From well structured transition systems to program verification (invited paper)

Alain Finkel
LSV, ENS Paris-Saclay
CNRS, Université Paris-Saclay, France, IUF
finkel@lsv.fr

Abstract: We describe the use of the theory of WSTS for verifying programs.

Well structured transition systems (WSTS) were introduced in [2] and surveys can be found in [4, 1]. WSTS are ordered transition systems $(S, \to, \leq)$ enjoying two properties: (1) the ordering $\leq$ is well on S and (2) $\to$ is monotone with respect to $\leq$. Many classical properties are decidable for WSTS as termination, control-state reachability, coverability, boundedness,...etc. Given a program P and a safety property ϕ, let's describe two steps for verifying that P satisfies ϕ by using WSTS:

1. The first step is to build a transition system $(S, \to)$ associated with (P, ϕ) and in general it is an abstraction of the real program P because we may (and must) forget some useless parts of the program that have no effect on property ϕ. We have also to translate the property ϕ into a state-property in $(S, \to)$ that is decidable for WSTS.
2. The second step is to look for a decidable ordering having these two desired properties. If we find such ordering, we just verify whether $(S, \to)$ satisfies the state-property ϕ. Let us consider the case in which we found a well ordering $\leq$ but $(S, \to, \leq)$ is unfortunately not monotone. One may look for a computable abstraction $(S', \to', \leq')$ of $(S, \to, \leq)$ where $(S', \leq')$ is an abstraction of $(S, \leq)$ such that the new transition relation $\to'$ (between abstract states in S') is monotone with respect to $\leq'$ which is still well and then A' is a WSTS.

Let us give two examples following this strategy that allows to analyse programs that are not directly translatable into WSTS.

1. Recall that the usual ordering on positive integers is well (Dickson Lemma) but it is not monotone on (general) counters machines because the zero-test guards are typically not monotone: but if we replace each zero test by a reset operation, the resulting machine is a WSTS (for the usual ordering extended on vectors) that over-approximates the original counter machine.
2. The subword ordering on finite words is well (Higman's Theorem) but it is not monotone on fifo machines ; however, it is monotone on fifo machines with the lossy semantics, hence lossy fifo machines are WSTS for the subword ordering. Lossy fifo machines over-approximates original fifo machines.

Both reset counter machines and lossy fifo machines can serve as WSTS abstractions of programs and these WSTS abstractions allow to study safety properties in original programs.

References

[1] Parosh Aziz Abdulla, Karlis Cerans, Bengt Jonsson & Yih-Kuen Tsay (2000): *Algorithmic Analysis of Programs with Well Quasi-ordered Domains*. *Inf. Comput.* 160(1-2), pp. 109–127, doi:10.1006/inco.1999.2843.

[2] Alain Finkel (1987): *A generalization of the procedure of Karp and Miller to well structured transition system*. In Thomas Ottmann, editor: *Proceedings of the 14th International Colloquium on Automata, Languages and Programming (ICALP'87)*, *Lecture Notes in Computer Science* 267, Springer-Verlag, Karlsruhe, Germany, pp. 499–508, doi:10.1007/3-540-18088-5_43. Available at `http://www.lsv.fr/Publis/PAPERS/PDF/F-icalp87.pdf`.

[3] Alain Finkel & Philippe Schnoebelen (1998): *Fundamental Structures in Well-Structured Infinite Transition Systems*. In Claudio L. Lucchesi & Arnaldo V. Moura, editors: *Proceedings of the 3rd Latin American Symposium on Theoretical Informatics (LATIN'98)*, *Lecture Notes in Computer Science* 1380, Springer, Campinas, Brasil, pp. 102–118, doi:10.1007/BFb0054314. Available at `http://www.lsv.ens-cachan.fr/Publis/PAPERS/PS/FinSch-latin98.ps`.

[4] Alain Finkel & Philippe Schnoebelen (2001): *Well-Structured Transition Systems Everywhere!* *Theoretical Computer Science* 256(1-2), pp. 63–92, doi:10.1016/S0304-3975(00)00102-X. Available at `http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/FinSch-TCS99.pdf`.