Unary Pushdown Automata
and Straight-Line Programs

Dmitry Chistikov and Rupak Majumdar
Max Planck Institute for Software Systems (MPI-SWS)
Kaiserslautern and Saarbrücken, Germany
{dch,rupak}@mpi-sws.org

Abstract. We consider decision problems for deterministic pushdown
automata over the unary alphabet (udpda, for short). Udpda are a simple
computation model that accept exactly the unary regular languages,
but can be exponentially more succinct than finite-state automata. We
complete the complexity landscape for udpda by showing that emptiness
(and thus universality) is $P$-hard, equivalence and compressed mem-
bership problems are $P$-complete, and inclusion is $\text{coNP}$-complete. Our
upper bounds are based on a translation theorem between udpda and
straight-line programs over the binary alphabet (SLPs). We show that
the characteristic sequence of any udpda can be represented as a pair
of SLPs—one for the prefix, one for the lasso—that have size linear in
the size of the udpda and can be computed in polynomial time. Hence,
decision problems on udpda are reduced to decision problems on SLPs.
Conversely, any SLP can be converted in logarithmic space into a udpda,
and this forms the basis for our lower bound proofs. We show $\text{coNP}$-
hardness of the ordered matching problem for SLPs, from which we derive
$\text{coNP}$-hardness for inclusion. In addition, we complete the complexity
landscape for unary nondeterministic pushdown automata by showing
that the universality problem is $\Pi_2P$-hard, using a new class of inte-
ger expressions. Our techniques have applications beyond udpda. We
show that our results imply $\Pi_2P$-completeness for a natural fragment of
Presburger arithmetic and $\text{coNP}$ lower bounds for compressed matching
problems with one-character wildcards.

1 Introduction

Any model of computation comes with a set of fundamental decision questions:
emptiness (does a machine accept some input?), universality (does it accept all
inputs?), inclusion (are all inputs accepted by one machine also accepted by
another?), and equivalence (do two machines accept exactly the same inputs?).
The theoretical computer science community has a fairly good understanding
of the precise complexity of these problems for most “classical” models, such as
finite and pushdown automata, with only a few prominent open questions (e.g.,
the precise complexity of equivalence for deterministic pushdown automata).

* The full version of the paper is available at http://arxiv.org/abs/1403.0509
In this paper, we study a simple class of machines: deterministic pushdown automata working on unary alphabets (unary dpda, or udpda for short). A classic theorem of Ginsburg and Rice [7] shows that they accept exactly the unary regular languages, albeit with potentially exponential succinctness when compared to finite automata. However, the precise complexity of most basic decision problems for udpda has remained open.

Our first and main contribution is that we close the complexity picture for these devices. We show that emptiness is already \( \mathbf{P} \)-hard for udpda (even when the stack is bounded by a linear function of the number of states) and thus \( \mathbf{P} \)-complete. By closure under complementation, it follows that universality is \( \mathbf{P} \)-complete as well. Our main technical construction shows equivalence is in \( \mathbf{P} \) (and so \( \mathbf{P} \)-complete). Somewhat unexpectedly, inclusion is \( \text{coNP} \)-complete. In addition, we study the \textit{compressed membership} problem: given a udpda over the alphabet \( \{a\} \) and a number \( n \) in binary, is \( a^n \) in the language? We show that this problem is \( \mathbf{P} \)-complete too.

A natural attempt at a decision procedure for equivalence or compressed membership would go through translations to finite automata (since udpda only accept regular languages, such a translation is possible). Unfortunately, these automata can be exponentially larger than the udpda and, as we demonstrate, such algorithms are not optimal. Instead, our approach establishes a connection to \textit{straight-line programs} (SLPs) on binary words—a well-studied model for word compression (see, e.g., Lohrey [20]). An SLP \( \mathcal{P} \) is a context-free grammar generating a single word, denoted \( \text{eval}(\mathcal{P}) \), over \( \{0, 1\} \). Our main construction is a translation theorem: for any udpda, we construct in polynomial time two SLPs \( \mathcal{P}' \) and \( \mathcal{P}'' \) such that the infinite sequence \( \text{eval}(\mathcal{P}') \cdot \text{eval}(\mathcal{P}'')^\omega \in \{0, 1\}^\omega \) is the characteristic sequence of the language of the udpda (for any \( i \geq 0 \), its \( i \)th element is 1 iff \( a^i \) is in the language). With this construction, decision problems on udpda reduce to decision problems on compressed words. Conversely, we show that from any pair \( (\mathcal{P}', \mathcal{P}'') \) of SLPs one can compute, in logarithmic space, a udpda accepting the language with characteristic sequence \( \text{eval}(\mathcal{P}') \cdot \text{eval}(\mathcal{P}'')^\omega \). Thus, lower bounds for computational complexity of decision problems for udpda may be obtained from the corresponding lower bounds for SLPs. Indeed, we show \( \text{coNP} \)-hardness of inclusion via \( \text{coNP} \)-hardness of the ordered matching problem for compressed words (i.e., is \( \text{eval}(\mathcal{P}_1) \leq \text{eval}(\mathcal{P}_2) \) letter-by-letter, where the alphabet comes with an ordering \( \leq \)), a problem of independent interest.

As a second contribution, we complete the complexity picture for unary non-deterministic pushdown automata (unpda, for short). For unpda, the precise complexity of most decision problems was already known [14]. The remaining open question was the precise complexity of the universality problem, and we show that it is \( \Pi_2^p \)-hard (membership in \( \Pi_2^p \) was shown earlier by Huynh [14]). An equivalent question was left open in Kopczyński and To [18] in 2010, but the question was posed as early as 1976 by Hunt III, Rosenkrantz, and Szymanski [12, Open Problem 2], where it was asked whether the problem was in \( \mathbf{NP} \) or \( \mathbf{PSPACE} \) or outside both. Huynh’s \( \Pi_2^p \)-completeness result for equivalence [14] showed, in particular, that universality was in \( \mathbf{PSPACE} \), and our