Tackling Fibonacci words puzzles by finite countermodels

Alexei Lisitsa[1]

Department of Computer Science, The University of Liverpool
A.Lisitsa@csc.liv.ac.uk

Abstract. In the paper we present an automated solution to the puzzle on Fibonacci words using first-order theorem disproving by finite model finding.

1 Introduction

The verification of properties of infinite state systems in general is undecidable problem and the development of new verification techniques and methods will never stop. One of the promising directions for tackling infinite-state, or parameterized, verification is to apply the methods based on direct encoding of states and transitions of the systems of interest in classical *first order* logic in a way that computations of the system are faithfully modelled by the derivations in first-order logic. The verification of *safety* properties, that is non-reachability of *unsafe* states, then translated into the task of disproving first-order formulae, which can be then tackled by automated finite countermodel finding.

The safety verification technique using first-order encoding and finite model finding has been originated in earlier work on the verification of cryptographic protocols [14, 13, 2, 3] and the first, particularly simple and convincing presentation of such ideas can be found in a paper [13].

More recent work [6–8, 10, 11] has showed that there is nothing specifically "cryptographic" or "security-related" in such an approach to verification and it can be applied to the numerous and wider classes of infinite-state and parameterized verification tasks. Rather surprisingly it has turned out that Finite Countermodel verification method (FCM) not only has particularly simple and elegant theoretical foundations but is robust and very efficient in practice.

In this paper we present a small case study and apply FCM method to resolve automatically the puzzle(s) on Fibonacci words. We illustrate both the elegance and the power of the method and explain the meaning of produced proofs.

2 Preliminaries

We assume that the reader is familiar with the basics of first-order logic. In particular, we use without definitions the following concepts: first-order predicate logic, first-order models, interpretations of relational, functional and constant symbols, satisfaction $\models$ of a formula in a model, semantical consequence $\models$, deducibility (derivability) $\vdash$ in first-order logic. We denote interpretations by square brackets, so, for example, $[f]$ denotes an interpretation of a functional symbol f in a model. We also expect the reader not to be surprised by the existence of *complete* finite model finding procedures for the first-order predicate logic [1, 12], which given a first-order sentence φ eventually produce a finite model for φ if such a model exists.

3 Fibonacci words

Fibonacci words probably are less known than their classical cousins Fibonacci numbers, but not less interesting. The infinite sequence F of Fibonacci words is defined recursively as

$$w_0 = b, w_1 = a, w_{i+2} = w_i w_{i+1}$$

and consists of the words: $b, a, ba, aba, baaba, ababaaba, baabaababaaba, \ldots$.

After quick inspection one may notice that none of the words shown above contains bb as a subword. The same is true for the subword aaa. Indeed, it is known that actually none of the infinitely many Fibonacci words contains bb or aaa as the subwords. The reader may wish to try to prove it himself/herself, or consult [15] for further hints.

Here we demonstrate how to establish such properties automatically using first-order theorem *disproving* by *finite countermodels* finding. Following [9] consider the theory FIB in first-order predicate logic:

- $(x * y) * z = x * (y * z)$
- $R(b, a)$
- $R(x, y) \to R(y, x * y)$

The first (semigroup) axiom expresses associativity of concatenation and remaining two axiomatize the binary predicate R, where intuitive meaning of $R(x, y)$ is x and y are two consecuitive Fibonacci words. We have now

Proposition 1. *If w is a Fibonacci word then $FIB \vdash \exists x R(t_w, x)$, where t_w denote a term encoding of w, i.e. $t_{aba} = (a * b) * a$*

Proof. It is sufficient to show that for all $i \geq 0$ $Fib \vdash R(t_{w_i}, t_{w_{i+1}})$. The proof proceeds by easy induction. For $i = 0$ we have $Fib \vdash R(b, a)$ and therefore $Fib \vdash R(t_{w_0}, t_{w_1})$. Assume $Fib \vdash R(t_{w_i}, t_{w_{i+1}})$. We also have $Fib \vdash R(x, y) \to R(y, x * y)$ (by definition of FIB). It follows that $Fib \vdash R(t_{w_{i+1}}, t_{w_i} * t_{w_{i+1}})$ (by application of *Modus Ponens*), and therefore $Fib \vdash R(t_{w_{i+1}}, t_{w_{i+2}})$ (by $w_{i+2} = w_i w_{i+1}$).

By contraposition we have now

Corollary 1. *1. If $FIB \nvdash \exists x \exists z \exists y R(z * b * b * y, x)$ then there is no Fibonacci word with bb as a subword, and similarly,*

2. If $FIB \nvdash \exists x \exists z \exists y R(z * a * a * a * y, x)$ then there is no Fibonacci word with aaa as a subword.

3.1 The case of 'bb'

To show $FIB \not\vdash \exists x \exists z \exists y R(z * b * b * y, x)$ it is sufficient to demonstrate a countermodel for $FIB \rightarrow \exists x \exists z \exists y R(z * b * b * y, x)$, or equivalently, a model for $FIB \wedge \neg \exists x \exists z \exists y R(z * b * b * y, x)$.

To find a model we apply generic *finite* model finding procedure [1], e.g. implemented in Mace4 finite model finder by W.McCune [12]. A model of size 5 is found in 0.05s. The property is proved, no *bb* will ever appear in a Fibonacci word. The actual output of Mace4 can be found in [6]. Here we describe the model.

The domain of the model $\mathcal{M}_1$ is a 5 elements set $D = \{0, 1, 2, 3, 4\}$. Interpretations of constants are: $[a] = 0$ and $[b] = 1$. The interpretation of $*$ is given by the following multiplication table:

$[*]$	0	1	2	3	4
0	0	3	0	3	4
1	2	4	4	1	4
2	2	1	2	1	4
3	0	4	4	3	4
4	4	4	4	4	4

The interpretation $[R]$ of the binary relation R is given by the table

$[R]$	0	1	2	3	4
0	0	0	1	0	0
1	1	0	0	0	0
2	1	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

which means $[R] = \{(1, 0), (2, 0), (0, 2)\}$

Now we explain how the existence of such a model proves the property. We have already presented a logical argument - the existence of the model prevents a formula above to be derivable and by virtue of that, prevents the corresponding word violating the property to be Fibonacci(an). That does not explain however *why* the property holds. Consider interpretations of (term encodings of) initial Fibonacci words: $[b] = 1$, $[a] = 0$, $[b * a] = 2$, $[a * b * a] = 0$, $[b * a * a * b * a] = 2, \ldots$. Using $[t_{w_{i+2}}] = [t_{w_{i+1}}] * [t_{w_{i+1}}]$ we conclude that $[t_{w_{2k}}] = 2$ and $[t_{w_{2k+1}}] = 0$ for all $k \geq 1$. Thus for any Fibonacci word w_i we have $[t_{w_i}] \in \{0, 1, 2\}$. Consider now the interpretation t_w of any word $w = xbby$. We claim that $[t_w] = 4 \notin \{0, 1, 2\}$. Indeed, $[b * b] = 4$ and for any $x, y \in D$ $x * 4 = 4$ and $4 * y = 4$, i.e. 4 is a zero element of the semigroup. That resolves the puzzle.

3.2 The case of 'aaa'

Similarly to the case above, to show $FIB \not\vdash \exists x \exists z \exists y R(z * a * a * a * y, x)$ it is sufficient to demonstrate a countermodel for $FIB \to \exists x \exists z \exists y R(z * a * a * a * y, x)$, or equivalently, a model for $FIB \land \neg \exists x \exists z \exists y R(z * a * a * a * y, x)$. It took 45s for Mace4 to find a countermodel $\mathcal{M}_2$ of size 11.

The domain of the model $\mathcal{M}_2$ is an 11 elements set $D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Interpretations of constants are: $[a] = 0$ and $[b] = 1$. The interpretation of $*$ is given by the following multiplication table:

[*]	0	1	2	3	4	5	6	7	8	9	10
0	2	4	5	9	7	5	10	5	5	8	2
1	3	1	6	3	1	5	6	1	3	3	6
2	5	7	5	8	5	5	2	5	5	5	5
3	6	1	5	3	1	5	6	5	5	3	6
4	9	4	10	9	4	5	10	4	9	9	10
5	5	5	5	5	5	5	5	5	5	5	5
6	5	1	5	3	5	5	6	5	5	5	5
7	8	7	2	8	7	5	2	7	8	8	2
8	2	7	5	8	7	5	2	5	5	8	2
9	10	4	5	9	4	5	10	5	5	9	10
10	5	4	5	9	5	5	10	5	5	5	5

The interpretation $[R]$ of the relation R is given by $[R] = \{(0, 3), (1, 0), (3, 9), (9, 3)\}$. Using the same arguments as in the case 'bb', we have $[t_{w_i}] \in \{0, 1, 3, 9\}$ for all Fibonacci words w_i and $[t_w] = 5 \notin \{0, 1, 3, 9\}$ for all $w = xaaay$. The property is established. Notice that as in the previous case the words with a forbidden pattern are interpreted by a zero element of the semigroup, that is 5 in this case.

3.3 Finite Models as Regular Invariants

The finite models produced above can be seen as the compact representations of the *regular* invariants (separators) sufficient to prove the properties. Indeed, consider, for example, the model $\mathcal{M}_2$ above and let $L = \{w \in \Sigma^* | [t_w] \in \{0, 1, 3, 9\}\}$ where $\Sigma = \{a, b\}$. Let $A \subseteq \Sigma^*$ be the language of all words containing aaa as a subword. Then we have $F \subseteq L$, $L \cap A = \emptyset$. The language L separates F and A and is regular due to well-known algebraic characterization of regular languages as the inverse images of homomorphisms from free monoids (semigroups) to finite monoids (semigroups) , see e.g. [5]. Thus the work of the finite model finder in this context can bee seen as a systematic attempt to build a regular separator, or invariant, sufficient to show the safety. This observation explains the relative completeness of FCM method with respect to the variants of *regular model checking* [8, 10, 11]. Notice that in general the termination of a finite model building procedure is not guaranteed. It can be fixed by setting an upper bound for the model size.

4 Challenge

The author was very surprised to find an automated solution of Fibonacci puzzle using Mace4 model finder and strongly suspect that it would be difficult to get such a solution in any other way. The author would like to challenge anyone to demonstrate an alternative automated solution to this puzzle. The conditions of the challenge will appear at
`http://www.csc.liv.ac.uk/~alexei/Fibonacci_Challenge/`

Have a fun!

5 Acknowledgement

The author learned about Fibonacci words and the puzzle above from the very nice presentation of the paper [4] by Naoki Kobayashi at TTATT 2012 Workshop, Nagoya, Japan, 02.06.2012. In the talk a *bounded* version of the problem considered, namely how to prove efficiently that all Fibonacci words up to w_{1000} do not contain fobidden patterns.

References

1. R. Caferra, A. Leitsch, N. Peltier, *Automated Model Building*, Applied Logic Series, 31, Kluwer, 2004.
2. Goubault-Larrecq, J., (2008), Towards producing formally checkable security proofs, automatically. In: Computer Security Foundations (CSF), pp. 224–238 (2008)
3. Guttman, J., (2009) Security Theorems via Model Theory, Proceedings 16th International Workshop on Expressiveness in Concurrency, EXPRESS, EPTCS, vol. 8 (2009)
4. Kobayashi, N., Matsuda, K., and Shinohara, A., Functional Programs as Compressed Data, Proceedings of TTATT 2012 1st International Workshop on Trends in Tree Automata and Tree Transducers, Nagoya, Japan, 02.06.2012.
5. Lallement, G., Semigroups and Combinatorial Applications, John Wiley & Sons, 1979
6. Lisitsa, A., (2009a), Verfication via countermodel finding `http://www.csc.liv.ac.uk/~alexei/countermodel/` (accessed 12.04.2013)
7. Lisitsa, A., (2010b), Reachability as deducibility, finite countermodels and verification. In Proceedings of ATVA 2010, LNCS 6252, 233–244
8. Lisitsa, A., (2010c), Finite model finding for parameterized verification, CoRR abs/1011.0447: (2010)
9. Lisitsa, A., (2012) , Finite models for Verification, a talk given in ENS Cachan, LSV, June 22, 2012, slides avialable at [6]
10. Lisitsa, A., Finite Models vs Tree Automata in Safety Verification, 23rd International Conference on Rewriting Techniques and Applications RTA' 2012, pp 225–239
11. Lisitsa, A., Finite Reasons for Safety, Journal of Automated Reasoning, DOI 10.1007/s10817-013-9274-9, February 2013
12. McCune, W., Prover9 and Mace4 `http://www.cs.unm.edu/~mccune/mace4/` (accessed 12.04.2013)

13. Selinger, P., (2001), Models for an adversary-centric protocol logic. Electr. Notes Theor. Comput. Sci. 55(1) (2001)
14. Weidenbach, C., (1999), Towards an Automatic Analysis of Security Protocols in First-Order Logic, in H. Ganzinger (Ed.): CADE-16, LNAI 1632, pp. 314–328, 1999.
15. Whealton, S., String Rewriting and the Fibonacci Word, http://www.washingtonart.net/whealton/fibword.html (accessed 12.04.2013)