Accepted Manuscript

On Fixed Points of Rational Transductions

Vesa Halava, Tero Harju, Esa Sahla

PII: S0304-3975(18)30264-0
DOI: https://doi.org/10.1016/j.tcs.2018.04.030
Reference: TCS 11560

To appear in: *Theoretical Computer Science*

Received date: 13 November 2017
Accepted date: 13 April 2018

Please cite this article in press as: V. Halava et al., On Fixed Points of Rational Transductions, *Theoret. Comput. Sci.* (2018), https://doi.org/10.1016/j.tcs.2018.04.030

On Fixed Points of Rational Transductions

Vesa Halava[1,2], Tero Harju[1], and Esa Sahla[*1]

[1]Department of Mathematics and Statistics, University of Turku, Finland
[2]Turku Complex System Institute, Department of Mathematics and Statistics, University of Turku, Finland
Emails: {vesa.halava, tero.harju, esa.sahla}@utu.fi

April 24, 2018

Abstract

We show that it is undecidable whether or not an injective rational function (realized by a finite transducer) $f : A^* \to A^*$ has a fixed point. The proof applies undecidability of the Post's Correspondence Problem for injective morphisms. As a corollary we obtain that the existence of a fixed point of injective computable functions is undecidable.

1 Introduction

We study the *fixed point problem* of functions starting from the finitely generated word semigroups:

Problem 1. *Let A be a finite alphabet. Let $f : A^* \to A^*$ be a function. Does there exists a word w such that $f(w) = w$.*

We shall prove that the fixed point problem is undecidable for functions defined by finite transducers, that is, functions defined by finite automata with output. Our result also gives a corollary for computable (recursive) functions over natural numbers.

It easily follows from the basic computability results on Turing machines that the existence of a fixed point is undecidable for general functions. Indeed, this follows from a modification of the classical diagonal argument or by the halting problem, i.e., by defining a function f in a way that it checks for any input i whether the ith Turing machine halts on its code, one can transform f into a function h having fixed point i if and only if the ith Turing machine halts on i. However, in such as construction the function h is clearly noncomputable.

[*]Corresponding author.