CPSC-354 Report

Junho Yi
Chapman University

September 14, 2025

Abstract

Contents

1 Introduction

2 Week by Week

2.1 Week 1

2.1.1 Notes and Exploration

Place holder for notes

In abstract rewriting, an object is in normal form if it cannot be rewritten any further, i.e. it is irreducible

Confluence system: in a system the result eventually converges into the same answer.

Termination: Means the system stops at some point.

Decidability:

church turing thesis:

Abstract rewriting system(ARS): mathematically the same as a directed graph A is a set of "strings" (can be anything) R is the relation

so in the MIU puzzle, A is M I U, the strings we use then R is the rules we are given. ie: (Mx,Mxx)|x e A U...

2.1.2 Homework 1

This week's HW is regarding the MU puzzle, and its relevance and application to formal systems. Here we use the MU puzzle to practice and familiarize ourselves with staying within the confines of a formal system. We are given 4 rules/restrictions, which is referred to as the "Requirement of Formality."

Our formal system consists of these 4 rules:

1. **RULE I**: If you possess a string whose last letter is I, you can add on a U at the end.
2. **RULE II**: Suppose you have Mx. Then you may add Mxx to your collection.
3. **RULE III**: If III occurs in one of the strings in your collection, you may make a new string with U in place of III.
4. **RULE IV**: If UU occurs inside one of your strings, you can drop it.

With these four rules in mind, we have one objective: stay within the rules and produce MU from MI.

As I worked through the rules, I logically deduced these points in this order:

1. When applying RULE II, if I exists somewhere in the string, the parity of I becomes even until RULE III is applied again.
2. When applying RULE III, the I's (which are even, if RULE III applies) swap parity, i.e. even $\mapsto$ odd.
3. The lowest continuous string of I's where RULE III can be applied is $IIII$ (four I's).
4. Because RULE III is the only way to reduce the number of I's, and it is only possible to apply RULE III if there is a minimum of four continuous I's (due to RULE II) and an even parity of I's, using RULE III to reduce the amount of I's will always result in a remainder (a leftover I).
5. Therefore, you can never get rid of I's fully with RULE III, or any other RULE usable by us without modifications of the rules.

From the above observations, we can see that there is no way to completely reduce the number of I's into zero with the given rules. This is my personal analysis of the MU puzzle; below is the "correct" analysis of the MU puzzle.

Proof (invariant mod 3). Let n be the number of I's in the current string. Then:

$$\text{Rule I: } n \mapsto n, \quad \text{Rule II: } n \mapsto 2n, \quad \text{Rule III: } n \mapsto n-3, \quad \text{Rule IV: } n \mapsto n.$$

Hence n mod 3 is preserved by Rules I, III, IV, and toggles between 1 and 2 under Rule II. Initially, MI has $n = 1 \equiv 1 \pmod 3$. No sequence of the above operations can yield $n \equiv 0 \pmod 3$. But MU has 0 I's, i.e. $n = 0 \equiv 0 \pmod 3$. Therefore MU is not derivable from MI. □

2.2 Week 2, Rewriting theory

2.2.1 Notes and Exploration

placeholder for notes

2.2.2 HW 2

1. $A = \{\}$
2. $A = \{a\}, \quad R = \{\}$
3. $A = \{a\}, \quad R = \{(a, a)\}$
4. $A = \{a, b, c\}, \quad R = \{(a, b), (a, c)\}$
5. $A = \{a, b\}, \quad R = \{(a, a), (a, b)\}$
6. $A = \{a, b, c\}, \quad R = \{(a, b), (b, b), (a, c)\}$
7. $A = \{a, b, c\}, \quad R = \{(a, b), (b, b), (a, c), (c, c)\}$

Homework: Draw a picture for each of the ARSs above. Are the ARSs terminating? Are they confluent? Do they have unique normal forms?

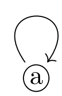
1.

Terminating ✓ | Confluent ✓ | Unique normal forms ✓

2.

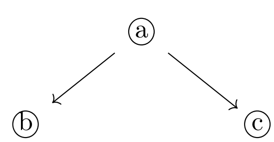
Terminating ✓ | Confluent ✓ | Unique normal forms ✓

3.



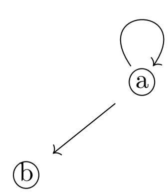
Terminating × | Confluent ✓ | Unique normal forms ×

4.



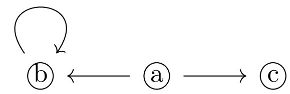
Terminating ✓ | Confluent × | Unique normal forms ×

5.



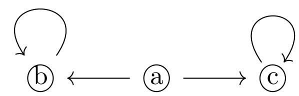
Terminating × | Confluent ✓ | Unique normal forms ✓

6.



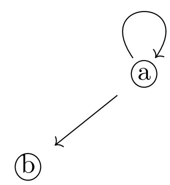
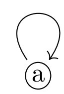
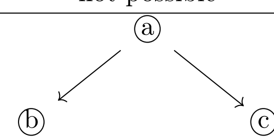
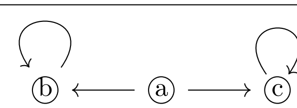
Terminating × | Confluent × | Unique normal forms ×

7.



Terminating × | Confluent × | Unique normal forms ×

Homework: Try to find an example of an ARS for each of the possible 8 combinations. Draw pictures of these examples.

confluent	**terminating**	**has unique normal forms**	**example**
True	True	True	◯ , ⓐ
True	True	False	not possible
True	False	True	
True	False	False	
False	True	True	not possible
False	True	False	
False	False	True	not possible
False	False	False	

2.3 Week 3, String rewriting

2.3.1 HW 3

For this week's homework, we are tasked with answering the questions on Exercises 5 and 5b.

Exercise 5:

- Reduce some example strings such as `abba` and `bababa`.

Reduce abba	**Reduce** bababa
abba	bababa
aba	ababa
aa	aaba
ε	aaa
	a

- **Why is the ARS not terminating?**

  The ARS does not terminate because the system is not strongly normalizing. Rules 1 and 2 are bi-directional swap rules, thus there exists an infinite reduction.

- **Find two strings that are not equivalent. How many non-equivalent strings can you find?**

  a and ε. If the question is asking for "how many strings not equivalent to ε," then we can find an infinite number of non-equivalent strings (i.e., any string with odd number of a's vs even number of a's).

- **How many equivalence classes does $\overset{*}{\longleftrightarrow}$ have? Can you describe them in a nice way? What are the normal forms?**

  The normal forms are a and ε, the two equivalence classes are strings with even number of a's and odd number of a's.

- **Can you modify the ARS so that it becomes terminating without changing its equivalence classes?**

  We have to get rid of either rule 1 or 2, to make the ARS strongly normalizing.

- **Write down a question or two about strings that can be answered using the ARS. Think about whether this amounts to giving a semantics to the ARS.**

  [Hint: The best answers are likely to involve a complete invariant.]

  Remark: A characterization of the equivalence classes that mentions the reduction relation is not interesting.

  Question: "Does this string have an even or odd number of a's? Which equivalence class does even/odd a's belong to?"

Excercise 5b:

- Reduce some example strings such as abba and bababa.

Reduce abba	**Reduce** bababa
abba	bababa
aba	ababa
aa	aaba
a	aaa
	aa
	a

- **Why is the ARS not terminating?**

  The ARS does not terminate because the system is not strongly normalizing. Rules 1 and 2 are bi-directional swap rules, thus there exists an infinite reduction.

- **Find two strings that are not equivalent. How many non-equivalent strings can you find?**

  a and ε. If the question is asking for "how many strings not equivalent to ε," then there are infinitely many (e.g., any string containing at least one a).

- **How many equivalence classes does $\stackrel{*}{\longleftrightarrow}$ have? Can you describe them in a nice way? What are the normal forms?**

  The normal forms are `a` and ε. The two equivalence classes are: strings with *no* `a`'s (equivalent to ε), and strings with *at least one* `a` (equivalent to `a`).

- **Can you modify the ARS so that it becomes terminating without changing its equivalence classes?**

  We have to get rid of either rule 1 or 2, to make the ARS strongly normalizing.

- **Write down a question or two about strings that can be answered using the ARS. Think about whether this amounts to giving a semantics to the ARS.**

  [Hint: The best answers are likely to involve a complete invariant.]

  Remark: A characterization of the equivalence classes that mentions the reduction relation is not interesting.

  Question: "Does this string have an `a` or not? Which equivalence class do 'no `a`' and 'some `a`' belong to?"

3 Essay

4 Evidence of Participation

5 Conclusion

References

[BLA] Author, Title, Publisher, Year.