ON THE DECIDABILITY OF EQUIVALENCE FOR DETERMINISTIC PUSHDOWN TRANSDUCERS *

Oscar H. IBARRA and Louis E. ROSIER
Department of Computer Science, University of Minnesota, Minneapolis, MN 55455, U.S.A.

Received 20 July 1981; revised version received 30 September 1981

Pushdown transducer, equivalence, alternate stacking

1. Introduction

A well-known open problem in formal language theory is whether equivalence is decidable for the class of deterministic pushdown automata (DPDA's). In the past few years this problem has received much attention and while the general problem still remains open, many different subclasses have been shown to have a decidable equivalence problem [2,4–5,12–19]. The basic techniques used to prove these results were first introduced by Valiant [17]. Valiant observed the following lemma:

Lemma 1. Suppose there is an effective procedure which, when given M_1 and M_2 belonging to a class T of automata, can enumerate a family $P(M_1, M_2)$ of PDA's with the following properties:

(1) $L(M_1) = L(M_2) \Rightarrow \exists M' \in P(M_1, M_2)$ such that $L(M') = \emptyset$.

(2) $L(M_1) \neq L(M_2) \Rightarrow \forall M' \in P(M_1, M_2)$, $L(M') \neq \emptyset$.

Then equivalence is decidable for T.

Actually this gives a partial decision procedure for deciding equivalence in T, but since inequivalence is also partially decidable we have a total decision procedure.

For different subclasses of DPDA's, Valiant described two methods of construction in which to build simulating PDA's for DPDA's M_1 and M_2. The two techniques have become known as 'Alternate Stacking' and 'Parallel Stacking'.

'Alternate Stacking' involves simulating two machines M_1 and M_2 both in T with one PDA M whose stack contents are $u_1, v_1, ..., u_n, v_n$ which are encodings of the stack contents $u_1, ..., u_n$ and $v_1, ..., v_n$ for M_1 and M_2 respectively. This can be done for any two DPDA's but is only successful if the top segment or segments can be kept uniformly bounded [17]. The 'Parallel Stacking' technique is similar in the sense that the two machines are simulated simultaneously using one stack; however, to keep the segment lengths bounded certain nondeterministic stack replacements are employed, thus changing the actual contents of the stack (and in a way, also the input being simulated). For our purposes, the important difference between the two techniques is that the simulation used with alternate stacking is 'faithful' in the sense that as far as the simulation is carried out it is not changed in any way from that of the machine being simulated (thus the simulation may be abandoned but it is never altered and then allowed to proceed as it is in the parallel stacking technique).

In this paper, we show how alternate stacking can be used to solve the equivalence problem for certain subclasses of deterministic pushdown transducers. We also use the technique to decide equivalence for other related classes of automata with pushdown store.

A deterministic pushdown transducer (DPDT) is a DPDA with outputs. (See [1] for a formal definition.) We will call the DPDA that is obtained by ignoring the output of a DPDT, the associated DPDA. In this paper we show the following: Let P be a class of single-valued DPDT's for which the class of associated

* This research was supported in part by NSF Grant MCS 81-02853.

DPDA's can be shown to have a decidable equivalence problem using the alternate stacking construction. Then P has also a decidable equivalence problem. (A DPDT is single-valued if for every input there is at most one output * (see [6]). Two single-valued DPDT's M_1 and M_2 are equivalent if they accept exactly the same set of inputs and for each input accepted produce exactly the same output.)

Generally the equivalence theorems for DPDA's impose (without loss of generality) a normal form on the DPDA's. For the case of DPDT's we make similar assumptions, i.e. that the machine (1) increases the pushdown store by at most one symbol per move, (2) has no blocking modes (i.e. must read all the input before terminating), (3) has no transitions defined on empty store, and (4) the accepting modes are a subset of the reading modes.

2. The equivalence of DPDT's

Definition. A multicounter nondeterministic pushdown automaton (MPDA) is a PDA which is augmented with any finite number of reversal bounded counters. An MDPDA is a deterministic MPDA. Formal definitions can be found in [10].

We will need the following lemma proved in [10].

Lemma 2. The emptiness problem (determining if a machine accepts the empty language) for MPDA's is decidable.

We now observe that since the solvability of the emptiness problem for PDA's was the crucial property required in Lemma 1, that we can replace the family of PDA's with a family of MPDA's and the lemma still holds.

We are now ready to prove the following:

Theorem 1. Let P be a class of single-valued DPDT's for which the class of associated DPDA's can be shown to have a decidable equivalence problem using the alternate stacking construction. Then P has also a decidable equivalence problem.

* If an input is not accepted, the output is undefined.

Proof. The proof follows from Lemmas 1 and 2. Given M_1 and M_2, we (effectively) enumerate a sequence of MPDA's $P(M_1, M_2)$ such that there exists an M' in $P(M_1, M_2)$ satisfying $L(M') = \emptyset$ if and only if $M_1 \equiv M_2$.

A member M'' in the enumeration operates its pushdown as in the alternate stacking construction used to decide the equivalence of the associated DPDA's. In addition M'' has two counters that are used to guess the position of a discrepancy in the transductions of accepted input strings. Both counters are initially set nondeterministically to the same integer. Then the simulation proceeds exactly as in the alternate stacking construction except the counters are decremented by the lengths of the outputs of a transition made by M_1 and M_2 respectively (one counter is used for each machine). When a counter reaches zero the symbol being transduced at that time is stored in the finite state control. In the case that the simulation is carried out to its conclusion for an input x and x is accepted by both M_1 and M_2, then the MPDA will accept the string x if and only if the position guessed in the transductions actually yields a discrepancy.

Then for the correct simulator M', $L(M') = \emptyset$ if and only if $M_1 \equiv M_2$ since:

(1) The computations of the PDA associated with M' will find a distinguishing input string (one accepted by either M_1 or M_2 but not both) if one exists, so we need only concern ourselves with cases when exactly the same inputs are accepted by M_1 and M_2.

(2) If there exists an input x such that it is accepted by both M_1 and M_2 but whose transductions differ, then the augmented procedure will insure acceptance of such a string.

Since the alternate stacking construction insures a faithful simulation whenever the input is accepted by both M_1 and M_2, the use of the counters will detect a discrepancy if one exists. Hence $M_1 \equiv M_2$ if and only if $L(M') = \emptyset$ where M' is the correct simulator. The conclusion now follows from the decidability of the emptiness problem for MPDA's (Lemma 2).

Below are some of the classes of DPDA's for which equivalence has been shown decidable using the alternate stacking technique. It follows from Theorem 1 that the classes of DPDT's obtained from these machines by attaching an output structure have a decidable equivalence problem.

(1) The class of superdeterministic DPDA's of Friedman and Greibach [4,5] which also contains the proper subclasses (see [4] for discussion):

(a) The ultrarealtime languages.

(b) The left-structured grammars of Yaffle [20].

(c) The stack uniform machines of Linna [12].

(d) The strict restricted machines of Igarashi [11].

(2) The nonsingular machines of Valiant [17].

(3) Taniguchi and Kasami [16] generalized Valiant's result so that only one of the machines needed to be nonsingular.

(4) The strict deterministic realtime PDA's [15].

3. Multicounter deterministic pushdown automata

The techniques of section 2 can be used to show the decidability of equivalence for certain types of DPDA's augmented by finite-reversal counters. An MDPDA (short for multicounter deterministic pushdown automata) M is an 8-tuple $M = \langle Q, \Sigma, \Gamma, q_0, \delta, F, k, r\rangle$ where k is the number of counters, r is the number of reversals allowed per counter, Σ and Γ are input and stack alphabets respectively. q_0 is the start state and Z_0 in Γ is the bottom-of-stack marker which appears only at the bottom of the stack and is never changed. Q is a set of states where $Q = Q_1 \times Q_2$ and intuitively $(p, q) \in Q$ means p is the state associated with the counters and q is the state associated with the pushdown store. $F \subseteq Q \times \Gamma$ is a set of accepting modes. δ is a mapping from

$$Q \times \Sigma \cup \{\epsilon\} \times \Gamma \times \prod_{i=1}^{k} \lambda(c_i)$$

where

$$\lambda(c_i) = \begin{cases} 0 & \text{if counter i contains 0,} \\ 1 & \text{otherwise,} \end{cases}$$

to

$$Q \times \Gamma^* \times \prod_{i=1}^{k} d_i \quad (d_i = -1, 0, 1).$$

Furthermore we require that:

$$\delta((p_1, q), a, A, \lambda(c_1), ..., \lambda(c_k)) =$$

$$= ((p_1', q'), w', d_1, ..., d_k)$$

and

$$\delta((p_2, q), a, A, \lambda(c_1'), ..., \lambda(c_k')) =$$

$$= ((p_2', q''), w'', d_1', ..., d_k')$$

imply that (a) $q' = q''$ and (b) $w' = w''$. (i.e. The contents of the counters do not affect the action of the machine on the pushdown store.)

Since the counters do not interfere with the operation of the pushdown store, one can think of an associated DPDA for each of these MDPDA's. Given the MDPDA M, the associated DPDA is $M' = \langle Q_2, \Sigma, \Gamma, q_0', \delta', F'\rangle$ where Q_1, Σ, Γ are as before, q_0' is the second member of the triple q_0. δ' is a mapping from $Q_2 \times \Sigma \cup \{\epsilon\} \times \Gamma$ to $Q_2 \times \Gamma^*$. $F' \subseteq Q_2 \times \Gamma$ such that $(q_2, A) \in F'$ if and only if there exists a $q_1 \in Q_1$ such that $((q_1, q_2), A) \in F$. From the above definition it can be seen that the associated DPDA is independent of the counters and the language accepted by this machine contains the language accepted by M (i.e. $L(M) \subseteq L(M')$).

MDPDA's are more powerful than their DPDA's. For example consider the languages:

$A = \{x \# x^R \mid$ the number of 0's in x is equal to the number of 1's$\}$.

$B = \{x \# 0^i 1^j a 2^i \# x^R \mid i, j \geq 1\}$
$\cup \{x \# 0^i 1^j b 2^j \# x^R \mid i, j \geq 1\}$.

$C = \{x \# a^i b^j c^i \# x^R \mid i \geq 1\}$.

Those languages can be accepted by MDPDA's whose associated DPDA's behave superdeterministically (see [4,5]). Note that A and C are not context-free languages. B on the other hand is a deterministic context-free language but cannot be accepted by a realtime DPDA [8]. Another trivial extension only necessitates that the DPDA follow the restrictions of the subclass to which it belongs only until the decision of acceptance is made. At which time the stack is no longer used anyway and so bounding the segments for the rest of the dual simulation is trivial. This for instance would allow GREATERTHAN $= \{x \# y^R \mid x \neq y, |y| \geq |x|\}$ to be accepted superdeterministically [5].

Note that these MDPDA's can accept languages of the form $L_1 \cap L_2$, where L_1 is a language accepted by a DPDA and L_2 is a language accepted by a deterministic finite-reversal multi-counter machine. However, the converse does not seem likely since the actions of the counters may depend on the contents of the pushdown.

In order to use the results of previous papers we must insure that the associated DPDA of an MDPDA is in normal form (e.g. see [4,17]). Since the counters do not affect the operation of the pushdown store the normalization for such an MDPDA follows directly from the earlier results. Any catching up needed can be realized by running only the counters on the pushdown (on ϵ-input) until synchronization is complete. So without loss of generality we assume all MDPDA's are normalized.

We are now ready to state the following:

Theorem 2. Let P be a class of MDPDA's for which the class of associated DPDA's have a decidable equivalence problem that can be shown using the alternate stacking construction. Then P has also a decidable equivalence problem.

The proof of Theorem 2 is similar to that of Theorem 1 and will be omitted. Suffice it to say that the simulating MPDA's do not as before guess the position of a discrepancy in the outputs (because the machines have no outputs), but instead have a counter for each counter in M_1 and M_2 and merely simulate directly. On a faithful simulation, the simulating machine accepts if and only if exactly one of M_1 or M_2 does.

Finally, we note that an output structure can be attached to MDPDA's and we get:

Corollary 1. Let P be the class of single-valued MDPDT's (multicounter deterministic pushdown transducers) for which the class of associated DPDA's have a decidable equivalence problem that can be shown using the alternate stacking construction. Then P has also a decidable equivalence problem.

4. Conclusion

One further result that can be obtained follows from what has previously been said and a result in [5]: Given an arbitrary context-free language L and an MDPDA M defined as in section 3 whose associated DPDA is superdeterministic * (and acceptance is by final state *and* empty store [5]), it is decidable whether $L \subseteq L(M)$.

It would be nice if these results could be extended to include the subclasses of DPDA's whose equivalence problem was shown decidable by the use of the parallel stacking technique with certain replacement strategies. For such classes of DPDT's and MDPDA's it is not even clear whether substitutions exist since the substitutions used for the associated DPDA's may no longer be allowed. If substitutions exist, additional problems may occur if the substitutions induced extra counter reversals in the simulating MPDA (if certain replacements exist and for those replacements the simulating machine needed to reverse the counters to bring the simulation into step, then the simulating machine may no longer be reversal bounded.) Other possible extensions may be to remove the restrictions imposed on the machines in sections 2 and 3. Doing so seems to invalidate the simulation strategy however. Lastly we ask the following question: If the equivalence problem for DPDA's is decidable are the equivalence problems for DPDT's and MDPDA's decidable ?

* Note that the superdeterministic PDA's in [5] accept by final state and empty store while the superdeterministic PDA's in [4] accept by accept mode only. The latter are clearly more general and for such machines the inclusion problem is not even decidable [4].

References

[1] A.V. Aho and J.D. Ullman, The Theory of Parsing, Translation and Compiling, Vol. 1: Parsing (Prentice Hall, Englewood Cliffs, NJ, 1972).

[2] C. Beeri, An improvement on Valiant's decision procedure for equivalence of deterministic finite-turn pushdown automata, Theoret. Comput. Sci. 3 (1976) 305–320.

[3] P.J. Denning, J.B. Dennis and J.E. Qualitz, Machines Languages and Computation (Prentice-Hall, Englewood Cliffs, NJ, 1978).

[4] E.P. Friedman and S.A. Greibach, Superdeterministic DPDA's: The method of accepting does affect decision problems, J. Comput. System Sci. 19 (1979) 79–117.

[5] E.P. Friedman and S.A. Greibach, Superdeterministic PDA's: A subcase with a decidable inclusion problem, J. ACM, to appear.

[6] E.M. Gurari and O.H. Ibarra, Finite-valued transducers, submitted for publication.

[7] M.A. Harrison, Introduction to Formal Language Theory (Addison-Wesley, Reading, MA, 1978).

[8] J.E. Hopcroft and J.D. Ullman, Introduction to Automata Theory, Languages and Computation (Addison-Wesley, Reading, MA, 1979).

[9] O.H. Ibarra, The unsolvability of the equivalence problem for ϵ-free NGSM's with unary input (output) alpha-

bet and applications, SIAM J. Comput. 7 (4) (1979).
[10] O.H. Ibarra, Reversal-bounded multicounter machines and their decision problems, J. ACM 25 (1) (1978) 116–133.
[11] Y. Igarashi, Tape bounds for some subclasses of deterministic context-free languages, Inform. and Control 37 (1978) 321–333.
[12] M. Linna, Two decidability results for deterministic pushdown automata, J. Comput. System Sci. 18 (1979) 92–107.
[13] M. Oyamaguchi, Some results on equivalence and subclass containment problems for DPDA's, Proc. 5th IBM symposium on Mathematical Foundations of Computer Science, Japan (1980).
[14] M. Oyamaguchi and N. Honda, The decidability of equivalence for deterministic stateless pushdown automata, Inform. and Control 38 (1978) 367–376.
[15] M. Oyamaguchi, N. Honda and Y. Inagaki, The equivalence problem for realtime strict deterministic languages, unpublished manuscript.
[16] K. Tanaguchi and T. Kasami, A result on the equivalence problem for deterministic pushdown automata, J. Comput. System Sci. 13 (1976) 38–50.
[17] L.G. Valiant, Decision procedures for families of deterministic pushdown automata, Ph.D. thesis, University of Warwick, U.K. (1973).
[18] L.G. Valiant, The equivalence problem for deterministic finite-turn pushdown automata, Inform. and Control 25 (1975) 123–133.
[19] L.G. Valiant and M.S. Paterson, Deterministic one-counter automata, J. Comput. System Sci. 10 (1975) 340–350.
[20] V.A. Yaffe, On two classes of CF-languages with a solvable equivalence problem, Kibernetika (Kiev) 2 (1974) 89–93 (in English translation).