Connecting Partial Words and Regular Languages

Jürgen Dassow[1], Florin Manea[2]* and Robert Mercaş[1]

[1] Otto-von-Guericke-Universität Magdeburg, Fakultät für Informatik, PSF 4120, D-39016 Magdeburg, Germany,
dassow@iws.cs.uni-magdeburg.de, robertmercas@gmail.com

[2] Christian-Albrechts-Universität zu Kiel, Institut für Informatik, D-24098 Kiel, Germany,
flm@informatik.uni-kiel.de

Abstract. We initiate a study of languages of partial words related to regular languages of full words. First, we study the possibility of expressing a regular language of full words as the image of a partial-words-language through a substitution that only replaces the hole symbols of the partial words with a finite set of letters. Results regarding the structure, uniqueness and succinctness of such a representation, as well as a series of related decidability and computational-hardness results, are presented. Finally, we define a hierarchy of classes of languages of partial words, by grouping together languages that can be connected in strong ways to regular languages, and derive their closure properties.

Keywords: partial word, regular language, finite automaton, language of partial words.

1 Introduction

Two DNA strands attach one to the other, normally, in a complementary way according to their nucleotides. That is, each purine, A or G, creates a hydrogen bond with one complementary pyrimidine, T or C, respectively. But, sometimes, it is the case that this process goes wrong, allowing G-T bonds. Starting from this situation, and motivated by the need of having a way to recover (as much as possible) and work with a correct DNA sequence, Berstel and Boasson [1] suggested the usage of partial words as a suitable mathematical model. Partial words are words that beside regular letters contain an extra "joker" symbol, also called "hole" or "do-not-know" symbol, that matches all symbols of the original alphabet, which were investigated already in the 1970s [3]. Going back to the initial example, one could recover the information regarding a DNA sequence from the badly bonded pair of DNA strands by associating actual letters to the positions where the bonds were correct and holes to the positions where the bonds were not correctly formed. Besides the above motivation, partial words may find applications in other fields, as well; for instance, they can be seen

* The work of Florin Manea is supported by the DFG grant 582014.

as data sequences that are corrupted either by white noise or other external factor, or even incomplete or insufficient information, that one needs in some process. In the last decade a lot of combinatorial and algorithmic properties of partial words have been investigated (see the survey [2], and the references therein). Surprisingly, so far, no study of classes of languages of partial words (or sets of partial words that have common features) was performed. The only work on a connected topic, that we are aware of, is [4]. There, the concept of restoration of punctured languages and several similarity measures between full-words-languages, related to this concept, were investigated. More precisely, puncturing a word means replacing some of its letters with holes; from a language of punctured words, its restoration was obtained by taking all the languages that can be punctured to obtain the respective language. The results of [4] regarded classes of full-words-languages defined by applying successively puncturing and restoration operations to classes of languages from the Chomsky hierarchy.

The study of the class of regular languages, the most restrictive class of the Chomsky-hierarchy, has been one of the central topics in theoretical computer science. This class of languages, defined usually either as the class of languages accepted by finite automata or as the class of languages described by regular expressions, was extensively studied throughout the last seventy years (starting from the early 1940s) and, besides its impact in theory (mostly in language theory, but also in complexity theory, for instance), it was shown to have a wide range of applications. Regular languages, and the various mechanisms used to specify them, were used, for instance, in compilers theory, circuit design, text editing, pattern matching, formal verification, DNA computing, or natural language processing (see [7]).

In this work, we aim to establish a stronger connection between the attractive notions mentioned above: partial words, on one side, and regular languages, on the other side. First, we show how we can (non-trivially) represent every regular language as the image of a regular language of partial words through a substitution that defines the letters that may replace the hole (called $\diamond$-substitution, in the following). Moreover, we show that such a representation can be useful: for some regular languages, there exist deterministic finite automata accepting languages of partial words that represent the full-word-language and are exponentially more succinct than the minimal deterministic finite automaton accepting that language. Unfortunately, it may also be the case when the minimal non-deterministic finite automaton accepting a language is exponentially more succinct than any deterministic automaton accepting a language of partial words representing the same language. Generally, automata accepting languages of partial words representing a given full-words language can be seen as intermediate between the deterministic finite automata and the non-deterministic automata accepting that language. We also present a series of algorithmic and complexity results regarding the possibility of representing a regular language as the image of a language of partial words through a $\diamond$-substitution.

Motivated by the above results, that connect in a meaningful way languages of partial words to regular languages of full words, and by the theoretical interest of studying systematically such languages, we define a series of classes of languages of partial words. Each of these classes contains languages that can be placed in a particular strong relation with the regular languages. Further, we investigate these classes from a language theoretic point of view, show that they form a hierarchy, and establish their closure properties.[3]

We begin the paper with a series of basic definitions. Let V be a non-empty finite set of symbols called an *alphabet*. Each element $a \in V$ is called a *letter*. A *full word* (or, simply, word) over V is a finite sequence of letters from V while a *partial word* over V is a finite sequence of letters from $V \cup \{\diamond\}$, the alphabet V extended with the distinguished hole symbol $\diamond$. The *length* of a (partial) word u is denoted by $|u|$ and represents the total number of symbols in u; the number of occurrences of a symbol a in a (partial) word w is denoted $|w|_a$. The *empty (partial) word* is the sequence of length zero and is denoted by λ. We denote by V^* (respectively, $(V \cup \{\diamond\})^*$) the set of words (respectively, partial words) over the alphabet V and by V^+ (respectively, $(V \cup \{\diamond\})^+$) the set of non-empty words (respectively, non-empty partial words) over V. The catenation of two (partial) words u and v is defined as the (partial) word uv. Recall that V^* (where the alphabet V may include the $\diamond$ symbol) is the free monoid generated by V, under the operation of catenation of words; the unit element in this monoid is represented by the empty word λ. A language L of full words over an alphabet V is a subset of V^*; a language of partial words L over an alphabet V (that does not contain the $\diamond$ symbol) is a subset of $(V \cup \{\diamond\})^*$. Given a language L we denote by $\mathbf{alph}(L)$ (the alphabet of L) the set of all the letters that occur in the words of L; for the precision of the exposure, we say that a language L of full (respectively, partial) words is over V, with $\diamond \notin V$, if and only if $\mathbf{alph}(L) = V$ (respectively, $\mathbf{alph}(L) = V \cup \{\diamond\}$). For instance, $L = \{abb, ab\diamond\}$ has $\mathbf{alph}(L) = \{a, b, \diamond\}$, thus, is a language of partial words over $\{a, b\}$. Note that the catenation operation can be extended to languages; more precisely, if L_1 and L_2 are languages over V, we define their catenation $L_1 L_2 = \{w_1 w_2 \mid w_1 \in L_1, w_2 \in L_2\}$.

Let u and v be two partial words of equal length. We say that u is *contained* in v, denoted by $u \sqsubset v$, if $u[i] = v[i]$ for all $u[i] \in V$; moreover, u and v are *compatible*, denoted by $u \uparrow v$, if there exists a word w such that $u \sqsubset w$ and $v \sqsubset w$. These notions can be extended to languages. Let L and L' be two languages of partial words with $\mathbf{alph}(L) \cup \mathbf{alph}(L') = V \cup \{\diamond\}$ and $\diamond \notin V$. We say that L is *contained* in L', denoted $L \sqsubset L'$, if, for every word $w \in L$, there exists a word $w' \in L'$ such that $w \sqsubset w'$. We say that L is *compatible* to L', denoted $L \uparrow L'$, if, for each $w \in L$, there exists $w' \in L'$ such that $w \uparrow w'$ and, for each $v' \in L'$, there exists $v \in L$ such that $v' \uparrow v$.

A substitution is a mapping $h : V^* \to 2^{U^*}$ with $h(xy) = h(x)h(y)$, for $x, y \in V^*$, and $h(\lambda) = \{\lambda\}$; h is completely defined by the values $h(a)$ for all $a \in V$. A morphism is a particular type of a substitution for which $h(a)$ contains exactly one element for all $a \in V$; i.e., a morphism is a function $h : V^* \to U^*$ with $h(xy) = h(x)h(y)$ for $x, y \in V^*$. A $\diamond$-substitution over V is a substitution

[3] A technical appendix containing full proofs of our results can be found at the webpage: `https://www.informatik.uni-kiel.de/zs/pwords`.

with $h(a) = \{a\}$, for $a \in V$, and $h(\diamond) \subseteq V$. Here we assume that $\diamond$ can replace any symbol of V.

In this paper, DFA stands for deterministic finite automaton and NFA for non-deterministic finite automaton; the language accepted by a finite automaton M is denoted $L(M)$. Also, the set of all the regular languages is denoted by **REG**; by $\mathbf{REG_{full}}$, we denote the set of all the regular languages of full words. Further definitions regarding finite automata and regular languages can be found in [7], while partial words are surveyed in [2].

2 Definability by Substitutions

Let us begin our investigation by presenting several results regarding the way regular languages can be expressed as the image of a language of partial words through a substitution.

Lemma 1. *Let $L \subseteq (V \cup \{\diamond\})^*\{\diamond\}(V \cup \{\diamond\})^*$ be a language of partial words and let σ be a $\diamond$-substitution over V. There exists a language L' such that $\sigma(L) = \sigma(L')$ and $|w|_\diamond = 1$ for all $w \in L'$.*

Lemma 2. *Let L be a regular language over V and let σ be a $\diamond$-substitution over V. Then there exists a maximal (with respect to set inclusion) language $L' \subseteq L$ which can be written as $\sigma(L'')$, where L'' is a language of partial words such that any word in L'' has exactly one hole. Moreover, L' and L'' are regular languages and, provided that L is given by a finite automaton accepting it, one can algorithmically construct a finite automaton accepting L' and L''.*

Proof. Let $\sigma(\diamond) = V'$.

We start by noting that a word w belongs to $\sigma(L'')$ for a language of partial words L'', whose elements contain at least one hole each, if and only if there exist the words x and y such that $w = xay$, for some $a \in V'$ and $\{x\}V'\{y\} \subseteq L$.

Now let $M = (Q, V, q_0, F, \delta)$ be a DFA accepting L.

Let $q \in Q$. We define the language R_q as follows. A word w of L is in R_q if and only if there exists the partial word $x\diamond y$, compatible with w, with $\delta(q_0, x) = q$, $S = \delta(q, V') \subseteq Q$, $\delta(S, y) \subseteq F$. Basically, R_q is the set of the words for which there exists a compatible partial word $x\diamond y$ with exactly one hole, such that x labels a path from q_0 to q in A and any word from $\{x\}V'\{y\}$ is in L.

Clearly, $R_q = \{x \mid x \in V^*, \delta(q_0, x) = q\}V'\{y \mid y \in V^*, \delta(q', y) \in F$ for all $q' \in \delta(q, V)\}$. It follows that R_q is regular and an automaton accepting this language can be constructed starting from M. Moreover, $R_q = \sigma(H_q)$, for $H_q = \{x \mid x \in V^*, \delta(q_0, x) = q\}\{\diamond\}\{y \mid y \in V^*, \delta(q', y) \in F$ for all $q' \in \delta(q, V)\}$.

Take now $L' = \cup_{q \in Q} R_q$ and $L'' = \cup_{q \in Q} H_q$. Then L' is regular, as all the languages R_q are regular, and $L' = \sigma(L'')$. We only have to show that L' is maximal. If there exists $L_1 \subset L$ and a language of partial words L_2, whose elements contain exactly one hole each, such that $\sigma(L_2) = L_1$, then for every word w of L_1 there exist the words x and y such that $x\diamond y \in L_2$ and $w \in \sigma(x\diamond y) = \{x\}V'\{y\} \subseteq \sigma(L_2) = L_1$. Thus, $w \in R_q$ for $q = \delta(q_0, x)$. Therefore, $L_1 \subseteq L'$. □

Note that the sets R_q with $q \in Q$ are not a partition of L, as they are not necessarily mutually disjoint.

Next we introduce two relations connecting partial-words-languages and full-words-languages.

Definition 1. *Let $L \subseteq V^*$ be a language and σ be a $\diamond$-substitution over V. We say that L is σ-defined by the language L', where $L' \subseteq (V \cup \{\diamond\})^*$ is a partial-words-language, if $L = \sigma(L')$. Moreover, we say that L is* essentially σ-defined *by L', where $L' \subseteq (V' \cup \{\diamond\})^*$, if $L = \sigma(L')$ and every word in L' contains at least a $\diamond$-symbol.*

Obviously, for any regular language L over V, there is a regular language L' of partial words and a $\diamond$-substitution σ over V such that $\sigma(L') = L$, i.e., L is σ-defined by L'. For instance, take the set L' of the words obtained by replacing in the words of L some occurrences of a symbol $a \in V$ by $\diamond$, and the $\diamond$-substitution σ over V that maps $\diamond$ to $\{a\}$. More relevant ways of defining a regular language, in the sense of Definition 1, are presented in this section. We begin by characterizing the essentially definable languages.

Assume that the regular language $L \subseteq V^*$ is essentially σ-definable for some $\diamond$-substitution σ over V. Then $\sigma(L') = L$ for some appropriate language L' such that any word of L' contains at least a hole. By Lemma 1, we get that there is a regular language L'' of partial words such that $\sigma(L'') = \sigma(L')$ and each word of L'' contains exactly one hole. Now by Lemma 2 and its proof we get immediately the following characterisation of σ-definable languages.

Theorem 1. *Let L be a regular language of full words over V and σ a $\diamond$-substitution over V. Then L is essentially σ-definable if and only if $L = \bigcup_{q \in Q} R_q$ (where R_q is given in the proof of Lemma 2).* □

We also easily get the following decidability results.

Theorem 2. *i) Given a regular language L over V and a $\diamond$-substitution σ over V, it is decidable whether L is essentially σ-definable.*
ii) Given a regular language L over V, one can algorithmically identify all $\diamond$-substitutions σ for which L is essentially σ-definable.

Proof. By the previous results, testing whether L is essentially σ-definable is equivalent to testing whether L and $L' = \cup_{q \in Q} R_q$ are equal. Because the equality of two regular languages is decidable, the first statement follows. The second statement follows by an exhaustive search in the (finite) set of all $\diamond$-substitutions σ over V for those that essentially define L. □

The following consequence of Lemma 2 is worth noting, as it provides a canonical non-trivial representation of regular languages.

Theorem 3. *Given a regular language $L \subseteq V^*$ and a $\diamond$-substitution σ over V, there exists a unique regular language $L_\diamond$ of partial words that fulfils the following three conditions: (i) $L = \sigma(L_\diamond)$, (ii) for any language L_1 with $\sigma(L_1) = L$ we have $\{w \mid w \in L_1, |w|_\diamond \geq 1\} \sqsubset \{w \mid w \in L_\diamond, |w|_\diamond \geq 1\}$, and (iii) $(L_\diamond \cap V^*) \cap \sigma(\{w \mid w \in L_\diamond, |w|_\diamond \geq 1\}) = \emptyset$.*

Proof. Using the sets defined in Lemma 2, take $L_\diamond = L'' \cup (L \setminus L')$. The conclusion follows easily. □

Motivated by this last result, we now turn to the descriptional complexity of representing a regular language of full words by regular languages of partial words. We are interested in the question whether there is a regular language $L \subseteq V^*$, a regular language $L' \subseteq (V \cup \{\diamond\})^*$, and a $\diamond$-substitution σ over V with $\sigma(L') = L$ such that the minimal DFA accepting L' has a (strictly) lower number of states than the minimal DFA accepting L? In other words, are there cases when one can describe in a more succinct way a regular language via a language of partial words and a substitution that define it? Moreover, can we decide algorithmically whether for a given regular language L there exist a language of partial words and a substitution providing a more succinct description of L?

Let L be a regular language of full words over V. We denote by $\min_{DFA}(L)$ the number of states of the (complete) minimal DFA accepting L. Furthermore, let $\min_{NFA}(L)$ denote the number of states of a minimal NFA accepting L. Moreover, for a regular language L let $\min^\diamond_{DFA}(L)$ denote the minimum number of states of a (complete) DFA accepting a regular language $L' \subseteq (V \cup \{\diamond\})^*$ (where $\diamond$ is considered as an input symbol) for which there exists a $\diamond$-substitution σ over V such that $\sigma(L') = L$.

We have the following relation between the defined measures.

Theorem 4. *i) For every regular language L we have*

$$\min_{DFA}(L) \geq \min^\diamond_{DFA}(L) \geq \min_{NFA}(L).$$

ii) There exist regular languages L such that

$$\min_{DFA}(L) > \min^\diamond_{DFA}(L) > \min_{NFA}(L).$$

By the previous result one can see that, for certain substitutions σ, minimal DFAs accepting languages of partial words that σ-define a given full-words-regular language can be seen as intermediate between the minimal DFA and the minimal NFA accepting that language: they provide a succinct representation of that language, while having a limited non-determinism.

In fact, one can show that the differences $\min_{DFA}(L) - \min^\diamond_{DFA}(L)$ and $\min^\diamond_{DFA}(L) - \min_{NFA}(L)$ can be arbitrarily large; more precisely, we may have an exponential blow-up with respect to both relations.

Theorem 5. *Let n be a natural number, $n \geq 3$. There exist regular languages L and L' such that* $\min^\diamond_{DFA}(L) \leq n + 1$ *and* $\min_{DFA}(L) = 2^n - 2^{n-2}$ *and* $\min_{NFA}(L') \leq 2n + 1$ *and* $\min^\diamond_{DFA}(L') \geq 2^n - 2^{n-2}$.

The following remark provides an algorithmic side of the results stated above.

Remark 1. Given a DFA accepting a regular language L we can construct algorithmically a DFA with $\min^\diamond_{DFA}(L)$ states, accepting a regular language of partial words L', and a $\diamond$-substitution σ over $\mathbf{alph}(L)$, such that L is σ-defined by L'. By exhaustive search, we take a DFA M with at most $\min_{DFA}(L)$

states, whose transitions are labelled with letters from an alphabet included in $\mathbf{alph}(L) \cup \{\diamond\}$, and a $\diamond$-substitution σ over $\mathbf{alph}(L)$. We transform M into an NFA accepting $\sigma(L(M))$ by replacing the transitions labelled with $\diamond$ by $|\sigma(\diamond)|$ transitions labelled with the letters of $\sigma(\diamond)$, respectively. Next, we construct the DFA equivalent to this NFA, and check whether it accepts L or not (that is, $\sigma(L(M)) = L$). From all the initial DFAs we keep those with minimal number of states, since they provide the answer to our question. It is an open problem whether such a DFA can be obtained by a polynomial time deterministic algorithm; however, we conjecture that the problem is computationally hard.

We conclude by showing the hardness of a problem related to definability.

Theorem 6. *Consider the problem P: "Given a DFA accepting a language L of full words, a DFA accepting a language L' of partial words, and a $\diamond$-substitution σ over* $\mathbf{alph}(L)$*, decide whether $\sigma(L') \neq L$." This problem is NP-hard.*

Proof. In [6], the following problem was shown to be NP-complete:
P': "Given a list of partial words $S = \{w_1, w_2, \ldots, w_k\}$ over the alphabet V with $|V| \geq 2$, each partial word having the same length L, decide whether there exists a word $v \in V^L$ such that v is not compatible with any of the partial words in S."

We show here how problem P' can be reduced in polynomial time by a many-one reduction to problem P. Indeed, take an instance of P': a list of partial words $S = \{w_1, w_2, \ldots, w_k\}$ over the alphabet V with $|V| \geq 2$, each having the same length L. We can construct in polynomial time a DFA M accepting exactly the language of partial words $\{w_1, w_2, \ldots, w_k\}$. Also, we can construct in linear time a DFA M' accepting the language of full words V^L. It is clear that for $L(M)$ and the substitution σ, mapping the letters of V to themselves and $\diamond$ to V, we have $\sigma(L(M)) \neq V^L$ (that is, the answer to the input M, M' and σ of problem P is positive) if and only if the answer to the given instance of P' is also positive. Since solving P' is not easier than solving P, we conclude our proof. □

Theorem 6 provides a simple way to show the following well known result.

Corollary 1. *The problem of deciding whether a DFA M and an NFA M' accept different languages is NP-hard.*

3 Languages of partial words

While the results of the last section study the possibility and efficiency of defining a regular language as the image of a (regular) language of partial words, it seems interesting to us to take an opposite point of view, and investigate the languages of partial words whose images through a substitution (or all possible substitutions) are regular. Also, languages of partial words compatible with at least one regular language (or only with regular languages) seem worth investigating.

The definitions of the first three classes considered in this section follow the main lines of the previous section. We basically look at languages of partial words that can be transformed, via substitutions, into regular languages.

Definition 2. *Let L be a language of partial words over V.*
1. *We say that L is $(\forall\sigma)$-regular if $\sigma(L)$ is regular for all the $\diamond$-substitutions σ over alphabets that contain V and do not contain $\diamond$.*
2. *We say that L is* **max***-regular if $\sigma(L)$ is regular, where σ is a $\diamond$-substitution over V' with $\sigma(\diamond) = V'$, and $V' = V$ if $V \neq \emptyset$, and V' is a singleton with $\diamond \notin V'$, otherwise.*
3. *We say that L is $(\exists\sigma)$-regular if there exists a $\diamond$-substitution σ over a non-empty alphabet V', that contains V and does not contain $\diamond$, such that $\sigma(L)$ is regular.*
The classes of all $(\forall\sigma)$-regular, **max***-regular, and $(\exists\sigma)$-regular languages are denoted by $\mathbf{REG}_{(\forall\sigma)}$, $\mathbf{REG}_{\mathbf{max}}$, and, respectively, $\mathbf{REG}_{(\exists\sigma)}$.*

We consider, in the following, two classes of languages of partial words that are defined starting from the concept of compatibility.

Definition 3. *Let L be a language of partial words over V.*
4. *We say that L is $(\exists)$-regular if exists a regular language L' of full words such that $L \uparrow L'$.*
5. *We say that L is $(\forall)$-regular if every language L' of full words such that $L \uparrow L'$ is regular.*
The class of all the $(\exists)$-regular languages is denoted $\mathbf{REG}_{(\exists)}$, while that of $(\forall)$-regular languages by $\mathbf{REG}_{(\forall)}$.

According to the definitions from [4], the $(\exists)$-regular languages are those whose restoration contains at least a regular language, while $(\forall)$-regular languages are those whose restoration contains only regular languages.

We start with the following result.

Theorem 7. *For every non-empty alphabet V with $\diamond \notin V$ there exist an undecidable language L of partial words over V, such that:*
i) $\sigma(L) \in \mathbf{REG}$ for all substitutions σ over V, and $\sigma'(L) \notin \mathbf{REG}$ for the $\diamond$-substitution σ' with $\sigma'(\diamond) = V \cup \{c\}$, where $c \notin V$.
ii) every language $L' \subseteq V^*$ of full words, which is compatible with L, is regular and there is an undecidable language $L'' \subseteq (V')^*$, where V' strictly extends V, which is compatible with L.

Proof. Let $L_1 \subseteq V^*$ be an undecidable language (for instance, L_1 can be constructed by the classical diagonalization technique $L_1 = \{a^n \mid \text{the } n^{th} \text{ Turing machine in an enumeration of the Turing machines with binary input does not accept the binary representation of } n\}$, where $a \in V$) and $L = V^* \cup \{\diamond w \mid w \in L_1\}$. Clearly, for any $\diamond$-substitution σ over V, we have $\sigma(L) = V^*$. However, if we take a letter $c \notin V$ and the $\diamond$-substitution σ' which replaces $\diamond$ by $V \cup \{c\}$ we obtain an undecidable language $\sigma'(L)$. This concludes the proof of (i). To show (ii) we just have to note that the only language contained in V^* compatible with L is V^*, and, if we take a letter $c \notin V$ and replace $\diamond$ by c (or, in other words, if we see $\diamond$ as the conventional symbol c), we obtain an undecidable language compatible with L. □

We can now show a first result regarding the classes previously defined.

Theorem 8. $\mathbf{REG} = \mathbf{REG}_{(\forall\sigma)} \subset \mathbf{REG}_{\mathbf{max}}$.

Proof. It is rather clear that $\mathbf{REG}_{(\forall\sigma)} \subseteq \mathbf{REG}_{\mathbf{max}}$.

Since $\mathbf{REG}$ is closed to substitutions it follows that $\mathbf{REG} \subseteq \mathbf{REG}_{(\forall\sigma)}$.

It is also not hard to see that $\mathbf{REG}_{(\forall\sigma)} \subseteq \mathbf{REG}$ (given a language L in $\mathbf{REG}_{(\forall\sigma)}$, one can take the special substitution that replaces $\diamond$ with a symbol that does not occur in $\mathbf{alph}(L)$ and obtain a regular language; therefore L is a regular language if $\diamond$ is seen as a normal symbol).

By Theorem 7, $\mathbf{REG}_{\mathbf{max}}$ contains an undecidable language; indeed, given an non-empty alphabet V, the language L defined in its proof for V is in $\mathbf{REG}_{\mathbf{max}}$ according to (i). The strictness of the inclusion $\mathbf{REG} \subsetneq \mathbf{REG}_{\mathbf{max}}$ follows. □

The next result gives some insight on the structure of the class $\mathbf{REG}_{\mathbf{max}}$.

Theorem 9. *Let $L \in \mathbf{REG}_{\mathbf{max}}$ be a language of partial words over $V \neq \emptyset$ and σ the $\diamond$-substitution used in the definition of $\mathbf{REG}_{\mathbf{max}}$. Then there exists a maximal language (with respect to set inclusion) $L_0 \in \mathbf{REG}_{\mathbf{max}}$ of partial words over V such that $\sigma(L_0) = \sigma(L)$. Moreover, given an automaton accepting L, an automaton accepting L_0 can be constructed.*

It is also not hard to see that any language from $\mathbf{REG}_{\mathbf{max}}$ whose words contain only holes is regular.

The following relation also holds:

Theorem 10. $\mathbf{REG}_{\mathbf{max}} \subset \mathbf{REG}_{(\exists\sigma)} \subset \mathbf{REG}_{(\exists)}$.

Proof. The non-strict inclusions $\mathbf{REG}_{\mathbf{max}} \subseteq \mathbf{REG}_{(\exists\sigma)} \subseteq \mathbf{REG}_{(\exists)}$ are immediate. We show now that each of the previous inclusions is strict.

Take $L = \{(ab)^n\diamond b(ab)^n \mid n \in \mathbf{N}\}$. Considering σ a $\diamond$-substitution as in the definition of $\mathbf{REG}_{\mathbf{max}}$, we have $\sigma(L) \cap \{w \mid w \in \{a,b\}^*, w \text{ contains } bb\}$ is the language $\{(ab)^n bb(ab)^n\}$ which is not regular. Thus, $\sigma(L)$ is not regular, and L is not in $\mathbf{REG}_{\mathbf{max}}$. However, it is clearly in $\mathbf{REG}_{(\exists\sigma)}$, as when we take the substitution $\sigma(a) = \{a\}$, $\sigma(b) = \{b\}$, and $\sigma(\diamond) = \{a\}$, we have $\sigma(L) = \{(ab)^{2n+1} \mid n \in \mathbf{N}\}$, which is a regular language. This shows that $\mathbf{REG}_{\mathbf{max}} \subset \mathbf{REG}_{(\exists\sigma)}$.

Now, take $L = \{(ab)^n\diamond b(ab)^n \mid n \in \mathbf{N}\} \cup \{(ab)^n a\diamond(ab)^n \mid n \in \mathbf{N}\}$. This language is not in $\mathbf{REG}_{(\exists\sigma)}$ by arguments similar to above, but it is in $\mathbf{REG}_{(\exists)}$ as it is compatible with $\{(ab)^{2n+1} \mid n \in \mathbf{N}\}$. □

As already shown, all the languages in $\mathbf{REG}_{(\forall)}$ are in $\mathbf{REG} = \mathbf{REG}_{(\forall\sigma)}$; however, not all the languages in $\mathbf{REG}$ are in $\mathbf{REG}_{(\forall)}$. The following statement characterizes exactly the regular languages that are in $\mathbf{REG}_{(\forall)}$.

Theorem 11. *Let L be a regular partial-words-language over V. Then $L \in \mathbf{REG}_{(\forall)}$ if and only if the set $\{w \mid |w|_\diamond \geq 1, w \in L\}$ is finite.*

The previous result provides a simple procedure for deciding whether a regular partial-words-language is in $\mathbf{REG}_{(\forall)}$ or not. We simply check (taking as input a DFA for that language) whether there are finitely many words that contain $\diamond$ or not. If yes, we accept the input and confirm that the given language is in $\mathbf{REG}_{(\forall)}$; otherwise, we reject the input.

Theorem 11 has also the following consequence.

Theorem 12. $\mathbf{REG}_{(\forall)} \subset \mathbf{REG}$.

In many previous works (surveyed in [2]), partial words were defined by replacing specific symbols of full words by $\diamond$, in a procedure that resembles the puncturing of [4]. Similarly, in [5], partial words were defined by applying the finite transduction defined by a deterministic generalised sequential machine (DGSM) to full words, such that $\diamond$ appears in the output word. Accordingly, we can define a new class of partial-words-languages, $\mathbf{REG_{gsm}}$, using this automata-theoretic approach. Let L be a language of partial words over V, with $\diamond \in \mathbf{alph}(L)$; L is **gsm**-regular, and is in $\mathbf{REG_{gsm}}$, if there exists a DGSM M and a regular language L' such that L is obtained by applying the finite transduction defined by M to L'. It is not hard to show that $\mathbf{REG_{gsm}} = \mathbf{REG} \setminus \mathbf{REG_{full}}$.

By the Theorems 8,10,11, and 12 we get the following hierarchies:

$$\mathbf{REG_{full}} \subset \mathbf{REG}_{(\forall)} \subset \mathbf{REG} = \mathbf{REG}_{(\forall\sigma)} \subset \mathbf{REG_{max}} \subset \mathbf{REG}_{(\exists\sigma)} \subset \mathbf{REG}_{(\exists)}$$

$$\mathbf{REG} \setminus \mathbf{REG_{full}} = \mathbf{REG_{gsm}} \subset \mathbf{REG} = \mathbf{REG}_{(\forall\sigma)}$$

Finally, the closure properties of the defined classes are summarized in the following table. Note that y (respectively, n) at the intersection of the row associated with the class $\mathcal{C}$ and the column associated with the operation $\circ$ means that $\mathcal{C}$ is closed (respectively, not closed) under operation $\circ$. A special case is the closure of $\mathbf{REG_{max}}$ under union and concatenation: in general this class is not closed under these operations, but when we apply them to languages of $\mathbf{REG_{max}}$ over the same alphabet we get a language from the same class.

Class	$\cup$	$\cap$	$\cap\mathbf{REG}$	$\mathbf{alph}(L)^* \setminus L$	$*$	$\cdot$	ϕ	ϕ^{-1}	σ
$\mathbf{REG}_{(\forall)}$	y	y	y	n	n	n	n	n	n
$\mathbf{REG} = \mathbf{REG}_{(\forall\sigma)}$	y	y	y	y	y	y	y	y	y
$\mathbf{REG_{max}}$	n/y	n	n	n	y	n/y	n	n	n
$\mathbf{REG}_{(\exists\sigma)}$	n	n	n	n	y	n	n	n	n
$\mathbf{REG}_{(\exists)}$	y	n	n	y	y	y	n	n	n

References

1. J. Berstel and L. Boasson. Partial words and a theorem of Fine and Wilf. *Theoretical Computer Science*, 218:135–141, 1999.
2. F. Blanchet-Sadri. *Algorithmic Combinatorics on Partial Words.* Chapman & Hall/CRC Press, 2008.
3. M. J. Fischer and M. S. Paterson. String matching and other products. In *Complexity of Computation, SIAM-AMS Proceedings*, volume 7, pages 113–125, 1974.
4. G. Lischke. Restoration of punctured languages and similarity of languages. *Mathematical Logic Quarterly*, 52(1):20–28, 2006.
5. F. Manea and R. Mercaş. Freeness of partial words. *Theoretical Computer Science*, 389(1-2):265–277, 2007.
6. F. Manea and C. Tiseanu. Hard counting problems for partial words. In *LATA*, volume 6031 of *Lect. Notes Comput. Sci.*, pages 426–438. Springer-Verlag, 2010.
7. G. Rozenberg and A. Salomaa. *Handbook of Formal Languages.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1997.