

Streaming String Transducers

Rajeev Alur

Department of Computer and Information Science,
University of Pennsylvania
alur@cis.upenn.edu

Streaming string transducers define (partial) functions from input strings to output strings. A streaming string transducer makes a single pass through the input string and uses a finite set of variables that range over strings from the output alphabet. At every step, the transducer processes an input symbol, and updates all the variables in parallel using assignments whose right-hand-sides are concatenations of output symbols and variables with the restriction that a variable can be used at most once in a right-hand-side expression. The expressiveness of streaming string transducers coincides with the class of “regular” transductions that can be equivalently defined using two-way deterministic finite-state transducers and using monadic second-order logic. The problems of checking functional equivalence of two streaming transducers, and of checking whether a streaming transducer satisfies pre/post verification conditions specified by finite automata, are solvable in PSPACE. These decision procedures also generalize to the model of streaming transducers over *data strings*—strings over symbols tagged with data values over a potentially infinite data domain that supports only the operations of equality and ordering. We identify a class of imperative and a class of functional programs, manipulating lists of data items, which can be effectively translated to such streaming data-string transducers. Our results lead to algorithms for assertion checking and for checking functional equivalence of two programs, written possibly in different programming styles, for commonly used routines such as insert, delete, and reverse.

This talk is based on results reported in [2] and [1].

References

1. Alur, R., Černý, P.: Expressiveness of streaming string transducers. In: Proc. 30th FSTTCS, pp. 1–12 (2010)
2. Alur, R., Černý, P.: Streaming transducers for algorithmic verification of single-pass list-processing programs. In: Proc. 38th POPL, pp. 599–610 (2011)