

Invariant Transitions They are of the form $p \xrightarrow{\text{inv}} fs$, where $p \in \mathcal{C}$ is a composition, and $fs \in (\mathcal{L} \rightarrow \mathcal{P})^*$ is a list containing the invariant function associated to each automaton in p .

The reader may have expected initialization or invariant transitions of the form:

$$vs \xrightarrow{\text{inv}} p$$

where vs is a list of locations, and p is a predicate. However this approach requires enumerating the state space explicitly to construct the $\xrightarrow{\text{inv}}$ relation. By using lists of functions we avoid this explicit construction.

Wild-card Transitions They are of the form $p \xrightarrow{\#} xs$, where $p \in \mathcal{C}$ is a composition, and $xs \in (-)^*$ is a sequence of wild-cards whose size coincides with the number of automata that are composed in parallel in p . These transition are not needed for reconstructing the environment transitions, they are used in the linear SOS rules to model the fact that nothing changes in a component of a parallel composition, when the other component performs an action.

In Table 3 we show some of the linear SOS rules for CIF compositions. We have omitted the rules for synchronizing actions, initialization, and wild-card transitions since they are similar to the invariant transitions.

The linear rules can be easily to obtained from the symbolic ones. For action rules, invariants and initialization predicates, and the synchronizing action label are simply omitted (since they can be obtained from other transitions). The linear rule for interleaving parallel composition is almost identical to the symbolic rule. The only differences are that the set A is obtained from a $\xrightarrow{\text{sync}}$ transition, and we use the wild-card transition to represent the fact that the locations of the other automaton are not relevant (at the symbolic level at least). A similar observation can be made for the rule for parallel composition. In this case since we do not have the synchronizing label, we reconstruct it from the $\xrightarrow{\text{sync}}$ transition. This label is equivalent to $a \in A$, thus a label true in both components is equivalent to $a \in A_p \wedge a \in A_q$, which is in turn equivalent to $a \in A_p \cap A_q$.

If a composition p contains no synchronizing actions, then the size of its induced transition system is linear w.r.t. the size of p . However, the size of the LiTS also depends on the number of synchronizing actions. The following property gives the formal details.

Property 1 (Size of the linear transition system) *Let p be a CIF composition, such that it contains n automata $\alpha_i \equiv (V_i, \text{init}_i, \text{inv}_i, E, \text{act}_{S_i})$, $0 \leq i < n$. Let a be the only synchronizing action in these automata. Then the number of transitions in the LiTS associated to p is given by:*

$$\bullet \sum_{0 \leq i < n} \#\{x \mid (v, g, x, u, v') \in E_i \wedge x \neq a\} + \div \sum_{0 \leq i < n} \#\{x \mid (v, g, x, u, v') \in E_i \wedge x = a\} \quad (3)$$

where $\#A$ is the number of elements in set A .

In spite of the fact that the number in (3) can be significantly large, in practice, communication among components is usually restricted to a few automata, and the number of edges of an automaton that contain a given synchronizing action a is small.

3.1 Relating LiTS and STS

In the same way symbolic transitions are related to explicit ones via soundness and completeness results, linear transitions have the same property w.r.t. symbolic transitions.