Erratum to the paper *Untimed Language Preservation in Timed Systems[1]*

Ocan Sankur

September 28, 2011

The proof of Corollary 1, stating the 2EXPTIME complexity for the algorithm for general timed automata is incorrect. Below is the corrected version of the corollary, where we give an EXPSPACE algorithm. The statement and the proofs are correct in the concise case (PSPACE).

Corollary 1 (Corrected) *For concise timed automata with progress cycles, language robustness can be decided in* PSPACE*. For general timed automata with progress cycles, language robustness can be decided in* EXPSPACE*.*

Proof. The proof of the paper is correct for the concise case.

For general timed automata, we describe a non-deterministic exponential space algorithm to decide $L(\mathcal{R}(\mathcal{A}_{\frac{1}{N_0}})) \not\subseteq L(\mathcal{R}(\mathcal{A}))$. Observe that $\mathcal{R}(\mathcal{A})$ can be complemented using the subset construction, and that each state in the complemented automaton has exponential size (since there are exponentially many regions). Let us call the deterministic complement automaton $\mathcal{R}(\mathcal{A})^c$. The algorithm consists in guessing a path in $\mathcal{R}(\mathcal{A}_{\frac{1}{N_0}})$ while, in parallel, simulating the path in $\mathcal{R}(\mathcal{A})^c$. This can be done in exponential space since a state of $\mathcal{R}(\mathcal{A}_{\frac{1}{N_0}})$ can be represented in exponential space. The algorithm accepts if the simulating set becomes empty, and otherwise rejects after a doubly exponential number of steps. In fact, $L(\mathcal{R}(\mathcal{A}_{\frac{1}{N_0}})) \not\subseteq L(\mathcal{R}(\mathcal{A}))$ is equivalent to $L(\mathcal{R}(\mathcal{A}_{\frac{1}{N_0}})) \cap L(\mathcal{R}(\mathcal{A})^c) \neq \emptyset$, and in this case, the intersection contains a word of size at most doubly exponential, since the product automaton has this size.

References

[1] Ocan Sankur, *Untimed Language Preservation in Timed Systems.* In MFCS'11, LNCS 6907, pages 556-567. Springer, 2011.