Revisiting MU-puzzle. A case study in finite countermodels verification –

Alexei Lisitsa

Department of Computer Science, University of Liverpool

Abstract. In this paper we consider well-known MU puzzle from Goedel, Escher, Bach: An Eternal Golden Braid book (GEB) by D. Hofstadter, as an infinite state safety verification problem for string rewriting systems. We demonstrate fully automated solution using finite countermodels method (FCM). We highlight advantages of FCM method and compare it with alternatives methods using regular invariants.

It is commonly accepted that an inductive reasoning is an important part of commonsense reasoning []. Automation of inductive reasoning brings ultimate challenges of undecidability – even semi-decision procedures are not possible under very modest assumptions. In this paper we demonstrate conceptually simple but powerful technique originated in the research on verification of cryptographic protocols [] and more generally of parameterized and infinite state systems [1,2,3] can be applied in commonsense reasoning contexts. We start with well-known MU Puzzle introduced in book [4]

1 MIU system and MU puzzle

In his famous book *Goedel, Escher, Bach: An eternal Golden Braid, 1979*, Douglas Hofstadter introduced a simple formal system, named MIU-system, which operates on strings made of three symbols, M, I and U. The system consists of one axiom, that is MI and four derivation rules:

I. If xI is a theorem, so is xIU.
II. If Mx is theorem, so is Mxx.
III. In any theorem III can be replaced by U.
IV. UU can be dropped from any theorem.

In other words, MIU system is a *string rewriting system* with an initial string MI and the rewriting rules $R = \{xI \Rightarrow xIU; Mx \Rightarrow Mxx; xIIIy \Rightarrow xUy; xUUy \Rightarrow xy\}$. We denote the language generated by this rewriting system by L_{MIU}. From now on we use interchangeably expressions "a string S is a theorem of MIU system" and "string S belongs to the language L_{MIU}".

MU puzzle is a specific problem about MIU system, that is "Is MU a theorem of MIU system?" The problem is discussed at length in [4] and the answer is negative. It follows from simple necessary condition: "the number of I symbols in any string in L_{MIU} cannot be multiple of three". The authors of [5] show that this condition augmented with structural requirement that any MIU theorem should start with M followed by an arbitrary word in I's and U's is also sufficient, obtaining thereby a simple decision procedure for MIU theorems.[1]

We show here an alternative way to get an answer (with a proof) for MU puzzle automatically, from first principles and not assuming the knowledge of the decision procedure. First notice that there are infinitely many theorems in MIU, so the negative answer can not be obtained just by exhaustion of all derivable strings. It is essentially infinite state verification problem.

In order to deal with a problem automatically we formulate a natural theory T_{MIU} in first-order logic which encodes the rewriting process. The vocabulary of the T_{MIU} consists of one unary predicate symbol T binary functional symbol $*$ which we use in infix notation an three constants M, I and U. Intended meaning of T(x) is "x is a theorem of MIU" and $*$ denotes concatenation to be used to build strings out of constants.

The theory T_{MIU} consist the following axioms:

1. $(x * y) * z = x * (y * z)$ (associativity of concatenation);
2. $e * x = x$;
3. $x * e = x$;
4. $T(M * I)$ (MI is a theorem of MIU);
5. $T(x * I) \to T(x * I * U)$ (rule I of MIU);
6. $T(M * x) \to T(M * x * x)$ (rule II of MIU);
7. $T(x * I * I * I * y) \to T(x * U * y)$ (rule III of MIU)
8. $T(x * U * U * y) \to T(x * y)$ (rule IV of MIU)

Now we have a simple

Proposition 1. *If $S \in L_{MIU}$ then $T_{MIU} \vdash_{FO} T(t_S)$ where t_S is a term encoding of S; e.g. $t_{IUM} \equiv I * U * M$*

Proof. Straightforward induction on the derivation of S in MIU. Indeed $T(M * I) \equiv T(t_{MI})$ is an axiom of T_{MIU}, so the base of induction holds true: $T_{MIU} \vdash_{FO} T(t_{MI})$. Assume the proposition holds true for a string S in L_{MIU}, and S' is obtained from S by application of the rule I. Then we have: (1) $T_{MIU} \vdash T(t_S)$ by induction assumption;(2) $T_{MIU} \vdash_{FO} T(t_S) \to T(t_{S'})$ by axiom 3 and finally, (3) $T_{MIU} \vdash T(t_{S'}))$ by Modus Ponens applied to (2) and (3). The cases of S' obtained from S by rules $II - IV$ are considered similarly using axioms $4 - 6$. The step of induction is proven.

We have an immediate

Corollary 1. – *If $T(t_S)$ is not FO provable from T_{MIU}, that is $T_{MIU} \not\vdash_{FO} T(t_S)$ then $S \notin L_{MIU}$;*

[1] They also notice that Hofstadter was aware about the decision procedure, but never formally wrote a proof.

– *For any non-ground term $t(\bar{x})$ in vocabulary $\{*, M, I, U\}$ over the set of variables X, if $T_{MIU} \not\vdash_{FO} \exists\bar{x}T(t(\bar{x}))$ then none of S such that t_S is a ground instance of $t(\bar{x})$ belongs to L_{MIU}.*

Returning to MU puzzle it should be clear now that to answer its question negatively it is sufficient to find a countermodel for $T_{MIU} \to T(t_{MU})$, or, in other words, a model for $T_{MIU} \wedge \neg T(t_{MU})$. We delegate this problem to Mace4[], the automated finite model finder for first-order logic. The countermodel of size 3 is found in 0.05s [2]. The property is proven: MU is not a theorem of MIU system. On the face of it, we have a simple logical argument: should MU be a theorem of MIU the formula $T(t_{MU})$ would be provable from T_{MIU}; since we found a countermodel for $T_{MIU} \to T(t_{MU})$, this is impossible. This argument does not explain though "the reasons" for impossibility. To recover more detailed argument let us have a look at the generated countermodel.

The domain $\mathcal{M}$ of the model is the set $0, 1, 2$ the interpretations of constants M, I and U are $0, 0$ and 1, respectively. The interpretation $[*]$ of concatenation (semigroup) operation * is given by the table

```
[*]  0  1 2
     ------
   0 |2,0,1
   1 |0,1,2
   2 |1,2,0
```

The interpretation $[T]$ of unary predicate T includes elements $1, 2$ of the domain, meaning T is true on $1, 2$ and false on 0. Now we notice that the model provides with an interpretation $[t_S] \in \{0, 1, 2\}$ of any term t_S. The following property holds: for any theorem S of MIU the interpretation $[t_S]$ should be an element of $\{1, 2\} = [T]$ (as $\mathcal{M}$ is a model of T_{MIU} and by Proposition 1). Returning to MU puzzle, we have interpretation $t_{MU} = [M * U] = 0[*]1 = 0 \notin \{1, 2\} = [T]$. Therefore MU is not a theorem of MIU. In summary, the interpretation $[*]$ above defines the set of strings $L_{\mathcal{M}} = \{s \mid [t_s]_{\mathcal{M}} \in \{0, 1\}\}$ for which (1) $L_{MIU} \subseteq L_{\mathcal{M}}$; (2) $MU \notin L_{\mathcal{M}}$. Thus, $L_{\mathcal{M}}$ is an invariant separating the theorems of MIU system and the string in question, MU. It is easy to see also that the invariant is a *regular* language. Interestingly, $L_{\mathcal{M}} \neq L_{MIU}$ as, for example, $[M * M] = 2 \in [T]$ hence $MM \in L_{\mathcal{M}}$ but $MM \notin L_{MIU}$ by decision procedure of [5]. Applying our method to show $MM \notin L_{MIU}$ we formulate the formula to disprove: $T_{MIU} \to T(M * M)$. Mace4 finds a countermodel $L_{\mathcal{M}'}$ of size 2, with the domain $\{0, 1\}$, the interpretations of constants M, I and U as $1, 0$ and 0, respectively; the interpretation $[T]$ of $T = \{1\}$. the interpretation of * is given by the table

```
[*]  0  1
     ----
   0 |0,1
```

[2] tech spec. of system used

```
1 |1,0
```

The corresponding invariant $\{s \mid [t_s]_{\mathcal{M}'} = 1\}$ captures the "oddness" of M count in strings, which is sufficient to separate MM from L_{MIU}.

What about MMM? this is also non-theorem of MIU by the decision procedure, but neither of the above models $\mathcal{M}$ or $\mathcal{M}'$ defines an appropriate separator. The minimal countermodel $\mathcal{M}''$ for $T_{MIU} \to T(M * M * M)$ is as follows

```
[*]  0  1
       ----
     0 |0,1
     1 |1,0
```

The natural question appears as to whether by an appropriate choice of target "non-theorems" of MIU one can get a countermodel defining an exact invariant coinciding with L_{MIU}. We answer this question positively by introducing "disjunctive targets" formulas. At this point we cease to pretend that we don't know the decision procedure and rather use it to make a conscious choice of target non-theorems. After some trials we came up with the following disjunction of non-theorem targets:

$$\varphi_d \equiv \exists x T(M * M * x) \vee \exists x T(I * x) \vee \exists x T(U * x) \vee T(M * U)$$

Neither MU nor any of the ground instances of existential disjuncts are elements of L_{MIU} (by decision procedure). For the formula $T_{MIU} \to \varphi_d$ finite model finder Mace4 finds a minimal countermodel $\mathcal{M}''$ of size 7.

The domain of $\mathcal{M}''$ is the set $\{0, 1, 2, 3, 4, 5, 6\}$; the interpretations of the constants M,I and U are $1, 0$ and 2 respectively. The interpretation $[T]$ of T is $\{4, 5\}$ and $[*]$ is given by the following multiplication table.

```
        0 1 2 3 4 5 6
[*]     -------------
    0   |3,6,0,2,6,6,6
    1   |4,6,1,5,6,6,6
    2   |0,6,2,3,6,6,6
    3   |2,6,3,0,6,6,6
    4   |5,6,4,1,6,6,6
    5   |1,6,5,4,6,6,6
    6   |6,6,6,6,6,6,6
```

Proposition 2. *The invariant $L_{\mathcal{M}''}$ defined by the countermodel $\mathcal{M}''$ coincides with L_{MIU}, that is the interpretation of any term t_S belongs to the interpretation $[T]$ of T iff "S starts with symbol M, followed by an arbitrary word in symbols I and U with a number of I being not multiple of 3.*

Proof: Straightforward but tedious check. In fact, we can automate this check and reduce it to a disproving task.

Furthermore. we propose a procedure which would allow to generate

Proposition 3. *There is no single target formula $T(\tau)$ with a ground τ for which a minimal countermodel defines L_{MIU}.*

Proof By the decision procedure of [5] any non-theorem of MIU system is either (i) a word starting with I letter; or (ii) a word starting with U letter; or (iii) a word starting from M letter and having two or more M letters; or (iv) a word starting from M letter following by a word in I and U letters with multiplicity of I being multiple of 3. We consider all these cases in their turn.

(i) For the formula $T_{MIU} \to \exists x T(I * x)$ Mace4 model finder generates the following minimal countermodel

```
interpretation( 2, [number = 1,seconds = 0], [
    function(*(_,_), [
        0,0,
        1,1]),
    function(aI, [0]),
    function(aM, [1]),
    function(aU, [0]),
    relation(R(_), [0,1])]).
```

It follows[3] that for any ground instance τ of $I*x$ the above is a countermodel, and therefore the minimal countermodel for any such τ is no larger[4] than the above model.

(ii) For the formula $T_{MIU} \to \exists x T(U * x)$ Mace4 model finder generates the same minimal countermodel as presented above in (i). The same argument follows.

(iii) For the formula $T_{MIU} \to \exists x \exists y R(M * x * M * y)$ Mace4 generates the following minimal countermodel.

```
interpretation( 3, [number = 1,seconds = 0], [
    function(*(_,_), [
        0,1,2,
        1,2,2,
        2,2,2]),
    function(aI, [0]),
    function(aM, [1]),
    function(aU, [0]),
    relation(R(_), [0,1,0])]).
```

[3] Interestingly, here we can either rely on the assumption of the correctness of Mace4, or the statement can be checked manually by straightforward induction on the length of the ground instance

[4] not to forget to discuss minimality

It follows that for any ground instance τ of $M * x * M * y$ the above is a countermodel, and therefore the minimal countermodel for any such τ is no larger than the above model.

(iv) For the formula $T_{MIU} \to T(M*I*I*I*U)$ Mace4 model finder generates the following countermodel.

```
interpretation( 3, [number = 1,seconds = 0], [
    function(*(_,_), [
        2,0,1,
        0,1,2,
        1,2,0]),
    function(aI, [0]),
    function(aM, [0]),
    function(aU, [1]),
    function(e, [1]),
    relation(R(_), [0,1,1])]).
```

Proposition 4. *There are not two formulae $T(\tau_1)$ and $T(\tau_2)$ with ground τ_1 and τ_2 such that a minimal countermodel for $T_{MIU} \to T(\tau_1) \vee T(\tau_2)$ defines L_{MIU}*

Proposition 5. *Full characterization of possible countermodels for any of a non-theorem in MIU.*

1.1 Discussion

we have shown in this section how to solve MU puzzle by first-order theorem disproving (finite model finding) fully automatically and from the first principles. As far we are aware, no fully automatic solution of this puzzle has been presented in the literature so far. We have further shown that the known decision procedure can be re-interpreted in terms of a single finite countermodel. MU puzzle in a instance of an infinite state verification problem and as such it was used as a case study to illustrate the verification methods based on Counter Example Guided Refinement in [6]. The verification presented in [6] was not fully automated and required a creative step in the choice of invariants. The solution we presented here is an instance of the application of very general finite countermodel verification method from [2,3].

Questions:

- Is it possible to find quantifier-free target formula defining required invariant?
- If the set of reachable strings is regular (say for SemiThue, is it always possible to generate it by a finite set of target formulae?
- Can we present some procedure using proving/disproving which would lead to generation of a regular set of reachable strings, if such does exist?

– One may consider finding a regular separation set as yet another way to define regular languages, complimentary to finite automata, regular expressions, etc. This is parameterized by the class of considered rewriting systems and by the type of target formulae.
– Is it possible to have a rewriting system which would have a regular separator for every non-reachable strings, but not a single regular separator for all of them. What about condition that system itself has a regular/non-regular set of reachable strings?
– The concept of minimality depends on the partical order between the models. Two possible variants at least: (i) order by the cardinality of the base set of the model' (ii) order by language inclusion. The first is easier to get as a result of finite model finder work.

Acknowledgments

```
interpretation( 3, [number = 1,seconds = 0], [
    function(*(_,_), [
        2,0,1,
        0,1,2,
        1,2,0]),
    function(aI, [0]),
    function(aM, [0]),
    function(aU, [1]),
    relation(T(_), [0,1,1])]).
```

References

1. Lisitsa, A.: Reachability as derivability, finite countermodels and verification. In: Automated Technology for Verification and Analysis - 8th International Symposium, ATVA 2010, Singapore, September 21-24, 2010. Proceedings. (2010) 233–244
2. Lisitsa, A.: Finite models vs tree automata in safety verification. In: 23rd International Conference on Rewriting Techniques and Applications (RTA'12) , RTA 2012, May 28 - June 2, 2012, Nagoya, Japan. (2012) 225–239
3. Lisitsa, A.: Finite reasons for safety - parameterized verification by finite model finding. J. Autom. Reasoning **51**(4) (2013) 431–451
4. Hofstadter, D.R.: Godel, Escher, Bach: An Eternal Golden Braid. Basic Books, Inc., New York, NY, USA (1979)
5. Swanson, L., McEliece, R.J.: A simple decision procedure for hofstadter's miu-system. The Mathematical Intelligencer **10**(2) (1988) 48–49
6. Clarke, E.M., Fehnker, A., Han, Z., Krogh, B.H., Ouaknine, J., Stursberg, O., Theobald, M.: Abstraction and counterexample-guided refinement in model checking of hybrid systems. Int. J. Found. Comput. Sci. **14**(4) (2003) 583–604