Appendix (Supplementary Material)

A Properties of Composition: Proofs for Section 2.2

The composition operation of [31] is defined for a pair of modules only. Here, we formally show that we can freely compose the modules. The proofs are conceptually simple but technically involved; we present them for scrutiny here.

In order to identify modules with the same structure, we first introduce the notion of isomorphism.

Definition 8 (Isomorphism of modules).
Let $M^{(1)} = (X^{(1)}, I^{(1)}, Q^{(1)}, T^{(1)}, \lambda^{(1)}, q_0^{(1)})$ *and* $M^{(2)} = (X^{(2)}, I^{(2)}, Q^{(2)}, T^{(2)}, \lambda^{(2)}, q_0^{(2)})$. *We say that* $M^{(1)}$ *is* TS-isomorphic *with* $M^{(2)}$ *if:*

- *there exist three bijections* $\xi_X : X^{(1)} \to X^{(2)}$, $\xi_I : I^{(1)} \to I^{(2)}$ *and* $\xi : Q^{(1)} \to Q^{(2)}$*;*
- $(\lambda^{(2)}(\xi(q)))(\xi_X(x)) = (\lambda^{(1)}(q))(x)$*;*
- $T^{(2)} = \{(\xi(q_1), K, \xi(q_2)) : (q_1, J, q_2) \in T^{(1)}; J(x) = K(\xi_I(x))\}$*;*
- $\xi(q_0^{(1)}) = q_0^{(2)}$.

We denote it by $M^{(1)} \equiv_{\xi_X,\xi_I,\xi} M^{(2)}$.

If both ξ_X *and* ξ_I *are identities, we call* $M^{(1)}$ *and* $M^{(2)}$ isomorphic *and denote it by* $M^{(1)} \equiv_\xi M^{(2)}$*, or simply* $M^{(1)} \equiv M^{(2)}$.

Now we can prove the commutativity and associativity of the composition rule in Definition 3.

Proposition 2 (Commutativity). *Let* $M^{(1)}$ *and* $M^{(2)}$ *be a pair of asynchronous modules.*

Then $M^{(1)}|M^{(2)} \equiv M^{(2)}|M^{(1)}$.

Proof. Let $M = M^{(1)}|M^{(2)} = (X, I, Q, T, \lambda, q_0)$ and $M' = M^{(2)}|M^{(1)} = (X', I', Q', T', \lambda', q_0')$.

First note that $X = X^{(1)} \uplus X^{(2)} = X^{(2)} \uplus X^{(1)} = X'$ and $I = (I^{(1)} \cup I^{(2)}) \setminus (X^{(1)} \uplus X^{(2)}) = (I^{(2)} \cup I^{(1)}) \setminus (X^{(2)} \uplus X^{(1)}) = I'$.

Let $\xi : Q \to Q'$ be the mapping between sets of states defined as $\xi(q^{(1)}, q^{(2)}) = (q^{(2)}, q^{(1)})$. It is easy to see that ξ is a bijection between Q and Q', and $\xi(q_0) = \xi(q_0^{(1)}, q_0^{(2)}) = (q_0^{(2)}, q_0^{(1)}) = q_0'$. It remains to check that $(q, \alpha, q') \in T$ if and only if $(\xi(q), \alpha, \xi(q')) \in T'$.

Let $(q, \alpha, q') \in T$ be an effect of using rule $\mathbf{SYN}$ to transitions $(q^{(1)}, \alpha^{(1)}, {q'}^{(1)})$ and $(q^{(2)}, \alpha^{(2)}, {q'}^{(2)})$. Then we can apply rule $\mathbf{SYN}$ to those transitions, changing their order, and obtain $(\xi(q), \alpha, \xi(q')) \in T'$. Similarly, for any $(q, \alpha, q') \in T'$ that was obtained by using rule $\mathbf{SYN}$ we have a transition $(\xi^{-1}(q), \alpha, \xi^{-1}(q')) \in T$.

Let $(q, \alpha, q') \in T$ be an effect of using rule $\mathbf{ASYN_L}$ (or $\mathbf{ASYN_R}$) to transitions $(q^{(1)}, \alpha^{(1)}, {q'}^{(1)})$ and $(q^{(2)}, \alpha^{(2)}, {q'}^{(2)})$. Then we can apply rule $\mathbf{ASYN_R}$ (rule $\mathbf{ASYN_L}$, respectively) to those transitions, changing their order, and obtain $(\xi(q), \alpha, \xi(q')) \in T'$. Similarly, for any $(q, \alpha, q') \in T'$ that was obtained by using rule $\mathbf{ASYN_L}$ or rule $\mathbf{ASYN_R}$ we have a corresponding transition $(\xi^{-1}(q), \alpha, \xi^{-1}(q')) \in T$.

Proposition 3 (Associativity). *Let $M^{(1)}$, $M^{(2)}$ and $M^{(3)}$ be pairwise asynchronous modules.*

Then $(M^{(1)}|M^{(2)})|M^{(3)} \equiv M^{(1)}|(M^{(2)}|M^{(3)})$.

Proof. Let $M = (M^{(1)}|M^{(2)})|M^{(3)} = (X, I, Q, T, \lambda, q_0)$ and $M' = M^{(1)}|(M^{(2)}|M^{(3)}) = (X', I', Q', T', \lambda', q_0')$.

First note that $X = (X^{(1)} \uplus X^{(2)}) \uplus X^{(3)} = X^{(1)} \uplus (X^{(2)} \uplus X^{(3)}) = X'$.

Moreover, since both $X^{(1)} \uplus X^{(2)}$ and $X^{(2)} \uplus X^{(3)}$ are subsets of $X^{(1)} \uplus X^{(2)} \uplus X^{(3)}$, we have

$$\begin{aligned} I &= (((I^{(1)} \cup I^{(2)}) \setminus (X^{(1)} \uplus X^{(2)})) \cup I^{(3)}) \setminus (X^{(1)} \uplus X^{(2)} \uplus X^{(3)}) \\ &= (I^{(1)} \cup I^{(2)} \cup I^{(3)}) \setminus (X^{(1)} \uplus X^{(2)} \uplus X^{(3)}) \\ &= (I^{(1)} \cup ((I^{(2)} \cup I^{(3)}) \setminus (X^{(2)} \uplus X^{(3)})) \setminus (X^{(1)} \uplus X^{(2)} \uplus X^{(3)}) = I'. \end{aligned}$$

Let $\xi : Q \to Q'$ be the mapping between sets of states defined as $\xi((q^{(1)}, q^{(2)}), q^{(3)}) = (q^{(1)}, (q^{(2)}, q^{(3)}))$. It is easy to see that ξ is a bijection between Q and Q', and $\xi(q_0) = \xi((q_0^{(1)}, q_0^{(2)}), q_0^{(3)}) = (q_0^{(1)}, (q_0^{(2)}, q_0^{(3)})) = q_0'$. It remains to check that $(q_1, \alpha, q_2) \in T$ if and only if $(\xi(q_1), \alpha, \xi(q_2)) \in T'$.

Note that any transition in M is the effect of the composition of two rules from the set $\{\mathbf{ASYN_L}, \mathbf{ASYN_R}, \mathbf{SYN}\}$, The first rule combines compatible transitions from $M^{(1)}$ and $M^{(2)}$, while the second one – transitions from $M^{(1)}|M^{(2)}$ and $M^{(3)}$. For any triple of transitions $((t^{(1)}, t^{(2)}), t^{(3)}) \in (T^{(1)} \times T^{(2)}) \times T^{(3)}$ which are pairwise compatible (as provided in all rules presumptions), we would get 9 results to be added to T. And there will be no other transitions in T (since any incompatibility would effect that at least one rule is not applicable).

Moreover, any transition in M' is the effect of a similar composition (with 9 transitions in T' for each pairwise compatible triple of transitions).

We show that any pairwise compatible triple of transitions $((q^{(1)}, \alpha^{(1)}, {q'}^{(1)}), ((q^{(2)}, \alpha^{(2)}, {q'}^{(2)}), ((q^{(3)}, \alpha^{(3)}, {q'}^{(3)})$ (i.e. not only $(\lambda(q^{(1)}) \cup \lambda(q^{(2)})) \sim \alpha^{(3)}$, $(\lambda(q^{(2)}) \cup \lambda(q^{(3)})) \sim \alpha^{(1)}$ and $(\lambda(q^{(3)}) \cup \lambda(q^{(1)})) \sim \alpha^{(2)}$ but also $\alpha^{(1)} \sim \alpha^{(2)}$, $\alpha^{(2)} \sim \alpha^{(3)}$ and $\alpha^{(3)} \sim \alpha^{(1)}$), would give a set of seven different effects of their execution $q' \in \{((x^{(1)}, x^{(2)}), x^{(3)})\}$ where $x^{(i)}$ is either $q^{(i)}$ or ${q'}^{(i)}$, but the second option is used at least once (i.e. we exclude $((q^{(1)}, q^{(2)}), q^{(3)})$ since it cannot be a result of any presented rule). Similarly, we would obtain the corresponding set of transitions $(\xi(q), \alpha, \xi(q')) \in T'$.

Indeed to obtain (q, α, q') or $(\xi(q), \alpha, \xi(q'))$ we need to combine transitions $(q^{(1)}, \alpha^{(1)}, {q'}^{(1)})$, $(q^{(2)}, \alpha^{(2)}, {q'}^{(2)})$ and $(q^{(3)}, \alpha^{(1)}, {q'}^{(3)})$ using the following rules:

- $q' = ({q'}^{(1)}, ({q'}^{(2)}, {q'}^{(3)}))$
  $\mathbf{SYN}$ composing $M^{(1)}$ with $M^{(2)}$ and $\mathbf{SYN}$ composing $(M^{(1)}|M^{(2)})$ with $M^{(3)}$
  or
  $\mathbf{SYN}$ composing $M^{(2)}$ with $M^{(3)}$ and $\mathbf{SYN}$ composing $M^{(1)}$ with $(M^{(2)}|M^{(3)})$

- $q' = ({q'}^{(1)}, {q'}^{(2)}, q^{(3)})$
  $\mathbf{SYN}$ composing $M^{(1)}$ with $M^{(2)}$ and $\mathbf{ASYN_L}$ composing $(M^{(1)}|M^{(2)})$ with $M^{(3)}$

or
$\mathbf{ASYN_L}$ composing $M^{(2)}$ with $M^{(3)}$ and $\mathbf{SYN}$ composing $M^{(1)}$ with $(M^{(2)}|M^{(3)})$

- $q' = ({q'}^{(1)}, q^{(2)}, {q'}^{(3)})$
  $\mathbf{ASYN_L}$ composing $M^{(1)}$ with $M^{(2)}$ and $\mathbf{SYN}$ composing $(M^{(1)}|M^{(2)})$ with $M^{(3)}$
  or
  $\mathbf{SYN}$ composing $M^{(2)}$ with $M^{(3)}$ and $\mathbf{ASYN_R}$ composing $M^{(1)}$ with $(M^{(2)}|M^{(3)})$

- $q' = (q^{(1)}, {q'}^{(2)}, {q'}^{(3)})$
  $\mathbf{ASYN_R}$ composing $M^{(1)}$ with $M^{(2)}$ and $\mathbf{SYN}$ composing $(M^{(1)}|M^{(2)})$ with $M^{(3)}$
  or
  $\mathbf{SYN}$ composing $M^{(2)}$ with $M^{(3)}$ and $\mathbf{ASYN_R}$ composing $M^{(1)}$ with $(M^{(2)}|M^{(3)})$

- $q' = (q^{(1)}, {q'}^{(2)}, q^{(3)})$
  $\mathbf{ASYN_R}$ composing $M^{(1)}$ with $M^{(2)}$ and $\mathbf{ASYN_L}$ composing $(M^{(1)}|M^{(2)})$ with $M^{(3)}$
  or
  $\mathbf{ASYN_L}$ composing $M^{(2)}$ with $M^{(3)}$ and $\mathbf{ASYN_R}$ composing $M^{(1)}$ with $(M^{(2)}|M^{(3)})$

- $q' = ({q'}^{(1)}, q^{(2)}, q^{(3)})$
  $\mathbf{ASYN_L}$ composing $M^{(1)}$ with $M^{(2)}$ and $\mathbf{ASYN_L}$ composing $(M^{(1)}|M^{(2)})$ with $M^{(3)}$
  or
  any rule composing $M^{(2)}$ with $M^{(3)}$ and $\mathbf{ASYN_L}$ composing $M^{(1)}$ with $(M^{(2)}|M^{(3)})$

- $q' = (q^{(1)}, q^{(2)}, {q'}^{(3)})$
  any rule composing $M^{(1)}$ with $M^{(2)}$ and $\mathbf{ASYN_R}$ composing $(M^{(1)}|M^{(2)})$ with $M^{(3)}$
  or
  $\mathbf{ASYN_R}$ composing $M^{(2)}$ with $M^{(3)}$ and $\mathbf{ASYN_R}$ composing $M^{(1)}$ with $(M^{(2)}|M^{(3)})$

Note that, directly by Definition 3 (minimality of T under the set of rules), there is no other transition to be considered. Also the final pruning is not a problem, since self-loops in both considered compositions are pruned in the same way. Hence we can conclude the proof.

B Strategic Properties of Curtailment

Lemma 1. *Let M be a closed system with a set of variables X. Moreover, $w = w_1 w_2 \dots$ is an infinite sequence over X, and $Y \subseteq X$. Let ϕ be an* **LTL** *(without next operator) formula which satisfies $var(\phi) \subseteq Y$ and $v = w_1|_Y\, w_2|_Y\, \dots$ be a special curtailment of w. Then $M, w \models \phi$ if and only if $M, v \models \phi$.*

Proof. The proof is by the induction on the length of formula ϕ (considering arbitrary values of w).

Case 1: $M, w \models p(Z)$, $Z \subseteq Y$

$M, w \models p(Z)$ iff $w_1|_Z = p(Z)$ iff $(w_1|_Y)|_Z = p(Z)$ iff $M, v \models p(Z)$.

Case 2: $M, w \models \neg\phi$, $M, w \models \phi$ iff $M, v \models \phi$

$M, w \models \neg\phi$ iff $M, w \not\models \phi$ iff $M, v \not\models \phi$ iff $M, v \models \neg\phi$.

Case 3: $M, w \models \phi_1 \wedge \phi_2$, $M, w \models \phi_1$ iff $M, v \models \phi_1$, $M, w \models \phi_2$ iff $M, v \models \phi_2$

$M, w \models \phi_1 \wedge \phi_2$ iff $M, w \models \phi_1$ and $M, w \models \phi_2$

iff $M, v \models \phi_1$ and $M, v \models \phi_2$ iff $M, v \models \phi_1 \wedge \phi_2$

Case 4: $M, w \models \phi_1 \,\mathrm{U}\, \phi_2$, $M, w[k, \infty] \models \phi$ iff $M, v[k, \infty] \models \phi$ for ϕ shorter than $\phi_1 \,\mathrm{U}\, \phi_2$,

$M, w \models \phi_1 \,\mathrm{U}\, \phi_2$

iff there exists j such that $M, w[i, \infty] \models \phi_1$ for $0 < i < j$ and $M, w[j, \infty] \models \phi_2$

iff there exists j such that $M, v[i, \infty] \models \phi_1$ for $0 < i < j$ and $M, v[j, \infty] \models \phi_2$

iff $M, v \models \phi_1 \,\mathrm{U}\, \phi_2$.

Lemma 2. *Let M be a closed system with a set of variables X, $w = w_1 w_2 \ldots$ is an infinite sequence over X, and $Y \subseteq X$. Let ϕ be an* **LTL** *(without next operator) formula which satisfies $var(\phi) \subseteq Y$ and $v = w_1 \ldots w_{k-1} w_k w_{k+2} \ldots$ where $w_k = w_{k+1}$. Then $M, w \models \phi$ if and only if $M, v \models \phi$.*

Proof. Similar to the proof of Lemma 1. The only interesting case is related to the operator **until**, where we need to skip $w[k+1]$ under one of the universal quantifiers and add one to j if j is larger than k.

Lemma 3. *Let M be a closed system with a set of variables X, w is an infinite sequence over X, and $Y \subseteq X$. Let ϕ be an* **LTL** *(without next operator) formula which satisfies $var(\phi) \subseteq Y$. Then $M, w \models \phi$ if and only if for every curtailment $w|_Y$ we have $M, w|_Y \models \phi$.*

Proof. By Lemma 1 we can assume that $X = Y$. Then, we can proceed as in the case of the proof of Lemma 2. Again, the only interesting case is the one related to the operator **until**.

Let us consider this case in details. We have $w = w_1 w_2 \ldots$ and an infinite sequence $1 = j_1 < j_2 < \ldots$ such that $v = v_1 v_2 \ldots$ and $\forall_i \forall_{j_i \leq k < j_{i+1}} v_k = w_k$. Let $M, w \models \phi_1 \,\mathrm{U}\, \phi_2$ and (by the induction hypothesis) $M, [k, \infty] \models \phi$ iff $M, v[i, \infty] \models \phi$ for any ϕ shorter than $\phi_1 \,\mathrm{U}\, \phi_2$ and $j_i \leq k < j_{i+1}$.

Then $M, w \models \phi_1 \,\mathrm{U}\, \phi_2$

iff there exists k such that $M, w[i, \infty] \models \phi_1$ for $0 < i < k$ and $M, w[k, \infty] \models \phi_2$

iff there exists k' such that $j_{k'} \leq k < j_{k'+1}$ $M, v[i, \infty] \models \phi_1$ for $0 < i < k'$ and $M, w[k', \infty] \models \phi_2$.

iff $M, v \models \phi_1 \,\mathrm{U}\, \phi_2$.

Hence we get that $M, w \models \phi$ if and only if $M, v \models \phi$, which ends the proof.

C Benchmark: Robots in a Factory

Factory. The factory is defined as a vertical alignment of consecutive places that can be easily traversed by the robots. We assume that each location is big enough to fit all robots and packages at the same time. The first, top cell in the factory is considered as a production line and at the beginning it holds one package for each robot. The last, bottom place is a storage area where the packages should be delivered.

Packages. Packages are boxes containing items produced by the factory production line. Each package is small enough to be carried by a robot, but too big for the robots to be able to carry two packages at once. Boxes are labeled with NFC tags that are used to address the package to the specified robot.

Robots. First type of agent in our scenario is a robot. Robots can move freely around the factory and interact with packages. They are equipped with NFC readers that allows them to read the package tags and rewrite them with the new data. We define each robot using several variables: the current position in the factory, the energy level, the package status, and the current operation mode. Each agent has the following actions available: move one field up/down, pick package at your current position or drop package for the next robot. Note that robot can pick only packages that are addressed to him and can only address his package for the the robot with higher id. The package can be placed only when the destination location is not already holding the package for the specified robot. Robots have limited batteries and execution of the move action requires energy. When the energy is depleted, the robot stuck in his current position, unable to move but still able to pick and drop the packages. We assume that the factory is big enough to contain all robots, so they will not collide when going through already occupied positions.

Location. To maintain current status of the factory and all locations in the first set of experiments, we define each place as a separate agent. Those agents contain information about currently present packages and their addresses (robots IDS). The location agent can interact with robots by recording which package was picked up and which was dropped. It does so by changing its internal variables to reflect the status of the available packages at the location. In the case of the second experiment, agents representing different locations are joined into one Factory agent. This new agent represents the factory and all its locations. Apart from the previous functionalities, the factory can also detect when all robots are out of energy and loop infinitely in this state which results in getting only infinite paths in the resulting global model.

Interaction between agents. Due to the lack of synchronization in the underlying syntax, the interaction between robots and locations, i.e., picking up packages and placing them, must be carefully designed. Note that allowing robots to freely move packages would not guarantee that the locations correctly update their statuses to reflect the packages positions. For that reason, we have designed a custom handshake protocol.

Each location is equipped with a marker that acts as a semaphore, like in the train scenario. When robot wants to pick up the package at his current location, he checks if the package is available and if so, he can change his operation mode to "pick". After noticing the robot waiting for his package, the location opens the semaphore for the specified robot. After that, the robot can pick up the package, changing his internal variables, but remaining in the "pick" mode. When the package is picked up, the location reflects that in its status and changes semaphore back to it default position. The robot can go back to his default "idle" mode only after the semaphore is no longer open for him. Thanks to this, we can abstract from the actions of the robots outside the considered coalition. Instead of the usual fairness constraints we took advantage of the fact that they cannot operate indefinitely due to energy limitations and the package redirection protocol.

Similar procedure is defined for placing packages.

Indistinguishability relations. To properly operate in the factory each robot knows his current position, energy level, package status and operation mode, but not that of other robots. He can also see all locations and their current statuses.

Configuration. The model is configured with three variables: the number of robots (R), the number of locations in the factory (F), and the initial energy for each robot (E).

Technical conditions. We have implemented the assume-guarantee verification and the verification algorithms in Python. The sources are available at github.com/agrprima22/agr. The experiments were run on a server equipped with two 2.80 GHz Intel Xeon Gold 6242 CPUs and 16 GB RAM, running 64-bit Linux. All times are given in seconds. The timeout was set to 2 hours.