DOWNWARD CLOSURES OF INDEXED LANGUAGES

GEORG ZETZSCHE
PROPOSAL FOR 25 MINUTE TALK

Abstractions. A fruitful idea in the analysis of complex systems is that of abstractions: Instead of working with the original model, one considers a model that has a simpler structure but preserves pertinent properties.

A prominent example of such an abstraction is the *Parikh image*, which is available whenever the semantics of a model is given as a formal language. If L is a language over an alphabet $X = \{x_1, \dots, x_n\}$, then its *Parikh image* $\Psi(L)$ consists of all vectors $(a_1, \dots, a_n) \in \mathbb{N}^n$ such that there is a $w \in L$ in which x_i occurs a_i times for $i \in \{1, \dots, n\}$. The Parikh image is a useful abstraction, provided that for the languages L at hand, $\Psi(L)$ can be described using simpler means than L itself. This means, most applications of Parikh images are confined to situations where $\Psi(L)$ is semilinear (or, equivalently, definable in Presburger arithmetic). For example, this is the case for context-free languages [10], languages accepted by blind (or revesal-bounded) counter automata [7], and combinations thereof [1, 4, 6].

Of course, there are important types of languages that do not guarantee semilinearity of their Parikh image. For example, Parikh images of languages accepted by higher-order pushdown automata are not semilinear in general: It is easy to construct a second-order pushdown automaton for the language $\{a^{2^n} \mid n \geq 0\}$.

Downward closures. However, there is an abstraction of languages that guarantees a simple description for *every language* whatsoever and still reflects important properties of the abstracted language—the downward closure. For words $u, v \in X^*$, we write $u \preceq v$ if $u = u_1 \cdots u_n$ and $v = v_0u_1v_1 \cdots u_nv_n$ for some $u_1, \dots, u_n, v_0, \dots, v_n \in X^*$. Then, the *downward closure of* L is defined as

$$L{\downarrow} = \{u \in X^* \mid \exists v \in L \colon u \preceq v\}.$$

In other words, the downward closure consists of the set of all (not necessarily contiguous) subwords of members of L. It has the remarkable property that it is regular for every set of words $L \subseteq X^*$, which is due to the fact that the subword ordering $\preceq$ is a well-quasi-ordering [5].

Moreover, downward closures preserve useul information about the abstracted language. Suppose L describes the behavior of a system that is observed through a lossy channel, meaning that on the way to the observer, arbitrary actions can get lost. Then, $L{\downarrow}$ is the set of words received by the observer [3]. Hence, given the downward closure as a finite automaton, we can decide whether two systems are equivalent under such observations, and even whether one system includes the behavior of another.

However, while we know that for every L, there *exists* a finite automaton for $L{\downarrow}$, it is not clear in general how to *compute* them. In fact, there are examples of languages classes for which downward closures are known not to be computable [2, 9]. For an overview of the language classes for which computability is known, see [11].

Recently, a new approach to the computation of downward closures has been proposed. For a wide range of language classes, the approach reduces the computation to the *simultaneous unboundedness problem (SUP)* [11]. The latter asks, given a language $L \subseteq a_1^* \cdots a_n^*$, whether for every $k \in \mathbb{N}$, there are $x_1, \ldots, x_n \geq k$ with $a_1^{x_1} \cdots a_n^{x_n} \in L$. In fact, this result can be used to compute downward closures for indexed languages. Since these coincide with those languages accepted by second-order pushdown automata [8], we have the following.

Theorem 1. *Downward closures are computable for second-order pushdown automata.*

In this talk, I will sketch the method for computing downward closures of indexed languages.

References

[1] P. Buckheister and G. Zetzsche. "Semilinearity and Context-Freeness of Languages Accepted by Valence Automata". In: *Proc. of the 38th International Symposium on Mathematical Foundations of Computer Science (MFCS 2013).* Vol. 8087. LNCS. Heidelberg: Springer-Verlag, 2013, pp. 231–242.

[2] H. Gruber, M. Holzer, and M. Kutrib. "The size of Higman-Haines sets". In: *Theoretical Computer Science* 387.2 (2007), pp. 167–176.

[3] P. Habermehl, R. Meyer, and H. Wimmel. "The Downward-Closure of Petri Net Languages". In: *Proc. of the 37th International Colloquium on Automata, Languages and Programming (ICALP 2010).* Vol. 6199. LNCS. Heidelberg: Springer-Verlag, pp. 466–477.

[4] M. Hague and A. W. Lin. "Model Checking Recursive Programs with Numeric Data Types". In: *Computer Aided Verification.* Vol. 6806. LNCS. Heidelberg: Springer-Verlag, 2011, pp. 743–759.

[5] L. H. Haines. "On free monoids partially ordered by embedding". In: *Journal of Combinatorial Theory* 6.1 (1969), pp. 94–98.

[6] T. Harju, O. Ibarra, J. Karhumäki, and A. Salomaa. "Some Decision Problems Concerning Semilinearity and Commutation". In: *Journal of Computer and System Sciences* 65.2 (2002), pp. 278–294.

[7] O. H. Ibarra. "Reversal-bounded multicounter machines and their decision problems". In: *Journal of the ACM* 25.1 (1978), pp. 116–133.

[8] A. N. Maslov. "Multilevel stack automata". In: *Problems of Information Transmission* 12.1 (1976), pp. 38–42.

[9] R. Mayr. "Undecidable problems in unreliable computations". In: *Theoretical Computer Science* 297.1-3 (2003), pp. 337–354.

[10] R. J. Parikh. "On Context-Free Languages". In: *Journal of the ACM* 13.4 (1966), pp. 570–581.

[11] G. Zetzsche. *An approach to computing downward closures.* To appear in Proc. of 42nd International Colloquium on Automata, Languages and Programming (ICALP 2015). Full version available at `http://arxiv.org/abs/1503.01068`. 2015.

Technische Universität Kaiserslautern, Fachbereich Informatik, Concurrency Theory Group

E-mail address: `zetzsche@cs.uni-kl.de`