Off with the head: termination provers and the word problem for 1-relation monoids

Reinis Cirpons
University of St Andrews, United Kingdom

James D. Mitchell
University of St Andrews, United Kingdom

Finn L. Smith
University of St Andrews, United Kingdom

Abstract

In this article we discuss the application of termination provers to the decidability of the word problem for 1-relation monoids. In particular, we describe how Adian's algorithm $\mathfrak{A}$ for a particular left cycle-free 2-generated 1-relation monoid M can be used to produce a string rewriting system whose termination implies that the word problem for M is decidable. Such monoids are among the only cases of 1-relation monoids where it is not known whether or not the word problem is decidable. Our findings show that this new class of SRS is not only theoretically significant, but is also challenging for existing termination provers.

2012 ACM Subject Classification Mathematics of computing → Discrete mathematics; Mathematics of computing → Combinatorics on words; Mathematics of computing → Mathematical software; Security and privacy → Logic and verification

Keywords and phrases termination of string rewriting, word problem in monoids, one-relation monoid

Funding *Reinis Cirpons*: This work was supported by the Engineering and Physical Sciences Research Council grant number EP/V520123/1

Acknowledgements We would like to thank Johannes Waldmann and Dieter Hofbauer for assistance with the verification of the more difficult SRS termination instances arising from our research, as well as their encouragement, which lead to this short article. We also thank our collaborators on our wider 1-relation monoid projects: Florent Hivert, Assia Mahboubi, and Guillaume Melquiond.

1 Introduction

The decidability of the word problem for 1-relation monoids (WP1M) has been one of the most prominent open problems in combinatorial algebra for at least the past 100 years.

If A is a non-empty set, then the set A^* consists of the finite (possibly empty) sequences of elements of A, we write ε for the empty sequence and $A^+ = A^* \setminus \{\varepsilon\}$. The set A is called an *alphabet*, its elements are *letters*, and elements of A^* are *words*. The set A^* is a monoid (called the *free monoid*) when endowed with the binary operation of juxtaposition. A *monoid presentation* is a pair $\langle A \mid R\rangle$ for some non-empty set A and some set $R \subseteq A^* \times A^*$. For example, $\langle a, b \mid baaba = a\rangle$ is a monoid presentation (omitting superfluous brackets). Formally, a presentation *defines* the quotient monoid of A^* by the least congruence containing R. Informally, we may think of such a monoid M as consisting of words in A where two words $u, v \in A^*$ represent the same element of M, written $u \leftrightarrow_R^* v$, if there is a sequence of relations in R transforming u to v. For example, in the presentation given above

$$baab \leftrightarrow_R^* ba(baaba)b = ba(baab)ab \leftrightarrow_R^* \cdots \leftrightarrow_R^* (ba)^k baab(ab)^k, \quad k \geq 0.$$

If A and R are both finite, then the monoid defined by $\langle A \mid R\rangle$ is *finitely presented*.

The *word problem* for a monoid M defined by a presentation $\langle A \mid R\rangle$ is the decision problem with input $u, v \in A^*$, that outputs "yes" if the words u and v represent the same element in M and "no" if they do not. The word problem for monoids is undecidable in general. The literature is replete with the search for minimal examples, in one sense or another, of finitely presented monoids with undecidable word problem. If "minimal" means "with the fewest possible relations", then the best example to date is that given in [7] of a 2-generator and 3-relation presentation defining a monoid with undecidable word problem. On the other hand, the word problem for groups defined by (group) presentations with 1 relation were shown to be decidable by Magnus [6]. Although intensively studied for many decades, the question of whether or not there exists a monoid defined by a 1- or 2-relation presentation with undecidable word problem is open.

Recently there has been something of a renaissance in interest in the decidability of the WP1M; see [9] and the references therein for more details. Despite, or perhaps precisely due to, the wide variety of results proving decidability of the WP1M for various subclasses of 1-relation monoid presentations, there is no simple answer to the seemingly obvious question:

> *what is the smallest known instance of a* 1*-relation monoid presentation where the decidability of its word problem is unknown?*

This question motivated the authors of the current paper to explore to what extent, and by what means, the word problem for 1-relation monoids can be solved using existing computational tools and known mathematical results. In order to make our findings easily accessible and reproducible, we are in the process of collating our computational findings into a database containing formal ROCQ [12] proofs of decidability, which we intend to distribute as part of an accompanying website, the "Online Encyclopedia of 1-relation Monoids".

Adian and Oganesian [1] proved that the word problem is decidable for all 1-relation monoids if it is decidable for all presentations of the form:

$$\langle a, b \mid bua = ava\rangle \quad \text{or} \quad \langle a, b \mid bua = a\rangle \tag{1}$$

for some $u, v \in \{a, b\}^*$. In particular, given any 1-relation monoid M it is straightforward to construct a presentation of the form given in (1) defining a monoid M' with the property that if the word problem is decidable for M', then it is decidable for M also. So, from this point on we will restrict our attention to presentations of the form given in (1).

An overwhelming majority of instances of the WP1M that we have considered were readily tackled by known mathematical results or the Knuth-Bendix algorithm [5]. For other, more difficult instances, however, we were able to reduce the WP1M instance to an instance of the string rewriting system (SRS) termination problem, some of which we could solve using termination provers such as matchbox [13] and MultumNonMulta [4]. The aim of this paper is: to describe how we constructed SRS termination instances from the WP1M; and to provide some examples.

2 A rewriting system formulation of Adian's algorithm

A monoid defined by a 2-generator 1-relation presentation $\mathcal{P} = \langle a, b \mid u = v\rangle$ satisfies the *left cycle-free* condition if and only if $\mathcal{P}$ is in one of the forms given in (1), see [9, Section 2.2]. The *relations words* of $\mathcal{P}$ are the words u and v. As a consequence of the left cycle-free condition, every word $w \in \{a, b\}^+$ has a unique factorization $w = p_1 p_2 \cdots p_n h t$ where $n \geq 0$, each $p_i \in \{a, b\}^+$ is a proper and non-empty prefix of either u or v, $h \in \{u, v, \varepsilon\}$ and $t \in \{a, b\}^*$ are such that:

(i) for each i, if p' is a longer (possibly improper) prefix of the same relation word as p_i, then $p_1 \cdots p_{i-1}p'$ is not a prefix of w,

(ii) $h \neq \varepsilon$ if and only if $p_1 \cdots p_n \neq w$ and there does not exist a proper and non-empty prefix of a relation word p' such that $p_1 \cdots p_n p'$ is a prefix of w.

We call this factorization the *prefix decomposition* of w, the factor h is called the *head* of the decomposition. If $h = \varepsilon$ we call the prefix decomposition *headless*. We denote the prefix decomposition visually by separating the factors with bars $|$ and highlighting the head in bold, if it is present. For example, with respect to the presentation $\langle a, b \mid baabbaa = aba \rangle$, the word $babbaababaabbaa$ has the prefix decomposition $ba|b|baab|\mathbf{aba}|abbaa$ with head aba. The word $abbaabab$ has the prefix decomposition $ab|baab|ab$, which is headless.

Adian's algorithm $\mathfrak{A}$ for a left cycle-free presentation $\mathcal{P} = \langle a, b \mid u = v \rangle$ takes as input a letter $x \in \{a, b\}$ and a word $w \in \{a, b\}^+$ and proceeds as follows:

1. If the first letter of w is x or if the prefix decomposition of w is headless, then return w;
2. If the head of the prefix decomposition of w equals u, then replace the head by v in w and go to step 1;
3. If the head of the prefix decomposition of w equals v, then replace the head by u in w and go to step 1.

For example, a run of Adian's algorithm for the presentation $\langle a, b \mid baabbaa = aba \rangle$ with input $x = a$, $w = bbaabbabaababa$ produces the following sequence of prefix decompositions:

$$b|baabba|baab|\mathbf{aba} \to b|baabba|\mathbf{baabbaa}|bbaa \to b|\mathbf{baabbaa}|babbaa \to ba|ba|ba|b|baa$$

hence the $\mathfrak{A}(a, bbaabbabaababa) = bababab baa$.

Note that Adian's algorithm does not necessarily terminate on all inputs, e.g. in the monoid given by the presentation $\langle a, b \mid baabbaa = a \rangle$, running $\mathfrak{A}(a, bbaaa)$ results in the following sequence of rewrites:

$$b|baa|\mathbf{a} \to b|baab|\mathbf{a}|abbaa \to b|\mathbf{baabbaa}|bbaaabbaa \to ba|b|baa|\mathbf{a}|bbaa \to \cdots$$

The decomposition $ba|b|baa|\mathbf{a}|bbaa$ contains the decomposition $|b|baa|\mathbf{a}|$ of the initial word, hence $\mathfrak{A}(a, bbaaa)$ will not terminate. In fact, understanding termination of Adian's algorithm leads to a solution of the word problem for the monoid defined by the underlying presentation.

▶ **Theorem 1** (c.f. [9, Theorem 4.3]). *Let $\mathcal{P}$ be a 2-generated 1-relation left cycle-free presentation. If there is an algorithm which, given $x \in \{a, b\}$ and $w \in \{a, b\}^+$, decides whether or not $\mathfrak{A}(x, w)$ terminates, then the word problem for the monoid defined by $\mathcal{P}$ is decidable.*

When Adian's algorithm terminates on all inputs we get Corollary 2, which has been used in several articles to prove the decidability of the word problem for subclasses of 1-relation monoids, see [9, Section 4.3].

▶ **Corollary 2.** *Let $\mathcal{P}$ be a 2-generated 1-relation left cycle-free presentation. If Adian's algorithm terminates on all inputs, then the word problem in the monoid defined by $\mathcal{P}$ is decidable.*

The analysis of termination of Adian's algorithm is complicated somewhat by its description. Despite being written as a general algorithm, it only consists of repeated rewrites on a word depending on the prefix decomposition. Furthermore, the process of prefix decomposition itself can be formulated as a rewriting system, which produces an ever longer factorization until it either finds a head or factorizes the whole word. In order to broaden the termination proving methods available to us, we use these observations to produce, for every cycle-free presentation $\mathcal{P}$, a string rewriting system $\mathcal{A}_{\mathcal{P}}$ for which Theorem 3 holds.

▶ **Theorem 3.** *Let $\mathcal{P}$ be a left cycle-free presentation. Then $\mathcal{A}_{\mathcal{P}}$ is terminating if and only if $\mathfrak{A}$ terminates on all inputs.*

We will only describe our construction as it applies to 2-generated 1-relation left cycle-free presentation, the general construction which applies for all left cycle-free presentation is given in Appendix A. A similar construction has been considered in [2].

▶ **Definition 4.** *Let $\mathcal{P} = \langle a, b \mid u = v\rangle$ be a left cycle-free presentation and let*

$$P = \{\alpha \in \{a,b\}^+ : \exists \beta \in \{a,b\}^* \ s.t.\ \alpha\beta \in \{u,v\}\}$$

be the set of all non-empty prefixes of relation words (including u and v themselves). Let $B = \{q_p : p \in P\}$ be a set of symbols disjoint from $\{a,b\}$ and let

$$\mathcal{B} = \{q_p x \to q_{px} : p \in P, x \in \{a,b\} \ s.t.\ px \in P\}$$
$$\mathcal{C} = \{q_p x \to q_p q_x : p \in P, x \in \{a,b\} \ s.t.\ p \notin \{u,v\} \text{ and } px \notin P\}$$
$$\mathcal{D} = \{q_u \to v, q_v \to u\}.$$

Then Adian's string rewriting system associated to $\mathcal{P}$ *is the string rewriting system $\mathcal{A}_{\mathcal{P}}$ on $\{a,b\} \cup B$ given by the union $\mathcal{A}_{\mathcal{P}} = \mathcal{B} \cup \mathcal{C} \cup \mathcal{D}$.*

The proof of Theorem 3 is rather technical and we won't reproduce it here in full, but we give some indication of its correctness. Let $w \in \{a,b\}^+$, $x \in \{a,b\}$ and $w' \in \{a,b\}^*$ be such that $w = xw'$. We claim that the rewriting system consisting of $\mathcal{B} \cup \mathcal{C}$ when applied to $q_x w'$ computes the prefix decomposition of w. Indeed, after every application, the word will be of the form $q_{p_1} q_{p_2} \dots q_{p_k} t$ for some prefixes $p_1, \dots, p_k \in P$ and $t \in A^*$ such that $w = p_1 \cdots p_k t$. Rules in $\mathcal{B}$ can be used to extend the rightmost prefix p_k in the factorization, if this leads to a longer prefix, including either relation word u and v. Rules in $\mathcal{B}$ apply if the rightmost prefix p_k is not a whole relation word and the letter following p_k does not extend it to a longer prefix. In this case, a new prefix starting at the following letter is added to the factorization. If $p_k \in \{u,v\}$, then no rewrite rule from $\mathcal{B} \cup \mathcal{C}$ applies and p_k is the head of the decomposition. Rules in $\mathcal{D}$ simulate the substitution of the head performed by Adian's algorithm. Let $y \in \{a,b\} \setminus \{x\}$. Then it follows that the output of $\mathfrak{A}(y,w)$ terminates if and only if $\mathcal{A}_{\mathcal{P}}$ terminates when applied to $q_x w'$, and the result of $\mathfrak{A}(y,w)$ can be recovered from the resulting word. This is justification for the forward implication of Theorem 3. The reverse implication follows with some extra work from the fact that the rules in $\mathcal{A}_{\mathcal{P}}$ all commute.

▶ **Example 5.** Let $\mathcal{P} = \langle a, b \mid baabaa = aba\rangle$. Then the associated Adian's SRS is $\mathcal{A}_{\mathcal{P}}$ is as follows:

$$\begin{array}{cccc} q_a a \to q_a q_a, & q_a b \to q_{ab}, & q_{ab} a \to q_{aba}, & q_{ab} b \to q_{ab} q_b, \\ q_b a \to q_{ba}, & q_b b \to q_b q_b, & q_{ba} a \to q_{baa}, & q_{ba} b \to q_{ba} q_b, \\ q_{baa} a \to q_{baa} q_a, & q_{baa} b \to q_{baab}, & q_{baab} a \to q_{baaba}, & q_{baab} b \to q_{baab} q_b, \\ q_{baaba} a \to q_{baabaa}, & q_{baaba} b \to q_{baaba} q_b, & q_{aba} \to baabaa, & q_{baabaa} \to aba. \end{array}$$

The SRS $\mathcal{A}_{\mathcal{P}}$ is terminating, which can be established e.g. with the help of the matchbox termination prover. It follows that the monoid defined by the presentation $\mathcal{P} = \langle a, b \mid baabaa = aba\rangle$ has decidable word problem.

Method	Count
Known mathematical results	222786
Knuth-Bendix	37624
Knuth-Bendix backtrack	502
Adian's SRS terminates (certified)	460
Adian's SRS terminates (no certificate)	135
Unsolved (Adian's SRS non-terminating)	75
Unsolved (Other)	50
Total	261632

Table 1 Distribution of solutions to the word problem by the method used for monoids defined by a left cycle-free 2-generated 1-relation presentation $\mathcal{P} = \langle a, b \mid u = v \rangle$, where $|u|, |v| \leq 10$. See Section 3 for more details.

3 Results

We attempted to solve the word problem for all monoids defined by a left cycle-free 2-generated 1-relation presentation $\mathcal{P} = \langle a, b \mid u = v \rangle$, where $|u|, |v| \leq 10$. The only previous published attempt at doing so appears in [10] for presentations with $|u|, |v| \leq 6$. The results of our efforts are collected in Table 1. The rows of the table are as follows:

- "Known mathematical results" refers to presentations which are resolved by results described in [9, Section 2, Section 4.3 and Section 5.3] and generally involves criteria on the presentation which can be checked in polynomial time.
- "Knuth-Bendix" refers to presentations for which we could find complete rewriting systems using the Knuth-Bendix algorithm as it is implemented in libsemigroups [8]. The rewriting systems were obtained by exhaustively checking all possible shortlex orderings, as well as subword substitutions via Tietze transformations of depth up to 5.
- "Knuth-Bendix backtrack" refers to presentations for which complete rewriting systems could be found using a variation of the Knuth-Bendix algorithm which explores all possible word orderings in a backtracking fashion, akin to [15, 14]. This process produces a locally confluent SRS, whose termination we prove using the matchbox termination prover.
- "Adian's SRS terminates (certified)" refers to presentations for which a proof and CPF [11] certificate of termination of Adian's SRS was found using matchbox.
- "Adian's SRS terminates (no certificate)" refers to presentations for which a proof of termination of Adian's SRS was found using either matchbox or MultumNonMulta, but no CPF certificate of termination could be produced. This is mainly because RFC-based [3] proof methods cannot be certified at the moment.
- "Unsolved (Adian's SRS non-terminating)" refers to presentations for which we did not find a solution to the word problem, but for which a proof of non-termination of Adian's SRS was found by either matchbox or MultumNonMulta.
- "Unsolved (Other)" refers to presentations for which we did not find a solution and we do not know whether Adian's SRS terminates on all inputs.

Our results highlight the utility of termination provers for solving the WP1M and the difficulty of proving and certifying termination of the Adian's SRS instances arising from Adian's algorithm. We intend to make the SRS instances arising from Adian's algorithm publicly available by adding them to the Termination Problems Data Base (`https://termination-portal.org/wiki/TPDB`).

References

1 S. I. Adian and G. U. Oganesian. On the word and divisibility problems for semigroups with one relation. *Math. Notes*, 41:235–240, 1987. [Mat. Zametki 41, 3 (1987)].

2 Janet Brugh Bouwsma. *Semigroups presented by a single relation.* PhD thesis, The Pennsylvania State University, 1993.

3 Alfons Geser, Dieter Hofbauer, and Johannes Waldmann. Match-bounded string rewriting systems. *Applicable Algebra in Engineering, Communication and Computing*, 15(3–4):149–171, October 2004. URL: `http://dx.doi.org/10.1007/s00200-004-0162-8`, `doi:10.1007/s00200-004-0162-8`.

4 Dieter Hofbauer. System description: MultumNonMulta. In Aart Middeldorp and René Thiemann, editors, *WST*, page 90, 2016.

5 D. E. Knuth and P. B. Bendix. *Simple Word Problems in Universal Algebras*, pages 342–376. Springer Berlin Heidelberg, Berlin, Heidelberg, 1983. `doi:10.1007/978-3-642-81955-1_23`.

6 W. Magnus. Das Identitätsproblem für Gruppen mit einer definierenden Relation. *Mathematische Annalen*, 106(1):295–307, December 1932. URL: `http://dx.doi.org/10.1007/BF01455888`, `doi:10.1007/bf01455888`.

7 Yu. V. Matiyasevich. Simple examples of undecidable associative calculi. *Sov. Math., Dokl.*, 8:555–557, 1967.

8 James Mitchell, Michael Young, Finn Smith, Reinis Cirpons, Florent Hivert, Jerry James, Nicolas M. Thiéry, Joe Edwards, Wilf Wilson, Dima Pasechnik, Isuru Fernando, Jan Engelhardt, Luke Elliott, and Max Horn. libsemigroups/libsemigroups: 2.7.0, 2023. URL: `https://zenodo.org/record/7761055`, `doi:10.5281/ZENODO.7761055`.

9 Carl-Fredrik Nyberg-Brodda. The word problem for one-relation monoids: a survey. *Semigroup Forum*, 103(2):297–355, August 2021. `doi:10.1007/s00233-021-10216-8`.

10 John Pedersen. Morphocompletion for one-relation monoids. In Nachum Dershowitz, editor, *Rewriting Techniques and Applications*, pages 574–578, Berlin, Heidelberg, 1989. Springer Berlin Heidelberg.

11 Christian Sternagel and René Thiemann. The certification problem format. In Christoph Benzmüller and Bruno Woltzenlogel Paleo, editors, Proceedings Eleventh Workshop on *User Interfaces for Theorem Provers,* Vienna, Austria, 17th July 2014, volume 167 of *Electronic Proceedings in Theoretical Computer Science*, pages 61–72. Open Publishing Association, 2014. `doi:10.4204/EPTCS.167.8`.

12 The Rocq Development Team. The rocq prover, April 2025. `doi:10.5281/zenodo.15149629`.

13 Johannes Waldmann. Matchbox: A tool for match-bounded string rewriting. In Vincent van Oostrom, editor, *Rewriting Techniques and Applications*, pages 85–94, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

14 Ian Wehrman, Aaron Stump, and Edwin Westbrook. Slothrop: Knuth-Bendix Completion with a Modern Termination Checker. In Frank Pfenning, editor, *Term Rewriting and Applications*, Lecture Notes in Computer Science, pages 287–296, Berlin, Heidelberg, 2006. Springer. `doi:10.1007/11805618_22`.

15 Sarah Winkler, Haruhiko Sato, Aart Middeldorp, and Masahito Kurihara. Multi-Completion with Termination Tools. *Journal of Automated Reasoning*, 50(3):317–354, March 2013. `doi:10.1007/s10817-012-9249-2`.

A The construction for general left cycle-free presentations

During our experimentation we found that making certain substitutions to a left cycle-free presentation which increase then number of relations, but keep the presentation left cycle-free could sometimes lead to string rewriting systems which were easier to prove termination of. Similar phenomena for the Knuth-Bendix procedure are well known. In order to apply such transformations, however, we need to use a slightly more general construction than the one given in Definition 4. We do so in this appendix.

For a set X we define the symmetric closure of a relation $R \subseteq X \times X$ to be $R^{\mathrm{Sym}} = R \cup \{(\beta, \alpha) : (\alpha, \beta) \in R\}$. For a partial function f, we write $f(x) = \bot$ to indicate that f is not defined on input x. For a non-empty word $w \in A^+$ we will denote the first letter of w by $\mathtt{first}(w) \in A$ and write $\mathtt{tail}(w)$ for the remainder after removing the first letter, so that $w = \mathtt{first}(w)\mathtt{tail}(w)$. Let M be a monoid defined by the presentation $\mathcal{P} = \langle A \mid R \rangle$. We assume that that $\alpha, \beta \in A^+$ are non-empty for all relations $(\alpha, \beta) \in R$.

We define $\mathcal{L}(M)$, the *left graph* of M with respect to $\mathcal{P}$, to be the undirected multigraph with vertex set A and an edge between letters $a, b \in A$ for *every* occurrence $(\alpha, \beta) \in R$ such that $a = \mathtt{first}(\alpha)$ and $b = \mathtt{first}(\beta)$. We say that the monoid M is *left cycle-free* if $\mathcal{L}(M)$ is acyclic with respect to some presentation $\mathcal{P}$.

Note that in a left cycle-free monoid, for every pair of adjacent letters a, b in $\mathcal{L}(M)$ we can find a unique relation $(\alpha, \beta) \in R^{\mathrm{Sym}}$ such that $a = \mathtt{first}(\alpha)$ and $b = \mathtt{first}(\beta)$. With this in mind, we define the partial functions $\sigma, \tau : A \times A \to A^+$ as follows. Given $a, b \in A$,

- if $a = b$ or a and b are not not connected in the graph $\mathcal{L}(M)$, then let $\sigma(a, b) = \tau(a, b) = \bot$;
- if $a \neq b$ are adjacent in $\mathcal{L}(M)$ then let $\sigma(a, b) = \alpha$ and $\tau(a, b) = \beta$ where $(\alpha, \beta) \in R^{\mathrm{Sym}}$ is the unique relation such that $a = \mathtt{first}(\alpha)$ and $b = \mathtt{first}(\beta)$;
- if $a \neq b$ are connected but not adjacent in $\mathcal{L}(M)$, then let $\sigma(a, b) = \sigma(a, v_2)$ and $\tau(a, b) = \tau(a, v_2)$, where $a = v_1, v_2, \ldots, v_n = b$ is the unique path connecting a, b in $\mathcal{L}(M)$. Note that $v_2 \neq b$ as a, b are not adjacent.

It follows from the construction that $(\sigma(a, b), \tau(a, b)) \in R^{\mathrm{Sym}}$ is a relation whenever $\sigma(a, b)$ and $\tau(a, b)$ are defined. Note also that $\mathtt{first}(\sigma(a, b)) = a$ whenever $\sigma(a, b) \neq \bot$.

▶ **Definition 6.** *Let $\mathcal{P} = \langle A \,|\, R \rangle$ be a left cycle-free presentation defining a monoid M. For every pair $(a, b) \in A \times A$ let $P_{(a,b)} = \{\alpha \in A^+ : \exists \beta \in A^* \text{ s.t. } \alpha\beta = \sigma(a, b)\}$ be the set of non-empty prefixes of $\sigma(a, b)$. Let B be set disjoint from A given by*

$$B = \{q_{(a,b)} : \forall a, b \in A\} \cup \bigcup_{(a,b) \in A \times A} \{q_{(a,\alpha)} : \alpha \in P_{(b,a)}\},$$

and let

$$\begin{aligned}
\mathcal{T}_M &= \{q_{(a,a)} \to a : a \in A\} \\
\mathcal{B}_M &= \{q_{(a,\gamma)} b \to q_{(a,\gamma b)} : a, b \in A, \gamma \in A^+, \gamma, \gamma b \in P_{(\mathit{first}(\gamma), a)}\} \\
\mathcal{C}_M &= \{q_{(a,\gamma)} b \to q_{(a,\gamma)} q_{(c,b)} : a, b, c \in A, \gamma \in A^+, \gamma, \gamma c \in P_{(\mathit{first}(\gamma), a)}, \gamma b \notin P_{(\mathit{first}(\gamma), a)}\} \\
\mathcal{D}_M &= \{q_{(a,\sigma(d,a))} \to q_{(a,\mathit{first}(\tau(d,a)))} \mathit{tail}(\tau(d, a)) : a, d \in A, \delta \in A^+, \sigma(d, a) \neq \bot\}.
\end{aligned}$$

The Adian rewriting system *is the rewriting system on $C = A \cup B$ given by the union $\mathcal{A}_{\mathcal{P}} = \mathcal{T} \cup \mathcal{B} \cup \mathcal{C} \cup \mathcal{D}$.*

Theorem 3 holds for $\mathcal{A}_{\mathcal{P}}$ given by Definition 6.