CPSC-354 Report

Ethan Tapia
Chapman University

August 31, 2025

Abstract

Contents

1 Introduction

2 Week by Week

2.1 Week 1

Lecture Summary

We introduced *formal systems* and worked with Hofstadter's MIU-system as a rule–based rewriting game. Alphabet: $\Sigma = \{M, I, U\}$. Axiom (start string): MI. Production rules:

(R1) If a string ends in I, append U: $xI \Rightarrow xIU$.

(R2) If a string is Mx, duplicate x: $Mx \Rightarrow Mxx$.

(R3) Replace any III by U: $xIIIy \Rightarrow xUy$.

(R4) Delete any UU: $xUUy \Rightarrow xy$.

Key idea: reason about *invariants* that rules preserve, instead of searching blindly through derivations.

Homework: The MU-puzzle

Definition 2.1 (I–count and residue)**.** For a string w, let $\#_I(w)$ be the number of I's in w, and define the residue

$$\varphi(w) \;=\; \#_I(w) \bmod 3 \in \{0, 1, 2\}.$$

Lemma 2.2 (Effect of each rule on $\#_I$)**.** *For any string w:*

1. ***(R1)*** *and* ***(R4)*** *do not change $\#_I$.*
2. ***(R2)*** *doubles the number of I's* after *the initial M, so φ is multiplied by 2 modulo 3.*
3. ***(R3)*** *decreases $\#_I$ by 3, so φ is unchanged.*

Proposition 2.3 (Invariant modulo 3)**.** *Every string derivable from MI has $\varphi \in \{1, 2\}$. In particular, no derivable string has $\varphi = 0$.*

Proof. We use induction on the length of a derivation from MI.

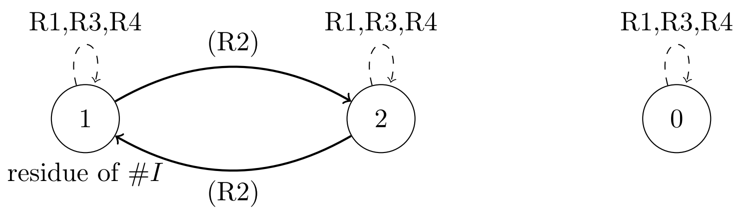
Base. $\varphi(MI) = 1$.

Step. Assume $\varphi \in \{1, 2\}$ for some derivable w. By Lemma 2.2, rules (R1), (R3), and (R4) keep φ unchanged, and rule (R2) maps $1 \leftrightarrow 2$ modulo 3. None of these operations yields 0 from a value in $\{1, 2\}$. Therefore the next string also has $\varphi \in \{1, 2\}$. ☐

Theorem 2.4 (MU is unreachable)**.** *MU cannot be derived from MI in the MIU-system.*

Proof. MU contains zero I's, hence $\varphi(MU) = 0$. By Proposition 2.3, every derivable string has residue 1 or 2. Thus MU is not derivable. ☐

Conclusion. Starting from MI we can toggle the residue $1 \leftrightarrow 2$ with (R2) and otherwise keep it fixed with (R1), (R3), (R4). We never reach residue 0, so no sequence of legal rule applications yields MU.



Question: If the MU-puzzle shows that some goals are unreachable due to invariants (like the mod-3 property of I's), how does this idea connect to undecidability in programming languages?

3 Essay

4 Evidence of Participation

5 Conclusion

References

[BLA] Author, Title, Publisher, Year.