ARTICLE IN PRESS
YINCO:4274
Information and Computation ••• (••••) •••–•••
Contents lists available at ScienceDirect
Information and Computation
www.elsevier.com/locate/yinco

Information rate of some classes of non-regular languages: An automata-theoretic approach

Cewei Cui [a], Zhe Dang [a], Thomas R. Fischer [a], Oscar H. Ibarra [b,*]

[a] School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99164, USA
[b] Department of Computer Science, University of California, Santa Barbara, CA 93106, USA

ARTICLE INFO

Article history:
Received 10 October 2014
Available online xxxx

Keywords:
Automata
Formal languages
Shannon information

ABSTRACT

We show that the information rate of the language accepted by a reversal-bounded deterministic counter machine is computable. For the nondeterministic case, we provide computable upper bounds. For the class of languages accepted by multi-tape deterministic finite automata, the information rate is computable as well.
© 2017 Elsevier Inc. All rights reserved.

1. Introduction

A software system often interacts with its environment. The complexity of an average observable event sequence, or behavior, can be a good indicator of how difficult it is to understand its semantics, test its functionality, etc. This is particularly true considering the fact that a modern software system is often too complex to analyze algorithmically by looking at the source code, line by line. Instead, the system is treated as a black-box whose behaviors can be observed by running it (with provided inputs), i.e., testing. One source to obtain all of the system's intended behaviors is from the design, though whether an intended behavior is the system's actual behavior must still be confirmed through testing. Despite this, the problem of computing the complexity of an average intended behavior from the design is important; in particular, the complexity can be used to estimate the cost of testing, even at the design stage where the code is not available yet.

In principle, a behavior is a word and the design specifies a set of words, i.e., a language L. There has already been a fundamental notion shown below, proposed by Shannon [22] and later Chomsky and Miller [5], that we have evaluated through experiments over C programs [25], fitting our need for the aforementioned complexity. For a number n, we use $S_n(L)$ to denote the number of words in L whose length is n. The *information rate* λ_L of L is defined as

$$\lambda_L = \overline{\lim_{n\to\infty}} \frac{\log S_n(L)}{n},$$

which always exists for a finite alphabet. Throughout this paper, the logarithm is base 2. The rate is closely related to data compression ratio [12] and hence has immediate practical applications [7,6,15,10]. Information rate is a real number. Hence, as usual, when we say that the rate is computable, it means that we have an algorithm to compute the rate up to any given precision (i.e., first n digits, for any n). A fundamental result is in the following.

Theorem 1. *The information rate of a regular language is computable [5].*

* Corresponding author.
E-mail addresses: ccui@eecs.wsu.edu (C. Cui), zdang@eecs.wsu.edu (Z. Dang), fischer@eecs.wsu.edu (T.R. Fischer), ibarra@cs.ucsb.edu (O.H. Ibarra).

http://dx.doi.org/10.1016/j.ic.2017.04.008
0890-5401/© 2017 Elsevier Inc. All rights reserved.

Please cite this article in press as: C. Cui et al., Information rate of some classes of non-regular languages: An automata-theoretic approach, Inf. Comput. (2017), http://dx.doi.org/10.1016/j.ic.2017.04.008

The case when L is non-regular (e.g., L is the external behavior set of a software system containing (unbounded) integer variables like counters and clocks) is more interesting, considering the fact that a complex software system nowadays is almost always of infinite-state and the notion of information rate has been used in software testing [25,26]. However, in such a case, computing the information rate is difficult (sometimes even not computable [16]) in general. Existing results, such as unambiguous context-free languages [17], Lukasiewicz-languages [23], and regular timed languages [2], are limited and mostly rely on structure generating functions introduced by Kuich [17] and the theory of complex/real functions, which are also difficult to generalize.

In this paper, instead of taking the path of using structure generating functions, we use automata-theoretic approaches to compute the information rate for some classes of non-regular languages, including languages accepted by machines equipped with restricted counters. Our approaches make use of the rich body of techniques in automata theory developed in the past several decades and, as we believe, the approaches themselves can also be applied to more general classes of languages.

We first investigate languages accepted by reversal-bounded nondeterministic counter machines [13]. A counter is a nonnegative integer variable that can be incremented by 1, decremented by 1, or stay unchanged. In addition, a counter can be tested against 0. Let k be a nonnegative integer. A *nondeterministic k-counter machine (NCM)* is a one-way nondeterministic finite automaton, with input alphabet Σ, augmented with k counters. For a nonnegative integer r, we use NCM(k,r) to denote the class of k-counter machines where each counter is *r-reversal-bounded*; i.e., it makes at most r alternations between nondecreasing and nonincreasing modes in any computation; e.g., the following counter value sequence

$$0\ 0\ 1\ 2\ 2\ 3\ \underline{3\ 2}\ 1\ 0\ \underline{0\ 1}\ 1$$

is of 2-reversal, where the reversals are underlined. For convenience, we sometimes refer to a machine M in the class as an NCM(k,r). In particular, when k and r are implicitly given, we call M as a *reversal-bounded NCM*. When M is deterministic, we use 'D' in place of 'N'; e.g., DCM. As usual, $L(M)$ denotes the language that M accepts.

Reversal-bounded NCMs have been extensively studied since their introduction in 1978 [13]; many generalizations are identified; e.g., ones equipped with multiple tapes, with two-way tapes, with a stack, etc. In particular, reversal-bounded NCMs have found applications in areas like Alur and Dill's [1] time-automata [9,8], Paun's [21] membrane computing systems [14], and Diophantine equations [24].

In this paper, we show that the information rate of the language L accepted by a reversal-bounded DCM is computable. The proof is quite complex. We first, using automata-theoretic techniques, modify the language into essentially a regular language, specified by an unambiguous regular expression that is without nested Kleene stars, further constrained by a Presburger formula on the symbol counts in the words of the regular language. We show that the information rate of L can be computed through the information rate of the constrained language, where the latter can be reduced to a simple and solvable convex minimization problem. Unfortunately, we are not able to generalize the technique to reversal-bounded NCM. However, it is known [3] that a reversal-bounded NCM can be made to be one with counter values linearly bounded (in input size). Using this fact, we are able to obtain a computable upper bound on the rate when a reversal-bounded NCM is considered. We also consider the case when the reversal-bounded NCM does not make a lot of nondeterministic choices (i.e., sublinear-choice). In this case, the rate is shown computable. The result leads us to study a class of languages accepted by multi-tape DFAs. The information rate of such a multi-tape language is computable as well.

2. Information rate of languages accepted by reversal-bounded counter machines

We now recall a number of definitions that will be used later. Let N be the set of nonnegative integers and k be a positive number. A subset S of N^k is a linear set if there are vectors $\mathbf{v}_0, \mathbf{v}_1, \cdots, \mathbf{v}_t$, for some t, in N^k such that $S = \{\mathbf{v} : \mathbf{v} = \mathbf{v}_0 + b_1\mathbf{v}_1 + \cdots + b_t\mathbf{v}_t, b_i \in N\}$. S is a semilinear set if it is a finite union of linear sets. Let $\Sigma = \{a_1, \cdots, a_k\}$ be an alphabet. For each word $\alpha \in \Sigma^*$, define the Parikh map [20] of α to be the vector $\#(\alpha) = (\#_{a_1}(\alpha), \cdots, \#_{a_k}(\alpha))$, where each symbol count $\#_{a_i}(\alpha)$ denotes the number of symbol a_i's in α. For a language $L \subseteq \Sigma^*$, the Parikh map of L is the set $\#(L) = \{\#(\alpha) : \alpha \in L\}$. The language L is semilinear if $\#(L)$ is a semilinear set. There is a classic result needed in the paper:

Theorem 2. *Let M be a reversal-bounded NCM. Then $\#(L(M))$ is a semilinear set effectively computable from M [13].*

Let Y be a finite set of integer variables. An atomic Presburger formula on Y is either a linear constraint $\sum_{y \in Y} a_y y < b$, or a mod constraint $x \equiv_d c$, where a_y, b, c and d are integers with $0 \le c < d$. A Presburger formula can always be constructed from atomic Presburger formulas using $\neg$ and $\wedge$. It is known that Presburger formulas are closed under quantification. Let S be a set of k-tuples in N^k. S is Presburger definable if there is a Presburger formula $P(y_1, \cdots, y_k)$ such that the set of nonnegative integer solutions is exactly S. It is well-known that S is a semilinear set iff S is Presburger definable.

Let M be a reversal-bounded deterministic counter machine. The main result of this paper shows that the information rate of $L(M)$ is computable. The proof has four steps. First, we show that the information rate of $L(M)$ can be computed through the information rate of a counting language (defined in a moment) effectively constructed from M. Second, we show that the information rate of a counting language can be computed through the information rate of a counting replacement language (also defined in a moment) effectively constructed from the counting language. Third, we show that the

Download English Version:

https://daneshyari.com/en/article/4950575

Download Persian Version:

https://daneshyari.com/article/4950575

Daneshyari.com