1-Normal DRA for Insertion Languages

Abhisek Midya[1(✉)], Lakshmanan Kuppusamy[2], V.S. Sumitha[3], and Alok Kumar Pani[3]

[1] Computer Science & Engineering, ICFAI Tech School, Hyderabad 501203, India
abhisekmidyacse@gmail.com
[2] Theoretical Computer Science Lab, School of Computer Science and Engineering, VIT University, Vellore 632014, India
klakshma@vit.ac.in
[3] Computer Science and Engineering, Christ University Faculty of Engineering, Bangalore 560074, India
{sumitha.vs,alok.kumar}@christuniversity.in

Abstract. Restarting automaton is a type of regulated rewriting system, introduced as a model for analysis by reduction. It is a linguistically motivated method for checking the correctness of a sentence. In this paper, we introduce a new definition of normal restarting automaton in which only one substring is removed using the DEL operation in a cycle. This DEL operation is applied to reverse the insertion operation in an insertion grammar. We use this 1-normal restarting automaton to solve the membership problem of insertion languages. Further, we introduce some interesting closure properties of 1-normal restarting automata.

Keywords: Insertion grammars · Membership problem · Restarting automaton

1 Introduction

The restarting automaton was introduced by Petr Jancar et al. in 1995 in order to model the 'analysis by reduction', which is a technique being used in linguistics to analyze sentences of *natural languages*. Analysis by reduction consists of step wise simplifications (reductions) of a given (lexically disambiguate) extended sentence until a correct simple sentence is obtained. It is accepted, until an error is found and the input is rejected. Each simplification replaces a short part of the sentence by an even shorter one.

A restarting automaton contains a finite control unit, a head with a look-ahead window attached to a tape. At several points it does cut-off substrings from the look-ahead window using DEL operation followed by *restart* (RST) operation. The head moves right along the tape until it takes any RST operation. RST implies that the restarting automaton places the look-ahead window over the left border of the tape and it completes one *cycle*. After performing a DEL/RST operation, the restarting automaton is unable to remember any step of computation that was performed already. We can say that it is a modification

S. Arumugam et al. (Eds.): ICTCSDM 2016, LNCS 10398, pp. 10–19, 2017.
DOI: 10.1007/978-3-319-64419-6_2

of the *list automaton* [7] and *forgetting automaton* [8]. Further, when each time the DEL operation is performed, the tape becomes smaller and smaller. A word u can be reduced to a word v if there is a cycle starting with u and ending with v. The computation ends by halting in an accepting or a rejecting state.

Insertion operations are introduced in [3] and based on these operations, insertion grammars are introduced in [4,5] and further studied in [6]. The motivation for insertion grammar comes from linguistic and as well from DNA processing and RNA editing. Informally, the insertion operation is defined as follows: If a string x is inserted between two parts w_1 and w_2 of a string w_1w_2 to get w_1xw_2, we call the operation *insertion*. The working nature of insertion grammar is counterpart to the functionality of contextual grammar [9], where based on the *selector* present in a string as a substring, the *contexts* are adjoined left and right of the substring.

In [1], it has been shown that restarting automaton with delete (simply, DRA) can represent the analyzer for characterizing the class of *contextual grammars with regular selector* (CGR). Also [2] showed that restarting automata recognize a family of languages which can be generated by certain type of contextual grammars, called *regular prefix contextual grammars with bounded infix* (RPCGBI). In this paper, we make a relationship between *restarting automaton* and *insertion languages*.

The *membership problem* for a language is defined as follows: Given a grammar G and a string w, whether w belongs to the language generated by G or not? In this paper, we introduce *1-normal DRA*. With the existing automaton - DRA, we introduce a variant of normal DRA where the DEL operation can be taken only once followed by restart in a cycle. We can say that 1-normal DRA is similar to *clearing restarting automata* [12].

The paper is organized as follows. Section 2 is Preliminaries that recall mainly the restarting automaton with delete operation (DRA) and insertion grammars. Section 3 introduces 1-normal DRA and discusses some properties of it. Section 4 discusses the relationship between the 1-normal DRA and insertion grammars. Section 5 discusses about some interesting properties of 1 Normal - DRA. Section 6 concludes the paper with some future work.

2 Preliminaries

Throughout the paper we will use the following notations. If Σ is an alphabet, then Σ^* denotes the set of all strings over Σ. For a string w, $|w|$ is the length of the string, sometimes called size of the string and $\emptyset$ denotes empty set. Any consecutive symbols of a string is called a substring. If a string x is a substring of y, then it is denoted by $x \in sub(y)$. A string $x \in \Sigma^*$ is called a scattered substring of a string $y \in \Sigma^*$ where $|x| \geq |y|$, then x can be obtained by omitting some symbols from y but maintaining the relative order of the remaining ones. For an automaton, the language accepted by M is denoted by $L(M)$ and for a given grammar γ, the language generated by γ is denoted by $L(\gamma)$.

2.1 Restarting Automaton with Delete Operation (DRA)

A restarting automaton with delete (denoted by DR-automaton or by DRA) is $M = (Q, \Sigma, \triangleleft, \triangleright, q_0, k, \delta)$ where Q is a finite set of states, Σ is the input alphabet, $\triangleleft, \triangleright$ are left and right borders respectively and $\triangleleft, \triangleright \notin \Sigma$, k is the size of the read-write window ($k \geq 1$).

The transition relation δ describes different types of transition steps which are given below. u' is assumed to be the content of the look-ahead window (and not necessarily the content of the tape).

- *MVR* - $(q', MVR) \in \delta(q, u')$, if M is in state q and sees a string u' where $u' \neq \triangleright$ in its look-ahead window, then this *MVR* step shifts the look-ahead window one position to the right and M enters into the state q'.
- *DEL* - $(q', v') \in \delta(q, u')$, if M is in state q and sees a string u' in its look-ahead window, deleting an item from the look-ahead window. u' is replaced by its scattered substring v' such that $|v'| < |u'|$. The border markers $\triangleleft, \triangleright$ must not disappear from the tape. After using the DEL operation the automaton can still read the remaining part of the tape also the automaton can place its head to the right of the just rewritten (deleted) string [1].
- *RST* - Restart. It causes M to move its look-ahead window to the left border marker $\triangleleft$ and re-enters into the initial state q_0.
- *ACCEPT* - *Accept* $\in \delta(q, u')$ where $q \in Q$. It gets into an accepting state.
- *REJECT* - If $\delta(q, u') = \emptyset$ (i.e., when δ is undefined), then M will reject.

A configuration of the automaton M is (u', q, v'), where $u' \in \{\triangleleft\Sigma^* \cup \lambda\}$ is the content from the *left border* till the position of the head, $q \in Q$ is the current state and $v' \in \{\triangleleft\Sigma^*\triangleright \cup \Sigma^*\triangleright\}$ is the content of the working list from the position of the head and to the right till the right end of the tape. In the initial configuration on an input word w, the control unit is in the fixed *initial state* $q_0 \in Q$, and the head is attached to the left border $\triangleleft$, i.e. $(\lambda, q_0, \triangleleft w \triangleright)$-scanning $\triangleleft$ and looking at the next $k-1$ symbols. We suppose that the states Q of the finite control are divided into two classes: the non-halting states (at least one instruction must be there which is applicable when the unit is in such a state) and the halting states (any computation ends by entering such a state), the halting states are further divided into the *accepting state* and the *rejecting state*.

In general, the restarting automaton is *non-deterministic*, i.e. there can be two or more instructions for a $\delta(q, u')$, it suggests that there can be more than one computation for an input string. Otherwise the automaton is said to be deterministic. Any finite computation of a DRA consists of certain phases. A phase, called a *cycle*, starts in a restarting configuration, the head moves along the tape and performing MVR, DEL operations until a RST operation is performed and thus a new restarting configuration is reached. If no further RST operation is performed, any finite computation necessarily finishes in a halting configuration -such phase is called *tail*.

[1] in our paper, we assume that after every DEL operation is immediately followed by RST, its forming DEL-RST.

The notation $u' \Rightarrow_M v'$ indicates that there exists a cycle in M starting in the initial configuration with the word u' and ending in the configuration having the word v', the relation $\Rightarrow_M^*$ is the reflexive and transitive closure of $\Rightarrow_M$. We say that u' becomes v' by M (or u' is reduced to v' by M) if $u' \Rightarrow_M v'$, we are certain that the word v' is strictly shorter than u' (v' is the scattered subword of u'). An input word w is accepted by M if there is a computation which starts in the initial configuration with w (bounded by borders $\lhd, \rhd$) on the list and finishes in an accepting configuration where the control unit is in one of the accepting states. $L(M)$ denotes the language consisting of all words accepted by M and we say that M recognizes the language $L(M)$.

A DEL step of an DRA may remove an arbitrary number of factors from the actual content of the look-ahead window. Therefore the following restriction has been included in DRA [1,2].

Definition 1 (**Normal DRA**). *A DRA is called normal if all the DEL operations are in the form $(q', v') \in \delta(q, u')$ where v' is a scattered substring of u', there exist words $x_1, x_2, x_3, x_4, x_5 \in \Sigma^*$ such that $u' = x_1x_2x_3x_4x_5$ and $v' = x_1x_3x_5$, that is two substrings of u' can be deleted.*

Proposition 1 (**Error preserving property of DRA**). *If $u' \Rightarrow_M^* v'$ and $u' \notin L(M)$ then $v' \notin L(M)$.*

2.2 Insertion Grammars

An Insertion grammar $\gamma = (T, A, I)$, where T is an *alphabet* set, A is a finite set of *strings* over T called *axioms*, I is the set of *insertion rules* of the form $(u, \lambda/x, v)$ where $u, v \in T^*$ and $x \in T^+$ which corresponds to the *rewriting rule* $uv \rightarrow uxv$,
Here u, v are called contexts and x is called inserted string for an insertion rule. As usual, $\Rightarrow^*$ denotes the *reflexive transitive closure* of $\Rightarrow$. A language $L(\gamma)$ generated by γ is defined by $L(\gamma) = \{w \in T^* \mid y \in A : y \Rightarrow^* w\}$.

3 1-Normal DRA

We first define 1-normal DRA. The functionality and the accepting configurations defined for DRA are the same for 1-normal DRA except the following changes. Normal DRA can delete at most two substrings from the current string but in this version at most one substring is deleted using DEL operation then it takes RST (restart) immediately without reading the remaining part of the tape, thus forming a new operation DEL-RST.

Definition 2 (**1 − Normal DRA**). *A restarting automata is called 1-normal DRA if all the DEL operations are in the form $(q', v') \in \delta(q, u')$ where v' is a scattered substring of u', there exist words $x_1, x_3 \in \Sigma^*, x_2 \in \Sigma^+$ such that $u' = x_1x_2x_3$ and $v' = x_1x_3$. In a cycle one substring can be deleted using DEL operation and RST is followed immediately.*

As insertion grammars do not contain non terminals, 1-normal DRA do not need to use any non terminal, so the error preserving property is satisfied for 1-normal DRA and correctness preserving property is satisfied for deterministic 1-normal DRA.

Before we go to analyze the relationship between 1-normal DRA and insertion grammars which is the objective of the paper, we first need to understand the relationship of DRA with contextual grammars [1]. External contextual grammars are introduced by S. Marcus in 1969 [9]. Internal contextual grammars [10] produce strings starting from an *axiom* and in each step *left context* and *right context* are adjoined to the string based on certain string called selector present as a substring in the derived string. u, v are called *left context* and *right context* respectively. For more details on contextual grammars, we refer to [11]. We recall that in insertion grammar, looking at the context (u, v), the string x is inserted. The selector in a contextual grammar can be of arbitrary type in nature, like regular, context free etc., but the strings u, v are finite. In insertion grammars all the strings u, v, x are finite. Normal DRA works in the opposite way of contextual grammars in accepting strings [1]. In a normal DRA M, w is given as an input. It checks the items of the look-ahead window with the contextual grammar G that any given rule P in G has been used or not. If it finds that any rule has been used then the automaton deletes the left and right context u, v and takes the RST operation, otherwise takes MVR and checks whether any rule in G can be applied. In this way, the automaton simulates the derivation of contextual grammar in reverse order and if the input string can be reduced back to the axiom z, it implies that the string w can be generated using the given grammar G, thus $w \in L(G)$. Here the *size* of the tape of the automaton M is same as the size of the string w. Step by step, the automaton M only deletes substrings of w, so the size of the tape becomes smaller and smaller. Tape size of 1-normal DRA will be $|w| + 2$ where the 2 is added for the left border $\lhd$ and the right border $\rhd$.

4 1-Normal DRA and Insertion Grammar

In this section we shall establish the relationship between 1-normal DRA and insertion grammars. We show that the membership problem for insertion languages can be solvable by the introduced 1-normal DRA. The paradigm of this version of 1-normal DRA is closely related to insertion grammars. Insertion grammar works just in the *opposite direction* of 1-normal DRA. The connection is established based on the following observation.

- For an insertion rule $(u, \lambda/x, v)$ where $u, v \in T^*$ and $x \in T^+$, the 1-normal DRA has to delete the substring x between u and v (this means that uxv is occurred as a substring in the given input string and the machine deletes this substring x). In that case, we informally say that an insertion rule is found/used in the look-ahead window as a substring.

Let M be 1-normal DRA. A reduction system induced by M is $RS(M) = (\Sigma^*, \Rightarrow_M)$. For each insertion grammar γ, we define a reduction system induced by γ as $RS(\gamma) = (T^*, \Rightarrow_\gamma^{-1})$ where $(u \Rightarrow_\gamma^{-1} v)$ iff $(v \Rightarrow_\gamma u), u, v \in T^*$.

With the above detail we will construct a 1-normal DRA M in such a way that if $z \Rightarrow_\gamma^* w$ then $w \Rightarrow_M^* z$ for $w, z \in T^*$, z-axiom, thus $RS(\gamma) = RS(M)$.

Let w be the input string given to 1-normal DRA. The automaton M checks the string of the look-ahead window of (size k) with the given grammar γ that any insertion rule from I has been found or not as a substring. If any insertion rule from I is found in the look-ahead window as a substring (uxv) then the automaton M deletes the inserted string $x \in T^+$ using the DEL operation.

Theorem 1. *For an insertion grammar γ, a 1-normal DRA M can be constructed in such a way that $RS(\gamma) = RS(M)$ and $L(\gamma) = L(M)$.*

Proof. Given an insertion-grammar $\gamma = (T, A, R)$, we have to construct a 1-normal DRA $M = (Q, \Sigma, \triangleleft, \triangleright, q_0, k, \delta)$, that accepts $L(\gamma)$ where

- $Q = \{q_0 = q', q, Accept, Reject\}$
- $\Sigma = T$ is the input alphabet
- $\triangleleft, \triangleright$ are left and right borders respectively and $\triangleleft, \triangleright \notin \Sigma$
- k is the size of the look-ahead window ($k \geq 1$).
- δ is defined as follows:

For an insertion rule of the form: $(x_1, \lambda/x_2, x_3)$ where $x_1, x_3 \in T^*, x_2 \in T^+$ (which offers a rewriting rule $x_1x_3 \rightarrow_\gamma x_1x_2x_3$), the instruction of the 1-normal DRA M will be $(q', v') \in \delta(q, u')$ where $u' = x_1x_2x_3, v' = x_1x_3$. Here u' is replaced by $v' : |v'| < |u'|$ where v' is a scattered substring of u', immediately followed by a RST instruction: RST $\in \delta(q, u')$ for any possible contents u' of the look-ahead window. If no insertion rule does belong to look-ahead window as a subword($uxv \notin u'$) and $\triangleright$ does not belong to look-ahead ($\triangleright \notin u'$) then the automaton takes MVR operation.

- ACCEPT- $Accept \in \delta(q, u')$ where $u' = \triangleleft z \triangleright, z \in A$.
- REJECT - $\delta(q, u') = \emptyset$. That is when δ is undefined. In other words, when 1-normal DRA is unable to take any of the DEL, MVR operations then the transition becomes undefined.

Size of the Look-ahead Window:
Size of the look-ahead window of M will be $k = max(k_c, k_b + 2)$ where k_c is the maximum length of the inserted string with its contexts - $k_c = max\{(|u| + |x| + |v|)$ where $(u, \lambda/x, v) \in I, u, v =$ contexts$\}$. k_b is the maximum axiom size- $k_b = max\{|z| : z \in A\}$. 2 is added there for the left border $\triangleleft$ and the right border $\triangleright$. The reason for 2 is added with k_b is to satisfy the accepting condition - $Accept \in \delta(q, u')$ where $u' = \triangleleft z \triangleright$ where $\triangleleft z \triangleright \leq k$.

1-normal DRA simulates the derivation of insertion-deletion grammar in reverse order, in case of insertion rule it deletes the inserted string using DEL instruction which is defined above. For insertion grammar the derivation starts from the axiom to the generated string, the automaton starts the reduction from the generated string to the axiom. If $z \Rightarrow_{\gamma}^{*} w$ then $w \Rightarrow_{M}^{*} z$ where $w, z \in T^*$, z-axiom, thus $RS(\gamma) = RS(M)$.

We have the following important result and the proof is obvious from Theorem 1, and from the discussions of above paragraphs of Theorem 1.

Theorem 2. *The membership problem for insertion languages can be solved by 1-normal DRA.*

5 The Power of 1-Normal DRA

In this section, we discuss the power of 1-normal DRA and discuss some interesting properties.

Theorem 3. *All regular languages can be recognized by 1-normal DRA, i.e. for each regular language L there exists a M such that $L(M) = L \cup \lambda$.*

Proof. Given a regular grammar $\gamma = (N, T, P, S)$ where N is the finite set of non-terminals,T is the finite set of terminals, P is the finite set of production rules of the following form:$A \rightarrow aB$ where $A \in N$, $A \rightarrow b$ where $b \in T$, S is the starting symbol. Now we have to construct a 1-normal DRA $M = (Q, \Sigma, \triangleleft, \triangleright, q_0, k, \delta)$, that accepts $L(\gamma)$ where $Q = \{q_0 = q', q, Accept, Reject\}$, $\Sigma = N \cup T$ is the input alphabet, $\triangleleft, \triangleright$ are left and right borders respectively and $\triangleleft, \triangleright \notin \Sigma$, k is the size of the look-ahead window ($k \geq 1$), δ is defined as follows: For a regular grammar rule is of the form: $A \rightarrow aB$ where $A \in N$, the instruction of the 1-normal DRA M will be $(q', v') \in \delta(q, u')$ where $u' = aB, v' = A$. Here u' is replaced by $v' : |v'| < |u'|$ where v' is a scattered substring of u', immediately followed by a RST instruction: RST $\in \delta(q, u')$ for any possible contents u' of the look-ahead window. Also, if a regular grammar rule is $A \rightarrow b$ where $b \in T$ then the instruction of 1-normal DRA will be $(q', v') \in \delta(q, u')$ where $u' = b, v' = A$. In the same way, u' is replaced by $v' : |v'| < |u'|$ where v' is a scattered substring of u', immediately followed by a RST instruction: RST $\in \delta(q, u')$.
If no regular rule does belong to look-ahead window as a subword ($aB \notin u'$ or $b \notin u'$) and $\triangleright$ does not belong to look-ahead ($\triangleright \notin u'$) then the automaton takes MVR operation.

- ACCEPT- $Accept \in \delta(q, u')$ where $u' = \triangleleft S \triangleright$, S is the starting symbol.
- REJECT - $\delta(q, u') = \emptyset$. That is when δ is undefined. In other words, when 1-normal DRA is unable to take any of the DEL, MVR operations then the transition becomes undefined.

Size of the Look-ahead Window:
Size of the look-ahead window of M will be $k = max(k_c, k_b + 2)$ where k_c is the length of the right-hand side of the production, $k_c = 2$. k_b is the size of the axiom - $k_b = 1$. 2 is added there in order to satisfy the accepting condition - $Accept \in \delta(q, u')$ where $u' = \triangleleft S \triangleright$.

Theorem 4. *There are context free languages which cannot be recognized by* 1-*normal DRA.*

Proof. We conclude Theorem 4 by focusing on Lemmas 1 and 2.

Lemma 1. *The language* $L_1 = \{p^n r q^n \mid n \geq 0\} \cup \{\lambda\}$ *cannot be recognized by* 1*-normal DRA.*

Proof. 1-normal DRA can delete at most one string in a cycle, so it will delete all p in first n cycles and from $(n+1)th$ cycle it will start deleting q. Actually in this case 1-normal DRA cannot delete two substrings in a same cycle, so it is unable to keep track of the equality of $p's$ and $q's$.

As we have seen in Lemma 1 that not all context-free languages are recognized by a 1-normal DRA. We still could characterize CFL using 1-normal DRA using inverse homomorphism and Greibach's hardest context-free language [2]. Greibach constructed a context-free language H [12], such that:

- Any context-free language can be parsed in whatever time or space it takes to recognize H.
- Any context-free language L can be obtained from H by an inverse homomorphism. That is, for each context-free language $L \subseteq \Sigma^*$, there exist a homomorphism ρ so that $L = \rho^{-1}(H)$. The definition of the Greibach's language follows. Let $\Sigma = \{x_1, x_2, \bar{x_1}, \bar{x_2}, \#, c\}$.
  Define $H = \{\lambda \cup \{a_1 c b_1 c z_1 d ... a_n c b_n c z_n d \mid n \geq 1, b_1 ... b_n \in \#D, a_i, z_i \in \Sigma^*\}$, for all i, $1 \leq i \leq n, b_1 \in \{x_1, x_2, \bar{x_1}, \bar{x_2}\}^*, b_i \in \{x_1, x_2, \bar{x_1}, \bar{x_2}\}^*$, for all $i \geq 2\}$.
- (Note that a_i and z_i can contain c and $\#$). D is a semi-Dyck language over the alphabet $\{x_1, x_2, \bar{x_1}, \bar{x_2}$, generated by the grammar with one non terminal S and the set of rules: $S \Rightarrow \lambda \mid SS \mid a_1 S \bar{a_1} \mid a_2 s \bar{a_2}$. Clearly it is a context free language.

Lemma 2. H *is not accepted by* 1*-normal DRA.*

Proof. Consider the language H as given above. Then, H cannot be accepted by any 1-normal DRA. The main problem of recognition H by 1-normal DRA is selection of $b_1, b_2, ..b_n$(see the formal definition of H). Unfortunately no 1-normal DRA can recognize H. We need to construct $M = (Q, \Sigma, \triangleleft, \triangleright, q_0, k, \delta)$, such that $L(M) = H$. Apparently, $w = c\#x_1^m cdc\bar{x}_1^m cd \in H$. Let the accepting computation will $w_1 \Rightarrow_M w' \Rightarrow_M ... \Rightarrow_M$ Axiom. Firstly, there must be a deleted substring of the form $x_1^r cdc\bar{x}_2^s$ for some $0 \leq r, s \leq m$. Here it is easy to see that $r = s$. The first applied instruction of 1-normal DRA in order to recognize will be $(q', v') \in \delta(q, u')$ where $u' = x_1^\alpha x_1^r cdc\bar{x}_1^r \bar{x}_1^\beta, v' = x_1^\alpha \bar{x}_1^\beta$. Now consider the word $w = c\#x_1^{m+1} x_1^r cdc\bar{x}_1^r \bar{x}_1^m c\#x_1\bar{x}_1 cd$. From here we can easily conclude $w \notin H$, but $w' = c\#x_1^{m+1}\bar{x}_1^m c\#x_1\bar{x}_1 cd$ is in H. So, contradiction of error preserving property.

Corollary 1. *(a)* $\mathcal{L}$*(1-normal DRA)*$\subset \mathcal{L}(Normal - DRA)$
(b) $CFL - \mathcal{L}$*(1-normal DRA)*$\neq \emptyset$

Proof. In Lemma 1, the language $L_1 = \{p^n r q^n \mid n \geq 0\} \cup \{\lambda\}$ can be recognized by normal DRA, so from this fact easily we can conclude corollary(a).

Let $L_2 = \{p^n q^n \mid n > 0\}$ and $L_3 = \{p^n q^{2n} \mid n > 0\}$ be two sample languages. It is easy to see from the Theorem 2 that both L_2, L_3 are recognized by 1-Normal DRA, in this way we can conclude corollary(b).

Theorem 5. *$\mathcal{L}$(1-normal DRA) is not closed under union and concatenation.*

Proof. Languages $L_2 \cup L_3$ and $L_2 \cdot L_3$ cannot be recognized by 1-Normal DRA. For a contradiction let us suppose that, there exist a 1-Normal DRA M such that $L(M) = L_2 \cup L_3$ ($L(M) = L_2 \cdot L_3$, respectively). Let K be the size of the look-ahead window. Let $p^n q^n \Rightarrow_M p^{n-s} b^{n-t}$ be the first step of an accepting condition, where $s, t > 0, s + t > 0$. Suppose when $d = p$ comes in the look-ahead window then 1-Normal DRA takes DEL-RST operation and delete q where $|d| \leq |k|$. Since we set n arbitrarily large where s, t are constants we get necessarily $s = t$. Now if $d = p^\alpha p^\eta q^\eta q^\beta$ then 1-Normal DRA takes DEL-RST operation and delete $p^\eta q^\eta$ where $\alpha + \beta + 2\eta \leq |k|$. Then $p^{m+\eta} q^{2m+\eta} \Rightarrow_M p^m q^{2m}$ which is a contradiction with the error preserving property as $p^{m+\eta} q^{2m+\eta}$ is not in $L_2 \cup L_3$ (and not in $L_2 \cdot L_3$ respectively).

Theorem 6. *There are non context free languages which can be recognized by 1-normal DRA.*

Proof. 1-normal DRA. M recognizing a language that is not context-free $(\{(ab)^{2^m} \mid m \geq 0\})$. The instructions of 1 - normal DRA are given below.

- : DEL-RST :$(q', v') \in \delta(q, u')$ where $u' = abb, v' = ab$.
- : DEL-RST :$(q', v') \in \delta(q, u')$ where $u' = bab, v' = bb$.
- : *Accept* $\in \delta(q, u')$ where $u' = \triangleleft z \triangleright$ where $z = ab$

The computation as follows, $\triangleleft ab\underline{a}babababababab\triangleright \Rightarrow_M \triangleleft a\underline{b}babababababab\triangleright \Rightarrow_M \triangleleft ab\underline{a}babababababab\triangleleft \Rightarrow_M \triangleleft a\underline{b}bababababab\triangleleft \Rightarrow_M \triangleleft ab\underline{a}bababababab\triangleright \Rightarrow_M \triangleleft a\underline{b}babababab\triangleright \Rightarrow_M \triangleleft ab\underline{a}babababab\triangleright \Rightarrow_M \triangleleft a\underline{b}bababab\triangleright \Rightarrow_M \triangleleft ab\underline{a}bababab\triangleright \Rightarrow_M \triangleleft a\underline{b}babab\triangleright \Rightarrow_M \triangleleft ab\underline{a}babab\triangleright \Rightarrow_M \triangleleft a\underline{b}bab\triangleright \Rightarrow_M \triangleleft ab\underline{a}b\triangleright \Rightarrow_M \triangleleft a\underline{b}b\triangleright \Rightarrow_M \triangleleft ab\triangleright$, accepted.

Theorem 7. *$\mathcal{L}$(1-normal DRA) is not closed under homomorphism.*

Proof. Consider $L = \{p^n q^n r^m s^{2m} \mid n, m \geq 0\}$ recognized by $\mathcal{L}(1-normal DRA)$ and the homomorphism $h : p \mapsto p, q \mapsto q, r \mapsto r, s \mapsto q$.

It is easy to see that each of the following languages:
$L_4 = \{p^n r q^n \mid n \geq 0\} \cup \{p^n q^n \mid n \geq 0\}, L_5 = \{p^n r q^m \mid n, m \geq 0, \} \cup \{\lambda\}$, $L_6 = \{p^m q^m \mid m \geq 0\}$ can be recognized by a 1-normal DRA. Using these languages we can show several non-closure properties of 1-normal DRA.

Theorem 8. *$\mathcal{L}$(1-normal DRA) is not closed under intersection, intersection with a regular language, and set difference.*

Proof. Intersection part follows the equality $L_1 = L_4 \cap L_5$ and Lemma 1. In order to proof the next part, just notice that L_5 is a regular language. Set differences the part follows the equality $L_3 = (L_4 - L_6) \cup \{\lambda\}$.

6 Conclusion

In this paper, we have introduced 1-normal DRA. We have solved the membership problem of insertion languages. We saw that s insertion grammars do not contain non terminals, 1-normal DRA do not need to use any non terminal, so the error preserving property is satisfied for 1-normal DRA and correctness preserving property is satisfied for deterministic 1-normal DRA.

There is scope of future work. H in Lemma 2, can be accepted by introducing auxiliary symbol to 1-normal DRA. Also, in case of running time, as here we are using non deterministic 1-normal DRA, we are unable to comment about the polynomial time complexity solution. We can solve the membership problem of insertion languages in polynomial time by extending our work.

Acknowledgments. Tool Development - Poorna Chandra Tejasvi, Btech third year student of Computer Science and Engineering, Christ University Faculty of Engineering, Bangalore.

References

1. Jancar, P., Mraz, F., Platek, M., Prochazka, M., Vogel, J.: Deleting automata with a restart operation. In: Bozapalidis, S. (ed.) Proceedings of Developments in Language language theory III(97), Thessaloniki, pp. 191–202 (1998)
2. Jancar, P., Mraz, F., Platek, M., Prochazka, M., Vogel, J.: Restarting automata, marcus grammars and context-free languages. In: Dassow, J., Rozenberg, G., Salomaa, A. (eds.) Developments in Language Theory, pp. 102–111. World Scientific Publishing (1996)
3. Haussler, D.: Insertion and Iterated Insertion as Operations on Formal Languages. Ph.D. Thesis, University of Colorado, Boulder (1982)
4. Haussler, D.: Insertion languages. Inf. Sci. **131**(1), 77–89 (1983)
5. Galiukschov, B.S.: Semicontextual grammars (in Russian). MAT. Logica i Mat. Ling. 38–50 (1981)
6. Kari, L., Thierrin, G.: Contextual insertions/deletions and computability. Inf. Comput. **131**(1), 47–61 (1996)
7. Chytil, M.P., Platek, M., Vogel, J.: A note on the Chomsky hierarchy. Bull. EATCS **27**, 23–30 (1985)
8. Jancar, P., Mráz, F., Plátek, M.: A taxonomy of forgetting automata. In: Borzyszkowski, A.M., Sokołowski, S. (eds.) MFCS 1993. LNCS, vol. 711, pp. 527–536. Springer, Heidelberg (1993). doi:10.1007/3-540-57182-5_44
9. Marcus, S.: Contextual Grammars. Revue Roumane de Mathematiques Pures et Appliques 14(10), 1525–1534 (1969)
10. Paun, G., Nguyen, X.M.: On the inner contextual grammars. Rev. Roum. Pures. Appl. **25**, 641–651 (1980)
11. Paun, G.: Marcus Contextual Grammars. Kluwer Academic Publishers, Dordrecht (1997)
12. Cerno, P., Mráz, F.: Clearing restarting automata. Fundam. Inform. **104**, 17–54 (2010)