rangle \in \mathcal{R}$. Then, $\hat{p}/h \notin F$ and $\hat{p} \mathrel{\hat{\underset{\sim}{\sqsubset}}_{\mathrm{RS}}} \hat{q}$ implies $\hat{q}/h \in W_1'' \subseteq W''$. Hence, $\hat{q}/h \notin F$ by Lemma 31.

(RS3) Let $\langle p/h, q/h \rangle \in \mathcal{R}_1$. We distinguish two cases, $p/h \overset{a}{\Longrightarrow}| p'/h$ and $p/h \overset{a}{\Longrightarrow}| \vec{p}/h$:

If $p/h \overset{a}{\Longrightarrow}| p'/h$, then this computation does not contain a state $\vec{p}/h$ since such states are stable, i.e., $\vec{p}/h \overset{\epsilon}{\Longrightarrow}| p'/h$ is not possible. Hence, the first step of $p/h \overset{a}{\Longrightarrow}| p'/h$ arises from (H2) and the others from (H1), i.e., $p/h \overset{a}{\Longrightarrow}| p'/h$ due to $p \xrightarrow{a}_{\mathrm{F}} p_1 \xrightarrow{\tau}_{\mathrm{F}} p_2 \cdots \xrightarrow{\tau}_{\mathrm{F}} p_n = p'$. We have that p' is stable since $p' \xrightarrow{\tau}$ would imply $p'/h \xrightarrow{\tau}$ by (H1). Furthermore, $p' \overset{h}{\not\longrightarrow}$; otherwise, $p'/h \notin F$ would imply $\exists p''.\, p' \overset{\epsilon}{\Longrightarrow}| \,(\overset{h}{\Longrightarrow}|\,)^* p'' \overset{h}{\not\longrightarrow}$, whence $p' (\overset{h}{\Longrightarrow}|\,)^* p'' \overset{h}{\not\longrightarrow}$ by the stability of p' and thus $p'/h \xrightarrow{\tau}$ by (H3).

Since $p \underset{\sim}{\sqsubset}_{\mathrm{RS}} q$ and $p \overset{a}{\Longrightarrow}| p'$, we have $q \overset{a}{\Longrightarrow}| q'$ for some q' with $p' \underset{\sim}{\sqsubset}_{\mathrm{RS}} q'$. Assume $q \overset{a}{\Longrightarrow}| q'$ arises from $q \xrightarrow{a}_{\mathrm{F}} q_1 \xrightarrow{\tau}_{\mathrm{F}} q_2 \cdots \xrightarrow{\tau}_{\mathrm{F}} q_m = q'$. By (RS4), we get $q \overset{h}{\not\longrightarrow}$ from $p \overset{h}{\not\longrightarrow}$ and $q' \overset{h}{\not\longrightarrow}$ from $p' \overset{h}{\not\longrightarrow}$.

Further, $q \overset{h}{\not\longrightarrow}$ implies $q/h \xrightarrow{a} q_1/h \xrightarrow{\tau} q_2/h \cdots \xrightarrow{\tau} q_m/h = q'/h$. Since $p/h, p'/h \notin F$ by assumption, we have $q/h, q'/h \in W''_1 \subseteq W''$. In addition, $q_i/h \in W''_2 \subseteq W''$, for all $1 \le i \le m-1$. This gives $q/h \overset{a}{\Longrightarrow}_{\mathrm{F}} q'/h$.

Since q' is stable and $q' \overset{h}{\not\longrightarrow}$, Rules (H1) and (H3) are not applicable to q'/h, i.e., $q/h \overset{a}{\Longrightarrow}| q'/h$. This finishes the first case.

We now consider $p/h \overset{a}{\Longrightarrow}| \vec{p}/h$. This computation has the form $p/h \overset{a}{\Longrightarrow}_{\mathrm{F}} p'/h \xrightarrow{\tau}_{\mathrm{F}} \vec{p}/h$ for some p'. Since $p'/h \xrightarrow{\tau}_{\mathrm{F}} \vec{p}/h$ implies $p' \xrightarrow{h}$ and thus p' stable, we can repeat some of the argument of the first case to obtain some q' with $p' \hat{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q'$, $q \overset{a}{\Longrightarrow}| q'$ due to $q \xrightarrow{a}_{\mathrm{F}} q_1 \xrightarrow{\tau}_{\mathrm{F}} q_2 \cdots \xrightarrow{\tau}_{\mathrm{F}} q_m = q'$, $q/h \xrightarrow{a} q_1/h \xrightarrow{\tau} q_2/h \cdots \xrightarrow{\tau} q_m/h = q'/h$ and $q/h \notin F$.

Now, $p'/h \xrightarrow{\tau}_{\mathrm{F}} \vec{p}/h$ due to $p' (\overset{h}{\Longrightarrow}|\,)^* p'_1 \overset{h}{\Longrightarrow}| p'_2 \cdots \overset{h}{\Longrightarrow}| p'_n \overset{h}{\not\longrightarrow}$. This implies, by (RS3), that $q' (\overset{h}{\Longrightarrow}|\,)^* q'_1 \overset{h}{\Longrightarrow}| q'_2 \cdots \overset{h}{\Longrightarrow}| q'_n$, for some q'_i with $p'_i \underset{\sim}{\sqsubset}_{\mathrm{RS}} q'_i$, for $1 \le i \le n$, and $q'_n \overset{h}{\not\longrightarrow}$ due to (RS4). Since $p' \xrightarrow{h}$, we have $q' \xrightarrow{h}$ due to (RS4), i.e., $q'/h \xrightarrow{\tau} (q'_1, q'_2, \ldots, q'_n)/h = \vec{q}/h \overset{\tau}{\not\longrightarrow}$ with $\langle \vec{p}/h, \vec{q}/h \rangle \in \mathcal{R}_2$.

It remains for us to argue that $q_i/h \notin F$, for all $1 \le i \le m$, and $\vec{q}/h \notin F$. The latter follows from $\vec{q}/h \in W''_1 \subseteq W''$ due to $\vec{p} \hat{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} \vec{q}$ and $\vec{p}/h \notin F$. Further, $q'/h \notin F$ is a consequence of $q'/h \in W''_1 \subseteq W''$ due to $p' \hat{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q'$ and $p'/h \notin F$. Finally, $q_i/h \in W''_2 \subseteq W''$, for all $1 \le i < m$.

Next, we establish (RS3) for some pair $\langle \vec{p}/h, \vec{q}/h \rangle \in \mathcal{R}_2$ and distinguish again two cases: $\vec{p}/h \overset{a}{\Longrightarrow}| p'/h$ and $\vec{p}/h \overset{a}{\Longrightarrow}| \vec{p}'/h$.

In the former case, $\vec{p}/h \xrightarrow{a}_{\mathrm{F}} p/h \overset{\epsilon}{\Longrightarrow}| p'/h$ due to $p_j \xrightarrow{a}_{\mathrm{F}} p = p'_1 \xrightarrow{\tau}_{\mathrm{F}} p'_2 \cdots \xrightarrow{\tau}_{\mathrm{F}} p'_k = p'$ with $p' \overset{\tau}{\not\longrightarrow}$ and $p' \overset{h}{\not\longrightarrow}$. Since $p_j \underset{\sim}{\sqsubset}_{\mathrm{RS}} q_j$ by assumption and since $p_j \overset{a}{\Longrightarrow}| p'$, we have $q_j \overset{a}{\Longrightarrow}| q'$ for some q' with $p' \underset{\sim}{\sqsubset}_{\mathrm{RS}} q'$. Assume $q_j \overset{a}{\Longrightarrow}| q'$ due to $q_j \xrightarrow{a}_{\mathrm{F}} q'_1 \xrightarrow{\tau}_{\mathrm{F}} q'_2 \cdots \xrightarrow{\tau}_{\mathrm{F}} q'_m = q'$. Hence, we obtain, by (H4) and (H1), $\vec{q}/h \xrightarrow{a}_{\mathrm{F}} q'_1/h \overset{\epsilon}{\Longrightarrow}| q'/h$. To see this, observe that $q' \overset{\tau}{\not\longrightarrow}$ and $q' \overset{h}{\not\longrightarrow}$ by $p' \overset{h}{\not\longrightarrow}$ and (RS4), i.e., q'/h is stable. Furthermore, $q'/h \in W''_1 \subseteq W''$ due to p', and thus $q'_i/h \in W''_2 \subseteq W''$, for all $1 \le i < m$; similarly, $\vec{q}/h \in W''_1 \subseteq W''$ due to $\vec{p} \hat{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} \vec{q}$ by assumption.

Additionally, $p' \overset{h}{\not\longrightarrow}$, $q' \overset{h}{\not\longrightarrow}$ and $p' \underset{\sim}{\sqsubset}_{\mathrm{RS}} q'$ imply $\langle p'/h, q'/h \rangle \in \mathcal{R}_1$. This completes the reasoning for the former case.

In the latter case, $\vec{p}/h \overset{a}{\Longrightarrow}_{\mathrm{F}} p'/h \xrightarrow{\tau}_{\mathrm{F}} \vec{p}'/h$, again due to $p_j \xrightarrow{a}_{\mathrm{F}} p'_1 \xrightarrow{\tau}_{\mathrm{F}} p'_2 \cdots \xrightarrow{\tau}_{\mathrm{F}} p'_k = p'$ and $p'(\overset{h}{\Longrightarrow}\!|\,)^* p'_1 \overset{h}{\Longrightarrow}\!|\, p'_2 \cdots \overset{h}{\Longrightarrow}\!|\, p'_l \not\xrightarrow{h}$ with $\vec{p}' = (p'_1, p'_2, \ldots, p'_l)$ and $p' \not\xrightarrow{\;\;}$ (due to $p' \xrightarrow{h}$). Since $p_j \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q_j$ and $p_j \overset{a}{\Longrightarrow}\!|\, p'$, we have some q' with $q_j \overset{a}{\Longrightarrow}\!|\, q'$ and $p' \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q'$. Exploiting the latter we get $q'(\overset{h}{\Longrightarrow}\!|\,)^* q'_1 \overset{h}{\Longrightarrow}\!|\, q'_2 \cdots \overset{h}{\Longrightarrow}\!|\, q'_l \not\xrightarrow{h}$, with $p'_i \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q'_i$ for all $1 \le i \le l$, and $q' \xrightarrow{h}$. Now, $q'/h \in W''_1 \subseteq W''$ due to p', as well as $\vec{q}/h \in W''_1 \subseteq W''$ due to $\vec{p}$. The other processes on the computation $\vec{q}/h \overset{a}{\Longrightarrow} q'/h$ are in $W''_2 \subseteq W''$. Finally, $q'/h \xrightarrow{\tau}_{\mathrm{F}} (q'_1, q'_2, \ldots, q'_l)/h$ since $(q'_1, q'_2, \ldots, q'_l) \in W''_1 \subseteq W''$ due to $\vec{p}'$. We now conclude $\vec{q}/h \overset{a}{\Longrightarrow}\!|\, (q'_1, q'_2, \ldots, q'_l)/h$ and $\langle \vec{p}'/h, (q'_1, q'_2, \ldots, q'_l)/h \rangle \in \mathcal{R}_2$.

(RS4) Let $\langle p/h, q/h \rangle \in \mathcal{R}_1$. Because of (RS4) and $p/h \notin F$, whence $p \notin F_P$, we have $\mathcal{I}(p) = \mathcal{I}(q)$. Hence, $\mathcal{I}(p/h) = \mathcal{I}(p) = \mathcal{I}(q) = \mathcal{I}(q/h)$ by the operational rules for hiding. Next, let $\langle \vec{p}/h, \vec{q}/h \rangle \in \mathcal{R}_2$. Then, $\mathcal{I}(\vec{p}/h) = (\bigcup_{1 \le i \le n} \mathcal{I}(p_i)) \setminus \{h\} = (\bigcup_{1 \le i \le n} \mathcal{I}(q_i)) \setminus \{h\} = \mathcal{I}(\vec{q}/h)$. To verify the second equality, observe that $p_i \notin F_P$ (by the definition of $\vec{p}/h$), $p_i \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q_i$ and (RS4).

For establishing Part (2) of Thm. 32, consider first a computation $p/h \overset{\epsilon}{\Longrightarrow}\!|\, p'/h$; this is due to $p \overset{\epsilon}{\Longrightarrow}\!|\, p'$ with (H1) as above. By the assumption $p \sqsubseteq_{\mathrm{RS}} q$, we know of the existence of some q' with $q \overset{\epsilon}{\Longrightarrow}\!|\, q'$ and $p' \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q'$. Additionally, $p' \not\xrightarrow{h}$ must hold; otherwise, by $p'/h \notin F$, (H3) would be applicable, contradicting that p'/h is stable. Since $p' \notin F_P$ and by (RS4) we get $q' \not\xrightarrow{h}$, which implies that q'/h is stable. Furthermore, $q'/h \in W''_1 \subseteq W''$ and, for all other processes $\overline{q}$ along the computation $q \overset{\epsilon}{\Longrightarrow}\!|\, q'$, we have $\overline{q}/h \in W''_2 \subseteq W''$. Hence, $q/h \overset{\epsilon}{\Longrightarrow}\!|\, q'/h$ and $\langle p'/h, q'/h \rangle \in \mathcal{R}_1$ defined for Part (1), whence $p'/h \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q'/h$.

Second, we consider a computation $p/h \overset{\epsilon}{\Longrightarrow}\!|\, \vec{p}/h$, i.e., $p/h \overset{\epsilon}{\Longrightarrow}_{\mathrm{F}} p'/h \xrightarrow{\tau}_{\mathrm{F}} \vec{p}/h$ for some suitable p'. Hence, $p \overset{\epsilon}{\Longrightarrow}\!|\, p'(\overset{h}{\Longrightarrow}\!|\,)^* p_1 \overset{h}{\Longrightarrow}\!|\, p_2 \cdots \overset{h}{\Longrightarrow}\!|\, p_n \not\xrightarrow{h}$ and $p' \xrightarrow{h}$. Again, we have $q \overset{\epsilon}{\Longrightarrow}\!|\, q'$ for some q' with $p' \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q'$ and, by (RS4), $q' \xrightarrow{h}$. By (RS3), $q'(\overset{h}{\Longrightarrow}\!|\,)^* q_1 \overset{h}{\Longrightarrow}\!|\, q_2 \cdots \overset{h}{\Longrightarrow}\!|\, q_n$ with $p_i \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} q_i$ for all $1 \le i \le n$ and, by (RS4), $q_n \not\xrightarrow{h}$. Similarly as above, we conclude $q/h \overset{\epsilon}{\Longrightarrow}_{\mathrm{F}} q'/h$; note that $\vec{q}/h \in W''_1 \subseteq W''$ since $\vec{p} \mathrel{\hat{\underset{\sim}{\sqsubset}}}_{\mathrm{RS}} \vec{q}$ and $\vec{p}/h \notin F$. Thus, $q/h \overset{\epsilon}{\Longrightarrow}\!|\, \vec{q}/h$ and $\langle \vec{p}/h, \vec{q}/h \rangle \in \mathcal{R}_2$, whence also $\vec{p}/h \mathrel{\underset{\sim}{\sqsubset}}_{\mathrm{RS}} \vec{q}/h$. $\Box$

In summary, our framework of Logic LTS and ready simulation allows the semantically clean integration of standard logic operators, such as conjunction and disjunction, as well as standard process-algebraic operators, such as parallel composition, external choice and hiding. This testifies to the elegance and expressiveness of our setting for specifying and compositionally reasoning about mixed operational and logic systems.

6 Related work

This section briefly discusses related work; a full discussion can be found in [6].

Ready semantics. Our ready-tree semantics is in essence the *path-based possible-worlds semantics* of van Glabbeek [7] which goes back to Veglioni and De Nicola [8], and our ready simulation was first suggested by Bloom et al. [11]. However, in contrast to the standard notions of these semantics, our setting deals with internal actions (as does [15]) as well as inconsistencies.

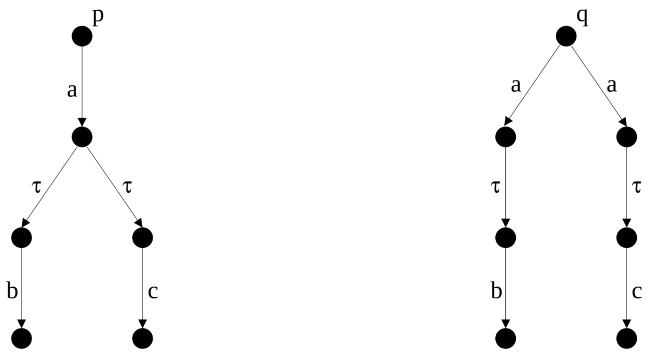

Fig. 11. Counter-example showing that the relation defined in Def. 33 is *properly* included in $\sqsubseteq_{\mathrm{RS}}$.

As an aside, we remark that another natural variant of adapting ready simulation to Logic LTS leads to a different, finer preorder. This variant uses a strong transition relation instead of a weak one in the premise of Cond. (RS3) in Def. 11 and does not require related processes to be stable:

Definition 33 (Finer ready simulation on Logic LTS)
Let $\langle P, \longrightarrow_P, F_P\rangle$ and $\langle Q, \longrightarrow_Q, F_Q\rangle$ be two Logic LTS. A relation $\mathcal{R} \subseteq P \times Q$ is a finer ready simulation relation*, if the following conditions hold, for any $\langle p, q\rangle \in \mathcal{R}$ and $a \in \mathcal{A}$:*

(RS2) $p \notin F_P \implies q \notin F_Q$;
(RS3a') $p \xrightarrow{a}_F p' \implies \exists q'.\, q \overset{a}{\Longrightarrow}_F q'$ *and* $\langle p', q'\rangle \in \mathcal{R}$;
(RS3b') $p \xrightarrow{\tau}_F p' \implies \exists q'.\, q \overset{\epsilon}{\Longrightarrow}_F q'$ *and* $\langle p', q'\rangle \in \mathcal{R}$;
(RS4') $p \notin F_P$ *and* p *stable* $\implies \exists q'.\, q \overset{\epsilon}{\Longrightarrow}| \, q'$, $\mathcal{I}(p) = \mathcal{I}(q')$ *and* $\langle p, q'\rangle \in \mathcal{R}$.

We say that p is finer ready simulated *by q, if there exists a finer ready simulation relation $\mathcal{R}$ with $\langle p, q\rangle \in \mathcal{R}$.*

This preorder coincides with Ulidowski's *lower refusal simulation* [16,17] which generalises Bloom et al.'s ready simulation by treating τ actions as unobservable, when adapted to Logic LTS in the obvious way [18]. However, although the preorders coincide, Def. 33 leads to more, smaller simulation relations. While it is easy to see that the above variant of ready simulation is contained in our ready simulation $\sqsubseteq_{\mathrm{RS}}$ of Def. 11, Fig. 11 demonstrates that this inclusion is proper. Here, $p \sqsubseteq_{\mathrm{RS}} q$, but p is not finer ready simulated by q.

Mixing styles of specification. Traditional research has often avoided explicitly mixing operational and logic styles of specification by translating one style into the other. For example, operational content may be translated into logic formulas, as is implicitly done in [19,20], where logic implication serves as refinement relation [21]. Dually, logic content may be translated into operational content. This is the case in automata-theoretic work, such as in Kurshan's work on ω-automata [22], which includes synchronous and asynchronous composition operators and uses maximal trace inclusion as refinement relation. However, both logic implication and trace inclusion are insensitive to deadlock and are thus inadequate in the presence of concurrency.

A seminal step towards a mixed setting was taken by Olderog in [12], where process-algebraic constructs are combined with *trace formulas*, and where failure semantics underlies refinement. In this approach, trace formulas can serve as processes, but not vice versa. Thus, and in contrast to our present work, [12] does not support the unrestricted mixing of operational and logic specification styles, which can be very useful as, e.g., demonstrated with our example in Sec. 4. The recent [23] mixes conjunction and synchronous product using some version of ready semantics in the context of implicit specification. This approach uses a similar concept of backward propagation in the definition of conjunction, but based on traces. Backward propagation of error/incompatibility information for trace-based semantics also appears in [24,25] in settings with input and output actions.

Finally, it should be noted that the term *consistency* as used here is different from the one in [13], where two specifications are called consistent if they have at least one common implementation. In our setting, this is trivially the case since $p \wedge q$ implements both p and q, for any p, q. Roughly speaking, then, p and q would be consistent in the sense of [13], if $p \wedge q \notin F$ in our setting.

7 Conclusions & future work

This article proved that ready simulation [11] is fully abstract with respect to conjunction and parallel composition on Logic LTS. In this sense, ready simulation is indeed a "logical" semantics. Establishing the full-abstraction result was non-trivial due to the challenges that arise when dealing with inconsistencies under conjunctive composition. This is evidenced by the complex compositionality proof with respect to conjunction, as well as the two-step "largest" precongruence proof that relied on our previous full-abstraction work on ready-tree semantics [6]. Our results show that conjunction is a tool for relating trace-based semantics to simulation-based semantics, via the concept of full abstraction. This sheds additional light onto van Glabbeek's linear time–branching time spectrum [7,15].

The insights gained from our results imply that ready simulation commends itself as a suitable behavioural relation for reasoning about specifications given in a mixed operational and logic style. Indeed, we showed that our framework can be extended by further logic operators, such as disjunction, and process-algebraic operators, such as external choice and hiding. Ready simulation is compositional for these operators, too, and satisfies all standard logic laws for conjunction and disjunction.

Future work shall proceed along two directions. Firstly, work is under way to integrate standard temporal logic operators, such as *"always"* and *"until"* operators, into our setting. This requires investigating which style of temporal logic is best suited, and extending Logic LTS with some sort of Büchi condition to be able to express liveness and fairness. Secondly, our setting should be rephrased in a term-based, process-algebraic style, so as to be able to explore axiomatisations of ready simulation in the presence of logic operators.

Acknowledgements

We thank Rob van Glabbeek and Irek Ulidowski for providing us with several pointers to the literature and for general discussions on ready simulation. We also thank the anonymous reviewers for their helpful comments.

References

[1] J. Bergstra, A. Ponse, S. Smolka, Handbook of Process Algebra, Elsevier, 2001.

[2] C. Hoare, Communicating Sequential Processes, Prentice Hall, 1985.

[3] R. Milner, Communication and Concurrency, Prentice Hall, 1989.

[4] E. Emerson, Temporal and modal logic, in: Handbook of Theoretical Computer Science, Vol. B, North-Holland, 1990, pp. 995–1072.

[5] G. Lüttgen, W. Vogler, Conjunction on processes: Full-abstraction via ready-tree semantics, in: FOSSACS 2006, Vol. 3921 of LNCS, Springer, 2006, pp. 261–276.

[6] G. Lüttgen, W. Vogler, Conjunction on processes: Full-abstraction via ready-tree semantics, TCS 373 (1–2) (2007) 19–40.

[7] R. Glabbeek, The linear time – branching time spectrum I, in: Handbook of Process Algebra, Elsevier, 2001, Ch. 1, pp. 3–99.

[8] S. Veglioni, R. De Nicola, Possible worlds for process algebras, in: CONCUR '98, Vol. 1466 of LNCS, Springer, 1998, pp. 179–193.

[9] S. Brookes, C. Hoare, A. Roscoe, A theory of communicating sequential processes, J. ACM 31 (3) (1984) 560–599.

[10] T. Bolognesi, E. Brinksma, Introduction to the ISO specification language LOTOS, Computer Networks 14 (1987) 25–59.

[11] B. Bloom, S. Istrail, A. Meyer, Bisimulation can't be traced, J. ACM 42 (1) (1995) 232–268.

[12] E.-R. Olderog, Nets, Terms and Formulas, Cambridge Tracts in Theoretical Computer Science 23, Cambridge Univ. Press, 1991.

[13] M. Steen, J. Derrick, E. Boiten, H. Bowman, Consistency of partial process specifications, in: AMAST '98, Vol. 1548 of LNCS, Springer, 1999, pp. 248–262.

[14] E.-R. Olderog, C. Hoare, Specification-oriented semantics for communicating processes, Acta Informatica 23 (1) (1986) 9–66.

[15] R. Glabbeek, The linear time – branching time spectrum II, available at http://theory.stanford.edu/˜rvg/abstracts.html#26 (1993).

[16] I. Ulidowski, Equivalences on observable processes, in: LICS '92, IEEE Computer Society Press, 1992, pp. 148–159.

[17] I. Ulidowski, Refusal simulation and interactive games, in: AMAST 2002, Vol. 2422 of LNCS, Springer, 2002, pp. 208–222.

[18] I. Ulidowski, G. Lüttgen, W. Vogler, Personal communication (November & December 2007).

[19] S. Graf, J. Sifakis, A logic for the description of non-deterministic programs and their properties, Inform. & Control 68 (1–3) (1986) 254–270.

[20] L. Lamport, The temporal logic of actions, TOPLAS 16 (3) (1994) 872–923.

[21] M. Abadi, G. Plotkin, A logical view of composition, TCS 114 (1) (1993) 3–30.

[22] R. Kurshan, Computer-Aided Verification of Coordinating Processes: The Automata-Theoretic Approach, Princeton Univ. Press, 1994.

[23] J.-B. Raclet, Residual for component specifications, in: Formal Aspects of Component Software, Vol. 215 of ENTCS, Elsevier, 2008, pp. 93–110.

[24] D. Dill, Trace Theory for Automatic Hierarchical Verification of Speed-Independent Circuits, MIT Press, 1988.

[25] L. de Alfaro, T. Henzinger, Interface automata, in: FSE 2001, ACM Press, 2001, pp. 109–120.