Parallel Composition in a Paper by Jensen, Larsen & Skou is not Associative

Jasper Berendsen Frits Vaandrager

5th September 2007

In [JLS00] Jensen, Larsen & Skou: 1) claim to have constructed a framework that captures an important part of the semantics for networks of timed automata as used in the Uppaal model checker. 2) show a simulation relation within the framework such that making an abstraction of a single parallel component leads to an abstraction of the whole system. We will show that their claim (1) contains a essential flaw, because parallel composition in their framework is not associative.

We use all definitions, lemma's, and theorems from [JLS00]. Consider in total only one variable u that can have only the value 0 or 1. The set of variables becomes $V = \{u\}$. Assume we are working with semantics of timed automata that do not use clocks. We can simply draw timed transition systems, without having to worry about denseness of time.

Assume three timed transition systems $\mathcal{T}_A$, $\mathcal{T}_B$, and $\mathcal{T}_C$. TTS $\mathcal{T}_A$ is able to write variable u, and $W_A = \{u\}$. The drawing for $\mathcal{T}_A$ is:

$$\langle p_A, u \mapsto 0\rangle \xrightarrow{a} \langle p'_A, u \mapsto 1\rangle$$

$\mathcal{T}_B$ is *not* able to write variable u, and $W_B = \emptyset$. The drawing for $\mathcal{T}_B$ is:

$$\langle p_B, u \mapsto 0\rangle \xrightarrow{\bar{a}} \langle p'_B, u \mapsto 0\rangle$$

$\mathcal{T}_C$ is able to write variable u, and $W_C = \{u\}$. The drawing for $\mathcal{T}_C$ is:

$$\langle p_C, u \mapsto 0\rangle$$

We will show that $(\mathcal{T}_A \| \mathcal{T}_B) \| \mathcal{T}_C \neq \mathcal{T}_A \| (\mathcal{T}_B \| \mathcal{T}_C)$, where $=$ means equivalent up to isomorphism.

In $(\mathcal{T}_A \| \mathcal{T}_B) \| \mathcal{T}_C$ we have transition: $\langle((p_A, p_B), p_C), u \mapsto 0\rangle \xrightarrow{\tau} \langle((p'_A, p'_B), p_C), u \mapsto 1\rangle$. We get this transition by first combining the transition of $\mathcal{T}_A$ with the transition of $\mathcal{T}_B$ via rule 4 of parallel composition. Thus we get: $\langle(p_A, p_B), u \mapsto 0\rangle \xrightarrow{\tau} \langle(p'_A, p'_B), u \mapsto 1\rangle$. Finally combine this transition with the single state of $\mathcal{T}_C$ via rule 2 of parallel composition.

However, $\langle(p_A, (p_B, p_C)), u \mapsto 0\rangle \xrightarrow{\tau} \langle(p'_A, (p'_B, p_C)), u \mapsto 1\rangle$ is *not* a transition in $\mathcal{T}_A \| (\mathcal{T}_B \| \mathcal{T}_C)$. To see this, first make the system $\mathcal{T}_B \| \mathcal{T}_C$. By using rule 2 we have: $\langle(p_B, p_C), u \mapsto 0\rangle \xrightarrow{\bar{a}} \langle(p_B, p_C), u \mapsto 0\rangle$. The variables writen by $\mathcal{T}_B \| \mathcal{T}_C$ are $W_B \cup W_C = \{u\}$ according to composition of signatures.

But now we are unable to apply rule 4 to get the τ transition, because $\mathcal{T}_B \| \mathcal{T}_C$ 'writes' 0 to u, while $\mathcal{T}_A$ 'writes' 1 to u, see conditions of rule 4.

References

[JLS00] Henrik Ejersbo Jensen, Kim Guldstrand Larsen, and Arne Skou. Scaling up uppaal automatic verification of real-time systems using compositionality and abstraction. In Mathai Joseph, editor, *FTRTFT*, volume 1926 of *Lecture Notes in Computer Science*, pages 19–30. Springer, 2000.