THEORIE-TAGE 2018
K. Reinhardt (Herausgeber): Theorietage 2018, Wittenberg, 24. – 27.9.2018
Universität Halle-Wittenberg, Technischer Bericht, S. 1–4.

One-Dimensional Tiling Systems and String Rewriting

Alfons Geser$^{(A)}$ Dieter Hofbauer$^{(B)}$ Johannes Waldmann$^{(A)}$

$^{(A)}$HTWK Leipzig, Germany

$^{(B)}$ASW – Berufsakademie Saarland, Germany

Abstract

We use one-dimensional tiling systems (strictly locally testable languages) to over-approximate reachability sets in string rewriting, and apply this to prove termination automatically. This refines the root labeling method by restricting to right-hand sides of forward closures.

1. Motivation

The k-tiles of a string are its factors (contiguous sub-words) of length k. The tiled version $\mathsf{tiled}_k(R)$ of a rewrite system R over Σ describes the action of R on tiled words. Since $\mathsf{tiled}_k(R)$ has a larger alphabet (namely, Σ^k), it may be easier to analyze:

Example 1.1 *For the rewriting system $R = \{aa \to aba\}$, we have $\mathit{tiled}_2(R) = \{[aa] \to [ab, ba]\}$. It is easy to see that $\mathit{tiled}_2(R)$ terminates, since each rule application reduces the number of occurrences of tile aa. The original system R does not admit such a proof of termination, since R does not remove any letters.*

2. Tiling Systems

A tiling system specifies a language by considering prefixes, factors, and suffixes of bounded length. We give an equivalent definition that allows a uniform description, using end markers $\triangleleft, \triangleright \notin \Sigma$. A similar method is used for two-dimensional tiling [3].

Definition 2.1 *For $w \in \Sigma^*$, the k-bordered version is $\mathit{bord}_k(w) = \triangleleft^{k-1} w \triangleright^{k-1}$ over $\Sigma \cup \{\triangleleft, \triangleright\}$. The k-tiled version $\mathit{tiled}_k(w)$ is the string over Σ^k of all factors of length k, or ϵ in case $|w| < k$. Let $\mathit{tiles}_k(w)$ denote $\mathit{alphabet}(\mathit{tiled}_k(w))$, the set of letters in $\mathit{tiled}_k(w)$.*

Example 2.2 *$\mathit{tiled}_2(\mathit{bord}_2(abbb)) = \mathit{tiled}_2(\triangleleft abbb \triangleright) = [\triangleleft a, ab, bb, bb, b\triangleright]$, $\mathit{tiles}_2(\mathit{bord}_2(abbb)) = \{\triangleleft a, ab, bb, b\triangleright\}$, $\mathit{tiles}_2(\mathit{bord}_2(a)) = \{\triangleleft a, a\triangleright\}$, $\mathit{tiles}_2(\mathit{bord}_2(\epsilon)) = \{\triangleleft\triangleright\}$, and $\mathit{tiled}_3(\mathit{bord}_3(a)) = [\triangleleft\triangleleft a, \triangleleft a\triangleright, a\triangleright\triangleright]$.*

The language defined by a set of tiles T of length k is $\{w \in \Sigma^* \mid \mathsf{tiles}_k(\mathsf{bord}_k(w)) \subseteq T\}$. This is an equivalent definition of the class of strictly locally k-testable languages [6, 8], a subclass of regular languages. We will use one-dimensional tiling systems to over-approximate reachability sets in string rewriting.

3. Rewriting and Reachability

A *string rewriting system* over alphabet Σ consists of rewrite rules. We use standard concepts and notation, with this extension: a *constrained rule* is a pair of strings l, r with a constraint $c \in \{\mathsf{factor}, \mathsf{suffix}\}$, indicating where the rule is to be applied. The rewrite relations are:

$$\to_{l,r,\mathsf{factor}} = \{(xly, xry) \mid x, y \in \Sigma^*\}, \qquad \to_{l,r,\mathsf{suffix}} = \{(xl, xr) \mid x \in \Sigma^*\},$$

A constrained rule (l, r, c) is denoted by $l \to_c r$. Standard rewriting corresponds to the factor constraint, therefore $\to$ abbreviates $\to_{\mathsf{factor}}$. For a rewrite system R, we define $\to_R$ as the union of the rewrite relations of its rules. For a relation ρ on Σ^* and a set $L \subseteq \Sigma^*$, let $\rho(L) = \{y \mid \exists x \in L, (x,y) \in \rho\}$. Hence the set of R*-reachable* strings from L is $\to_R^* (L)$, or $R^*(L)$ for short. A language $L \subseteq \Sigma^*$ is *closed w.r.t.* R if $\to_R(L) \subseteq L$.

Example 3.1 *For* $R = \{cc \to_{\mathsf{factor}} bc, ba \to_{\mathsf{factor}} ac, c \to_{\mathsf{suffix}} bc, b \to_{\mathsf{suffix}} ac\}$ *, we have* $bbb \to_{\mathsf{suffix}} bbac \to_{\mathsf{factor}} bacc$*. The reachability set* $R^*(\{bc, ac\})$ *is* $(a+b)b^*c$*. This set is closed w.r.t.* R*.*

R over Σ is called *terminating on* $L \subseteq \Sigma^*$ if for each $w \in L$, each R-derivation starting at w is finite, and R is *terminating* if it is terminating on Σ^*.

4. Closures

Given a rewrite system R over alphabet Σ, a *closure* $C = (l, r)$ of R is a pair of strings with $l \to_R^+ r$ such that each position of r took part in some step of the derivation. In particular, we use *forward closures* [5]. Their right-hand sides can be computed by (factor and) suffix rewriting.

Proposition 4.1 *[4]* $\mathsf{RFC}(R) = (R \cup \mathsf{forw}(R))^*(\mathsf{rhs}(R))$*, where*

$$\mathsf{forw}(R) = \{l_1 \to_{\mathsf{suffix}} r \mid (l_1 l_2 \to r) \in R, l_1 \neq \epsilon \neq l_2\}.$$

They are related to termination by

Theorem 4.2 *[1]* R *is terminating (on* Σ^**) if and only if* R *is terminating on* $\mathsf{RFC}(R)$*.*

Example 4.3 *For* $R = \{cc \to bc, ba \to ac\}$ *we have* $\mathsf{forw}(R) = \{c \to_{\mathsf{suffix}} bc, b \to_{\mathsf{suffix}} ac\}$ *and* $\mathsf{RFC}(R) = (a+b)b^*c$*, cf. Example 3.1. As* $\mathsf{RFC}(R)$ *contains no* R*-redex,* R *is trivially terminating on* $\mathsf{RFC}(R)$*, therefore by Theorem 4.2,* R *is terminating.*

In the following, we use tiled rewriting to approximate $\mathsf{RFC}(R)$. This allows to obtain the termination proof of Example 4.3 automatically.

5. Tiled Rewrite Systems

We enlarge the alphabet of a rewrite system by tiling.

Definition 5.1 *For a rule $l \to_{\mathsf{factor}} r$ over Σ we define*

$$\mathit{tiled}_k(l \to_{\mathsf{factor}} r) = \{\mathit{tiled}_k(xly) \to_{\mathsf{factor}} \mathit{tiled}_k(xry) \mid x \in \mathit{tiles}_{k-1}(\lhd^*\Sigma^*), y \in \mathit{tiles}_{k-1}(\Sigma^*\rhd^*)\}$$

and for a given set of tiles $S \subseteq \mathit{tiles}_k(\Sigma^*)$

$$\mathit{tiled}_S(l \to_{\mathsf{factor}} r) = \mathit{tiled}_k(l \to_{\mathsf{factor}} r) \cap S^* \times S^* \times \{\mathsf{factor}\}.$$

Both tiled_S and tiled_k are extended to sets of rules.

Example 5.2 *$\mathit{tiled}_2(ba \to_{\mathsf{factor}} ac)$ contains 16 rules, among them $[\lhd b, ba, a\rhd] \to [\lhd a, ac, c\rhd]$, $[\lhd b, ba, aa] \to [\lhd a, ac, ca], \ldots, [ab, ba, a\rhd] \to [aa, ac, c\rhd], \ldots, [cb, ba, ac] \to [ca, ac, cc]$. For $S = \{ac, ba, bb, cc\}$ we get $\mathit{tiled}_S(ba \to_{\mathsf{factor}} ac) = \{[bb, ba, ac] \to [ba, ac, cc]\}$ and for any strict subset T of S, $\mathit{tiled}_T(ba \to_{\mathsf{factor}} ac) = \emptyset$.*

Derivations w.r.t. R and $\mathsf{tiled}_k(R)$ are bi-similar, and we obtain

Theorem 5.3 *For $S \subseteq \mathit{tiles}_k(\Sigma^*)$, if $\mathit{Lang}(S)$ is closed w.r.t. R, then R is terminating on $\mathit{Lang}(S)$ if and only if $\mathit{tiled}_S(R)$ is terminating.*

Example 5.4 *(cont.) $R = \{cc \to bc, ba \to ac\}$. $\mathit{RFC}(R) = \mathit{Lang}(S)$ for the set of tiles $S = \{\lhd a, \lhd b, ab, ac, bb, bc, c\rhd\}$. The set $\mathit{RFC}(R)$ is closed w.r.t. R by definition and $\mathit{tiled}_S(R)$ is empty, therefore terminating. By Theorem 5.3, R is terminating on $\mathit{RFC}(R)$ and by Theorem 4.2, R is terminating.*

We obtain a set of tiles for using Theorem 5.3 by the following algorithm.

Algorithm 5.5
- *Input: A rewrite system R over Σ, a set of tiles $T \subseteq \mathit{tiles}_k(\Sigma^*)$.*
- *Output: A set of tiles $S \subseteq \mathit{tiles}_k(\Sigma^*)$ such that $T \subseteq S$ and $\mathit{Lang}(S)$ is closed w.r.t. R.*
- *Implementation: $S = \bigcup_i S_i$ for the sequence given by*

$$S_0 = T,\ S_{i+1} = S_i \cup \mathit{alphabet}(\mathit{rhs}(\mathit{tiled}_k(R) \cap \mathit{lhs}^{-1}(S_i^*))).$$

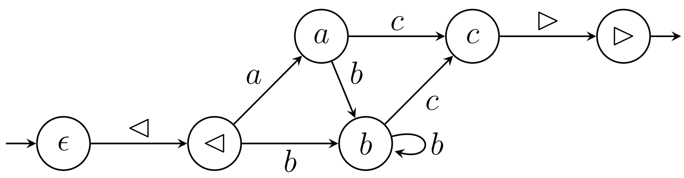
In each step, each rule is extended by contexts of length $k-1$ on both sides such that the extended left-hand side can be covered. Then the tiles of the extended right-hand side are added. The algorithm terminates since (S_i) is increasing w.r.t. $\subseteq$ and bounded by $\mathsf{tiles}_k(\Sigma^*)$.

6. Representing Tiling Systems by Automata

For an efficient implementation of the closure algorithm 5.5, we represent a set of tiles of length k by a deterministic (not necessarily complete or minimal) automaton over $\Sigma \cup \{\lhd, \rhd\}$ with states from $\lhd^{<k} \cup \mathsf{tiles}_{k-1}(\Sigma^*) \cup \{\rhd^{k-1}\}$, initial state ϵ and final state $\rhd^{k-1}$. For each transition $p \xrightarrow{c} q$, state q is the suffix of length $k-1$ of $p \cdot c$. Such an automaton A represents the set of tiles

$$\mathsf{tiles}(A) = \{p \cdot c \mid p \xrightarrow{c}_A q, |p| = k-1\}.$$

Example 6.1 *(Example 5.4 cont'd) This automaton represents* $\{\triangleleft a, \triangleleft b, ab, ac, bb, bc, c\triangleright\}$*:*



Adding tiles in Algorithm 5.5 then corresponds to adding states and edges. With the automata representation, we can quickly check whether a left-hand side of a rule is covered by the current set of tiles. Our implementation can handle automata with 10^4 transitions (tiles) in a few seconds.

7. Discussion

We have presented a method to compute a regular over-approximation of reachability sets, using tiling systems, represented as automata, and we applied this to termination analysis. The root labeling method [7] corresponds to tiling on the full set Σ^*, for width 2. Our method allows any width, and restricts the set of tiles. Restriction to right-hand sides of forward closures (RFC) had already been applied to enhance the power of the matchbound termination proof method [2]. Our method decouples the RFC method from the matchbound method.

References

[1] NACHUM DERSHOWITZ, Termination of Linear Rewriting Systems. In: SHIMON EVEN, ODED KARIV (eds.), *Automata, Languages and Programming, 8th Colloquium, Acre (Akko), Israel, July 13-17, 1981, Proceedings*. Lecture Notes in Computer Science 115, Springer, 1981, 448–458.

[2] ALFONS GESER, DIETER HOFBAUER, JOHANNES WALDMANN, Match-Bounded String Rewriting Systems. *Appl. Algebra Eng. Commun. Comput.* **15** (2004) 3-4, 149–171.

[3] DORA GIAMMARESI, ANTONIO RESTIVO, Two-Dimensional Languages. In: ARTO SALOMAA, GRZEGORZ ROZENBERG (eds.), *Handbook of Formal Languages*. 3, Springer, 1997, 215–267.

[4] MIKI HERMANN, *Divergence des systèmes de réécriture et schématisation des ensembles infinis de termes*. Habilitation, Université de Nancy, France, 1994.

[5] DALLAS S. LANKFORD, D. R. MUSSER, *A finite termination criterion*. Technical report, Information Sciences Institute, Univ. of Southern California, Marina-del-Rey, CA, 1978.

[6] ROBERT MCNAUGHTON, SEYMOUR PAPERT, *Counter-Free Automata*. MIT Press, 1971.

[7] CHRISTIAN STERNAGEL, AART MIDDELDORP, Root-Labeling. In: ANDREI VORONKOV (ed.), *Rewriting Techniques and Applications, 19th International Conference, RTA 2008, Hagenberg, Austria, July 15-17, 2008, Proceedings*. Lecture Notes in Computer Science 5117, Springer, 2008, 336–350.

[8] YECHEZKEL ZALCSTEIN, Locally testable languages. *Journal of Computer and System Sciences* **6** (1972) 2, 151 – 167.