On Closures in String Rewriting

Alfons Geser
HTWK Leipzig

Abstract

The set of overlap closures can be characterized as the semantics of composition trees. Rewrite systems for composition trees can be used to prove that the existence of loops implies the existence of looping forward closures, or to prove the correctness of a characterization of the set of right-hand sides of closures. We show that, for a quasi-terminating string rewrite system, the existence of cycles is equivalent to the existence of looping forward closures. This improves upon a result of Guttag et al. 1983.

2012 ACM Subject Classification Theory of computation → Rewrite systems

Keywords and phrases Termination, String rewriting, Forward Closure, Overlap Closure, Composition Tree, Looping Derivation, Cyclic Derivation

1 Overlap Closures and Composition Trees

We generally assume that a string rewrite system R over an alphabet Σ is given.

A set of closures is usually defined inductively. Guttag et al. [5] introduced overlap closures for term rewriting. The following is a version for string rewriting.

▶ **Definition 1.** *The set OC of* overlap closures *is defined as the least set of Σ^*-pairs that*
(1) *includes R and satisfies*
(2) *if $(s, tx) \in OC$ and $(xu, v) \in OC$ for some $t, x, u \neq \epsilon$ then $(su, tv) \in OC$;*
(2') *if $(s, xt) \in OC$ and $(ux, v) \in OC$ for some $t, x, u \neq \epsilon$ then $(us, vt) \in OC$;*
(3) *if $(s, xuy) \in OC$ and $(u, v) \in OC$ then $(s, xvy) \in OC$;*
(3') *if $(u, v) \in OC$ and $(xvy, t) \in OC$ then $(xuy, t) \in OC$.*

When dealing with closures, it is useful to represent a closure by a composition tree. The composition tree describes the way the closure is formed.

▶ **Definition 2** ([4]). *Define the signature $\Omega = \{1, 2, 2', 3, 3', 4\}$, where 1 is nullary, 4 is ternary, and the other symbols are binary. The set CT of* composition trees *is defined as the set of ground terms over Ω.*

▶ **Definition 3.** *A composition tree c represents a set $\langle c\rangle$ of Σ^*-pairs, as follows:*

$$\begin{aligned}
\langle 1\rangle &= R,\\
\langle 2(c_1, c_2)\rangle &= \{(su, tv) \mid (s, tx) \in \langle c_1\rangle, (xu, v) \in \langle c_2\rangle, t, x, u \neq \epsilon\},\\
\langle 2'(c_1, c_2)\rangle &= \{(us, vt) \mid (s, xt) \in \langle c_1\rangle, (ux, v) \in \langle c_2\rangle, t, x, u \neq \epsilon\},\\
\langle 3(c_1, c_2)\rangle &= \{(s, xvy) \mid (s, xuy) \in \langle c_1\rangle, (u, v) \in \langle c_2\rangle\},\\
\langle 3'(c_1, c_2)\rangle &= \{(xuy, t) \mid (u, v) \in \langle c_1\rangle, (xvy, t) \in \langle c_2\rangle, \},\\
\langle 4(c_1, c_2, c_3)\rangle &= \{(swu, tzv), \mid (s, tx) \in \langle c_1\rangle, (u, yv) \in \langle c_2\rangle, (xwy, z) \in \langle c_3\rangle, t, x, y, v \neq \epsilon\}\ .
\end{aligned}$$

This is conveniently extended to sets S of composition trees:

$$\langle S\rangle = \bigcup_{c \in S} \langle c\rangle.$$

For each case in the inductive definition, we have a symbol in Ω that names this case. For instance symbol 1 means Case (1): the overlap closure is a rule in R. If an overlap closure (s, tx) is represented by the composition tree c_1 and an overlap closure (xu, v) is represented by the composition tree c_2, then the overlap closure (su, tv) obtained by Case (2), is represented by the composition tree $2(c_1, c_2)$. We may so write Case (2) more succinctly as:

If $\langle c_1 \rangle \subseteq \mathsf{OC}$ and $\langle c_2 \rangle \subseteq \mathsf{OC}$ then $\langle 2(c_1, c_2) \rangle \subseteq \mathsf{OC}$.

Let CT_0 denote the composition trees that do not contain the symbol 4. Then the set of overlap closures is characterized like this:

▶ **Lemma 4.** *$OC = \langle CT_0 \rangle$.*

Adding symbol 4 does not increase expressiveness, since we have $\langle 4(c_1, c_2, c_3) \rangle \subseteq \langle 2(c_1, 2'(c_2, c_3)) \rangle$.

▶ **Lemma 5.** *$OC = \langle CT \rangle$.*

Symbol 4 will be needed later in Section 3.

▶ **Example 6.** Suppose $R = \{ad \to abb, bd \to db, bdbdd \to c\}$. Then (bd, db) is a closure by Case (1). Using this closure twice, we get a closure (bdd, ddb), by Case (2) with $t = d = u$, $x = b$. From the two closures (bd, db) and $(bdbdd, c)$, we get a closure $(bbddd, c)$ by Case (3′) with $v = db$, $x = b$, $y = dd$. From the closures (ad, abb), (bdd, ddb), and $(bbddd, c)$, we get a symbol-4 closure $(addbdd, acb)$ by $s = ad$, $t = a$, $x = bb$, $u = bdd$, $y = dd$, $v = b$, $w = d$, $z = c$. With composition trees, this is expressed as $(bd, db) \in \langle 1 \rangle$, $(bdd, ddb) \in \langle 2(1,1) \rangle$, $(bbddd, c) \in \langle 3'(1,1) \rangle$, and $(addbdd, acb) \in \langle 4(1, 2(1,1), 3'(1,1)) \rangle$. The same closure, $(addbdd, acb)$, may be assigned a different composition tree: $(add, abdb) \in \langle 2(1,1) \rangle$, $(bdd, ddb) \in \langle 2(1,1) \rangle$, and $(bdbdd \to c) \in \langle 1 \rangle$ yield the symbol-4 closure $(addbdd, acb) \in \langle 4(2(1,1), 2(1,1), 1) \rangle$, by $s = add$, $t = a$, $x = bdb$, $u = bdd$, $y = dd$, $v = b$, $w = \epsilon$, $z = c$. ◀

2 Looping Closures and Tree Rearrangement

In order to demonstrate what one can do with composition trees, we recall a characterization result for the existence of loops, i.e. of derivations of the form $t \to^+ utv$. Let us call an overlap closure of the form (t, utv) a *looping overlap closure*. Likewise, let us call a forward closure of the form (t, utv) a *looping forward closure*.

The starting point is the following characterization of overlap closures.

▶ **Lemma 7.** *[4] OC is equal to the set of pairs (t, u) such that there is a derivation from t to u in which every inner interposition of t is touched.*

Here "inner interposition" means position between letters.

▶ **Example 8.** In Example 6, to the overlap closure $(addbdd, acb)$ there are the derivations

$$\underline{a|d}|d|b|d|d \to abb|d|\underline{b|d}|d \to abb|d|d\underline{b|d} \to ab\underline{b|d}|ddb \to a\underline{bdb|dd}b \to acb,$$
$$a|d|d|\underline{b|d}|d \to a|d|d|d\underline{b|d} \to \underline{a|d}|d|ddb \to ab\underline{b|d}|ddb \to a\underline{bdb|dd}b \to acb,$$

corresponding to the composition trees $4(1, 2(1,1), 3'(1,1))$ and $4(2(1,1), 2(1,1), 1)$, respectively. Matching left hand sides are underlined. The inner interpositions that are not yet touched are indicated by a stroke. ◀

▶ **Lemma 9.** *[4] Every SRS that has a loop, has a looping overlap closure.*

Proof. If $(\epsilon, r) \in R$ then R has a loop $\epsilon \to r$, and $(\epsilon, r) \in \mathsf{OC}$. So assume that $\epsilon \notin \mathsf{lhs}(R)$. The proof is by induction on the length of t in the given derivation $t = t_0 \to t_1 \to \cdots \to t_n = utv$. If every inner interposition of t is touched during this derivation, then we have an overlap closure (t, utv) by Lemma 7. Otherwise, there is an inner inperposition in t that is not touched, and which has therefore a residual, unique by $\epsilon \notin \mathsf{lhs}(R)$, in every string t_i. At these residuals, the derivation can be split into two derivations $t'_0 \to^= t'_1 \to^= \cdots \to^= t'_n$ and $t''_0 \to^= t''_1 \to^= \cdots \to^= t''_n$ such that $t_i = t'_i t''_i$ for all $i \in \{0, \ldots, n\}$. We have $utv = ut'_0 t''_0 v = t'_n t''_n$. By a case analysis on whether $|ut'_0| < |t'_n|$ or $|ut'_0| > |t'_n|$ or $|ut'_0| = |t'_n|$, one of the two derivations forms a loop, and by the inductive hypothesis the claim follows. ◀

Lemma 9 can be strengthened towards forward closures. The set of forward closures, introduced by Lankford and Musser [7] for term rewriting, can be defined in the string rewriting case simply as follows:

▶ **Definition 10.** *For a string rewrite system R the set* FC *of forward closures is defined as* $\mathsf{FC} = \langle \mathsf{Term}(\{1, 2, 3\}) \rangle$.

Here, $\mathsf{Term}(\{1, 2, 3\})$ denotes the set of ground terms over the signature $\{1, 2, 3\}$, a subsignature of Ω.

▶ **Theorem 11.** *[4] Every SRS that has a loop, has a looping forward closure.*

For the proof, we first introduce an interpretation ϱ on composition trees by $\varrho(1) = 2$,

$$\varrho(2(c_1, c_2)) = \varrho(3(c_1, c_2)) = \varrho(c_1)\varrho(c_2), \qquad \varrho(2'(c_1, c_2)) = \varrho(3'(c_1, c_2)) = \varrho(c_1)\varrho(c_2) + 1 \ .$$

For composition trees c_1, c_2 we define $c_1 <_\varrho c_2$ by $\varrho(c_1) < \varrho(c_2)$, and we say that c_1 is smaller than c_2. The order $<_\varrho$ is a reduction order.

▶ **Lemma 12.** *If there is a looping overlap closure with the composition tree $3'(c_1, c_2)$ or $2'(c_1, c_2)$, then there is also a looping overlap closure with a smaller composition tree.*

Proof. Let $(u, v) \in \langle c_1 \rangle$ and let $(xvy, t) \in \langle c_2 \rangle$ such that $(xuy, t) \in \langle 3'(c_1, c_2) \rangle$, by the definition of $\langle 3'(c_1, c_2) \rangle$. Moreover let this overlap closure be looping, i.e. let $t = rxuys$ for some $r, s \in \Sigma^*$. Then (xvy, t) and (u, v) form the looping overlap closure $(xvy, rxvys) \in \langle 3(c_2, c_1) \rangle$, and $\varrho(3(c_2, c_1)) < \varrho(3'(c_1, c_2))$ holds.

Similarly, one can show, by a case analysis, that to a looping overlap closure with composition tree $2'(c_1, c_2)$ there is a looping overlap closure with one of the smaller composition trees $c_1, 2(c_1, c_2), 3(c_2, c_1)$. ◀

▶ **Definition 13.** *The sets $[T]$ and $[F]$ of composition trees are given by the regular tree grammar with variables T, F, start variable T and rules*

$$T \to F \mid 2'(T, T) \mid 3'(T, T), \qquad F \to 1 \mid 2(F, F) \mid 3(F, F).$$

Note that by definition, $[F] = \mathsf{Term}(\{1, 2, 3\})$ and so $\langle [F] \rangle = \mathsf{FC}$ holds.

We now construct a term rewrite system P on Ω that satisfies three conditions:

1. P has a subset of $[T]$ as its normal forms. It does so by removing all $2'$ and $3'$ symbols from the left and right arguments of 2 and of 3. In other words, its left-hand sides are $f(g(c_1, c_2), c_3)$ and $f(c_1, g(c_2, c_3))$ where $f \in \{2, 3\}$ and $g \in \{2', 3'\}$.
2. P is terminating by $<_\varrho$.
3. For every left-hand side u in P, and the set V of all corresponding right-hand sides, we get $\langle u \rangle \subseteq \langle V \rangle$. Let us call this condition *semantic coverage*.

Semantic coverage can be rephrased as follows:

▶ **Lemma 14.** *For every $c \in \mathsf{CT}$ that admits a P rewrite step, and for every $(s,t) \in \langle c \rangle$ there is a $c' \in \mathsf{CT}$ such that both $c \to_P c'$ and $(s,t) \in \langle c' \rangle$.*

Note that c' may depend on s and t. Suitable such rules from P each with left-hand side $3(2'(c_1,c_2),c_3)$ are $3(2'(c_1,c_2),c_3) \to 2'(c_1,2(c_2,c_3))$, $3(2'(c_1,c_2),c_3) \to 2'(3(c_1,c_3),c_2)$, $3(2'(c_1,c_2),c_3) \to 2'(c_1,3(c_2,c_3))$. The term rewrite system P has 14 rules.

Proof of Theorem 11. Suppose R has a loop. By Lemma 9, it has a looping overlap closure (t,utv). We use induction on the size of the composition tree c of (t,utv). If c contains no $2'$ nor $3'$ symbols then it is the composition tree of a forward closure, and we are done. If $c = 3'(c_1,c_2)$ or $c = 2'(c_1,c_2)$ for some c_1,c_2, then by Lemma 12, there is an overlap closure with a smaller composition tree, and the claim follows by inductive hypothesis. Otherwise, the composition tree c admits a P rewrite step, $c \to_P c'$, and (t,utv) admits the smaller composition tree c'. Then again the claim follows by inductive hypothesis. ◀

Guttag et al. [5] show that a quasi-terminating string rewrite system has a cycle if and only if, it has a cyclic overlap closure, i.e. an overlap closure of the form (t,t). This result can be strengthened towards forward closure, as is shown next.

▶ **Theorem 15.** *If R is quasi-terminating then every loop is a cycle, i.e. whenever $t \to^+ utv$ then $u = \epsilon = v$.*

Proof. Suppose that $t \to^+ utv$ is a proper loop, i.e. $|uv| > 0$ holds. Then the infinite derivation $t \to^+ utv \to^+ u^2tv^2 \to^+ \dots$ shows that t has infinitely many descendants: $|t| < |utv| < |u^2tv^2| < \dots$. Hence R is not quasi-terminating. ◀

So for quasi-terminating R, if R has a cycle, then it has a loop, then it has a looping forward closure by Theorem 11. Conversely, if R has a looping forward closure, it has a loop, which is a cycle by Theorem 15. This proves:

▶ **Corollary 16.** *Let R be quasi-terminating. Then R has a cycle if, and only if, R has a looping* forward closure.

3 Right-hand Sides of Closures

In order to arrive at an inductive definition of the set of right-hand sides of forward closures, one needs an inductive definition of the set of forward closures that descends only at the left, i.e. the composition tree has only 1 in all its right branches. Hermann [6, Corollaire 2.16] introduced such a characterization for term rewriting:

▶ **Definition 17.** *For a string rewrite system R, let the set FC' be defined as the least set of derivations that includes R and satisfies*
1. *if $(s,tx) \in \mathsf{FC}'$ and $(xu,v) \in R$ for $t,x,u \neq \epsilon$ then $(su,tv) \in \mathsf{FC}'$.*
2. *if $(s,xuy) \in \mathsf{FC}'$ and $(u,v) \in R$ then $(s,xvy) \in \mathsf{FC}'$,*

▶ **Theorem 18.** *For every string rewrite system R, we have $\mathsf{FC}' = \mathsf{FC}$.*

The point of Hermann's characterization is that an impoverished variant inductively characterizes the set $\mathsf{rhs}(\mathsf{FC})$ of right-hand sides of forward closures.

▶ **Corollary 19.** *[2] For any string rewrite system R, the set $\mathsf{rhs}(\mathsf{FC})$ is equal to the least set S of strings that includes $\mathsf{rhs}(R)$ and satisfies*

1. *if* $tx \in S$ *and* $(xu, v) \in R$ *for* $t, x, u \neq \epsilon$ *then* $tv \in S$.
2. *if* $xuy \in S$ *and* $(u, v) \in R$ *then* $xvy \in S$,

This characterization of $\mathsf{rhs}(\mathsf{FC})$ is the starting point of several termination proof methods based on Dershowitz's [1] characterization of termination by termination on $\mathsf{rhs}(\mathsf{FC})$.

One can use composition trees to render Hermann's definition more compactly:

▶ **Definition 20.** *The set* $[H]$ *is given by the regular tree grammar with the start variable* H *and rules* $H \to 1 \mid 2(H, 1) \mid 3(H, 1)$.

▶ **Lemma 21.** *For every string rewrite system* R*, we have* $\mathit{FC}' = \langle [H] \rangle$.

Theorem 18 then amounts to $\langle \mathsf{Term}[\{1, 2, 3\}] \rangle = \langle [H] \rangle$. A proof can be done like in Section 2: First an interpretation δ on $\mathsf{Term}(\{1, 2, 3\})$ is defined by $\delta(1) = 2$ and $\delta(f(c_1, c_2)) = \delta(c_1) + 2 \cdot \delta(c_2)$ for all $f \in \{2, 3\}$. The interpretation δ induces a reduction order $<_\delta$ on $\mathsf{Term}(\{1, 2, 3\})$. Then a term rewrite system Q_F on $\{1, 2, 3\}$ is defined which (1) has a subset of $[H]$ as its set of normal forms, (2) terminates by $<_\delta$, and (3) satisfies semantic coverage. For this purpose, it must remove all 2 and 3 symbols from the right arguments of 2 and of 3. In other words, its left-hand sides are $f(c_1, g(c_2, c_3))$ where $f, g \in \{2, 3\}$. It turns out that Q_F must comprise the rules

$$
\begin{aligned}
&2(c_1, 2(c_2, c_3)) \to 2(2(c_1, c_2), c_3), && 2(c_1, 2(c_2, c_3)) \to 2(3(c_1, c_2), c_3),\\
&2(c_1, 3(c_2, c_3)) \to 3(2(c_1, c_2), c_3), && 3(c_1, 2(c_2, c_3)) \to 3(3(c_1, c_2), c_3),\\
&3(c_1, 3(c_2, c_3)) \to 3(3(c_1, c_2), c_3).
\end{aligned}
$$

This finishes the proof sketch of Theorem 18.

Now we aim at a similar characterization of the set of *overlap closures*. The following material is an excerpt from Geser et al. [3].

▶ **Definition 22.** *The set* CT_N *is given by the regular tree grammar with variables* T, D *(top, deep), start variable* T*, and rules*

$$T \to 3'(1, T) \mid D, \qquad D \to 1 \mid 2(D, 1) \mid 2'(D, 1) \mid 3(D, 1) \mid 4(D, D, 1).$$

▶ **Definition 23.** *The sets* OC_N *and* OC' *are defined as* $\mathit{OC}_N = \langle \mathit{CT}_N \rangle$ *and* $\mathit{OC}' = \langle [D] \rangle$.

The rules for T represent an initial derivation before a closure in OC':

▶ **Lemma 24.** $\mathit{OC}_N = \{(s, t) \mid s \to_R^* s' \wedge (s', t) \in \mathit{OC}'\}$.

The definition of OC' could also be spelled out as an inductive definition similar to that of OC, where closures are overlapped with rules, Case ($3'$) is dropped and Case (4) is added.

▶ **Theorem 25.** *For every string rewrite system* R*, we have* $\mathit{OC} = \mathit{OC}_N$.

Then by Lemma 24, we have $\mathsf{rhs}(\mathsf{OC}) = \mathsf{rhs}(\mathsf{OC}_N) = \mathsf{rhs}(\mathsf{OC}')$, whence we get:

▶ **Corollary 26.** $\mathit{rhs}(\mathit{OC})$ *is equal to the least set* S *that includes* $\mathit{rhs}(R)$ *and satisfies*

1. *if* $tx \in S$ *and* $(xu, v) \in R$ *for some* $t, x, u \neq \epsilon$ *then* $tv \in S$;
2. *if* $xt \in S$ *and* $(ux, v) \in R$ *for some* $t, x, u \neq \epsilon$ *then* $vt \in S$;
3. *if* $xuy \in S$ *and* $(u, v) \in R$ *then* $xvy \in S$;
4. *if* $tx \in S$ *and* $yv \in S$ *and* $(xwy, z) \in R$ *for some* $t, x, y, v \neq \epsilon$ *then* $tzv \in S$.

This characterization of $\mathsf{rhs}(\mathsf{OC})$ turns out useful in proofs of relative termination, where the right-hand sides of forward closures cannot be applied.

In the remainder of this section, a proof of Theorem 25 will be sketched again in a similar way to the proof in Section 2. We are going to construct a term rewrite system Q on Ω that has a subset of CT_N as its set of normal forms. It must remove all non-1 symbols from the left argument of $3'$, and remove all non-1 symbols from the rightmost argument of $2, 2', 3$, and 4. Also, it must remove all $3'$ that are below some non-$3'$. These conditions already determine the set of left-hand sides of Q. For each left-hand side ℓ, the set of corresponding right-hand sides must cover ℓ semantically. These right-hand sides are obtained by a case analysis. The term rewrite system Q has 55 rules. For lack of space, only a few rules are exemplified here.

We bubble-up $3'$ symbols, e. g., $2(3'(c_1, c_2), c_3) \to 3'(c_1, 2(c_2, c_3))$, and we rotate to move non-1 symbols, e. g., $2(c_1, 2(c_2, c_3)) \to 2(2(c_1, c_2), c_3)$. Rotation below $3'$ goes from left to right, all other rotations go from right to left. The rules $2(c_1, 2'(c_2, c_3)) \to 4(c_1, c_2, c_3)$ and $2'(c_1, 2(c_2, c_3)) \to 4(c_2, c_1, c_3)$ show that symbol 4 cannot be avoided. Of course, Q_F is a subset of Q.

Termination of Q follows from the reduction order $>$ given by a lexicographic combination of an interpretation ρ that decreases under rotation, and an interpretation σ that decreases under bubbling. A lemma like Lemma 14 takes care of the semantic coverage property.

Proof of Theorem 25. For "$\supseteq$", observe that $\mathsf{CT} \supseteq \mathsf{CT}_N$ whence $\mathsf{OC} = \langle \mathsf{CT} \rangle \supseteq \langle \mathsf{CT}_N \rangle = \mathsf{OC}_N$ by Lemma 5 and the definition of OC_N. For "$\subseteq$", we prove that $c \in \mathsf{CT}$ and $(s,t) \in \langle c \rangle$ implies $(s,t) \in \langle \mathsf{CT}_N \rangle$. We do so by induction on c, ordered by the reduction order $>$. If c is in Q-normal form then $c \in \mathsf{CT}_N$, and so $\langle c \rangle \subseteq \langle \mathsf{CT}_N \rangle$. Now suppose that c admits a Q rewrite step. Then by semantic coverage there is $c' \in \mathsf{CT}$ such that both $c \to_Q c'$ and $(s,t) \in \langle c' \rangle$. From $c > c'$, the claim follows by inductive hypothesis for c'. ◀

Acknowledgements: I am indebted to Dieter Hofbauer and Johannes Waldmann for their plentiful feedback. Hans Zantema† encouraged me to study looping derivations. An anonymous referee provided valuable advice.

References

1 Nachum Dershowitz. Termination of linear rewriting systems. In Shimon Even and Oded Kariv, editors, *Automata, Languages and Programming, 8th Colloquium, Acre (Akko), Israel, July 13-17, 1981, Proceedings*, volume 115 of *LNCS*, pages 448–458. Springer, 1981.

2 Alfons Geser, Dieter Hofbauer, and Johannes Waldmann. Match-bounded string rewriting systems. *Appl. Algebra Eng. Commun. Comput.*, 15(3-4):149–171, 2004.

3 Alfons Geser, Dieter Hofbauer, and Johannes Waldmann. Sparse Tiling Through Overlap Closures for Termination of String Rewriting. In Herman Geuvers, editor, *4th International Conference on Formal Structures for Computation and Deduction (FSCD 2019)*, volume 131 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 21:1–21:21, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

4 Alfons Geser and Hans Zantema. Non-looping string rewriting. *Theoret. Informatics Appl.*, 33(3):279–301, 1999.

5 John V. Guttag, Deepak Kapur, and David R. Musser. On proving uniform termination and restricted termination of rewriting systems. *SIAM J. Comput.*, 12(1):189–214, 1983.

6 Miki Hermann. *Divergence des systèmes de réécriture et schématisation des ensembles infinis de termes*. Habilitation, Université de Nancy, France, March 1994.

7 Dallas S. Lankford and D. R. Musser. A finite termination criterion. Technical report, Information Sciences Institute, Univ. of Southern California, Marina-del-Rey, CA, 1978.