Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Journal of Computer and System Sciences 71 (2005) 440–466
JOURNAL OF COMPUTER AND SYSTEM SCIENCES
www.elsevier.com/locate/jcss

Measuring nondeterminism in pushdown automata

Jonathan Goldstine[a,*], Hing Leung[b], Detlef Wotschke[c,1]

[a] *Department of Computer Science and Engineering, The Pennsylvania State University, University Park, Pennsylvania, USA*
[b] *Department of Computer Science, New Mexico State University, Las Cruces, New Mexico, USA*
[c] *Fachbereich Informatik, Johann Wolfgang Goethe-Universität, Frankfurt, Germany*

Received 14 June 2000; received in revised form 3 March 2004

Available online 15 June 2005

Abstract

The amount of nondeterminism that a pushdown automaton requires to recognize an input string can be measured by the minimum number of guesses that it must make to accept the string, where guesses are measured in bits of information. When this quantity is unbounded, the rate at which it grows as the length of the string increases serves as a measure of the pushdown automaton's "rate of consumption" of nondeterminism. We show that this measure is similar to other complexity measures in that it gives rise to an infinite hierarchy of complexity classes of context-free languages differing in the amount of the resource (in this case, nondeterminism) that they require. In addition, we show that there are context-free languages that can only be recognized by a pushdown automaton whose nondeterminism grows linearly, resolving an open problem in the literature. In particular, the set of palindromes is such a language.

Keywords: Nondeterminism; Pushdown automata

1. Introduction

The simplest way to define time- or space-limited deterministic Turing machines is to require that *every* computation on an input of length n consume no more than $f(n)$ units of time or space. It is then

* Corresponding author.
E-mail addresses: goldstin@cse.psu.edu (J. Goldstine), hleung@cs.nmsu.edu (H. Leung), wotschke@psc.informatik.uni-frankfurt.de (D. Wotschke).
[1] Supported in part by the Stiftung Volkswagenwerk under Grant No. I/72 256.

0022-0000/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcss.2005.04.001

natural to extend the same definition to nondeterministic Turing machines. This corresponds to charging a nondeterministic machine for its worst effort. However, a nondeterministic Turing machine that satisfies the weaker requirement of having at least one accepting computation satisfying a well-behaved resource bound $f(n)$ for each string that it accepts can be converted to a machine satisfying the stronger requirement that every computation satisfies this resource bound simply by having the machine also calculate $f(n)$ within the resource bound $f(n)$ in order to shut the original calculation down if it exceeds this bound. Thus, the distinction between charging for best effort or worst effort is unimportant in this context. This is, of course, no longer the case for a pushdown automaton, since a pushdown automaton is not powerful enough to calculate $f(n)$ while carrying out its original computation, and so the decision of whether to count its best or worst effort now becomes significant.

Kintala and Fischer initiated the study of nondeterminism as a measurable resource in 1977 [10]. They defined the amount of nondeterminism that a Turing machine T uses to recognize an input string w to be the minimum number of nondeterministic moves that T makes during computations that accept w. (The amount of nondeterminism needed on inputs of length n is then the maximum amount needed for input strings of that length.) With this definition, they were able to prove the existence of an infinite hierarchy of complexity classes within the family of languages accepted by Turing machines in real time [2] and within the family of languages accepted in relativized polynomial time [11].

Vermeir and Savitch initiated the quantitative study of nondeterminism in context-free languages in 1981 [22]. They studied two measures of nondeterminism in pushdown automata, a dynamic measure and a static measure. The dynamic measure, designated the *maxmax* measure in [19], counts the maximum number of nondeterministic steps for any computation accepting an input string of length n. Thus, instead of charging a machine on the basis of its best performance, as Kintala and Fischer did, Vermeir and Savitch charged a pushdown automaton for its most costly accepting computation. This measure is easy to handle technically, since one can apply the usual pumping lemmas to “pump up” the number of nondeterministic steps in a computation. However, for precisely this reason, the complexity hierarchy of context-free languages under this measure collapses into just three classes: deterministic context-free languages, finite unions of deterministic context-free languages, and all context-free languages.[2] Indeed, Vermeir and Savitch [22] and Nasyrov [15] both cite the inability of this dynamic measure to produce a hierarchy of more than three levels as a justification for focusing attention on the static measure of nondeterminism (which Nasyrov generalizes to produce a dynamic measure [14,15]), or on a dynamic measure of the amount of ambiguity rather than of nondeterminism in a pushdown automaton [15]. While we share the view that the inability to produce a meaningful hierarchy is a critical weakness of this measure, we believe that this is a defect in the definition of the dynamic measure used in [22], not in the concept of a dynamic measure.

In the present paper, we prove that a different dynamic measure of nondeterminism in pushdown automata, one we believe to be better motivated than the maxmax measure, does produce an infinite hierarchy of context-free language families. We define the amount of nondeterminism that a pushdown

[2] If this measure counted the amount of information, measured in bits, represented by the nondeterministic choices made by a particular computation—as we will do in the present paper—then the finite unions of deterministic languages would be separated into different complexity classes according to the number of deterministic languages needed in the union. While this would produce an infinite hierarchy, it would merely be the known hierarchy produced by forming unions of deterministic context-free languages [9], and so even with this change, the maxmax measure would not produce a new hierarchy of context-free languages.