Metric Lexical Analysis[⋆]

Cristian S. Calude[1], Kai Salomaa[2], and Sheng Yu[3]

[1] Computer Science Department, The University of Auckland, Private Bag 92109
Auckland, New Zealand
cristian@cs.auckland.ac.nz

[2] Department of Computing and Information Science, Queen's University
Kingston, Ontario, Canada K7L 3N6
ksalomaa@cs.queensu.ca

[3] Department of Computer Science, The University of Western Ontario
London, Ontario, Canada N6A 5B7
syu@csd.uwo.ca

Abstract. We study automata-theoretic properties of distances and quasi-distances between words. We show that every additive distance is finite. We also show that every additive quasi-distance is regularity-preserving, that is, the neighborhood of any radius of a regular language with respect to an additive quasi-distance is regular. As an application we present a simple algorithm that constructs a metric (fault-tolerant) lexical analyzer for any given lexical analyzer and desired radius (fault-tolerance index).

1 Introduction

You are frustrated when you type a UNIX command incorrectly and cannot find what the correct spelling is. You may be wondering why the system does not give any suggestions on what command you might want to type. Those questions concern the concepts of distances between words and neighborhoods of languages with respect to a distance and a radius.

Much work has been done in spell checking and correction, and other online dictionary applications using various methods [5,7,8,9]. Here, we study some automata-theoretic properties of different measurements of distances between words.

Let Σ be a finite alphabet. By the *neighborhood* of a word $w \in \Sigma^*$ of radius α with respect to a distance measure δ, we mean the set of all words u that have the distance measure $\delta(u, w)$ at most α. We denote this neighborhood by $E(\{w\}, \delta, \alpha)$. Naturally, the neighborhood of a language L of a radius α with respect to δ, denoted $E(L, \delta, \alpha)$, is the union of $E(\{w\}, \delta, \alpha)$ for all words $w \in L$. A distance δ is said to be finite if $E(\{w\}, \delta, \alpha)$ is finite for all $w \in \Sigma^*$ and

[⋆] The work reported here has been supported by the Natural Sciences and Engineering Research Council of Canada Grants OGP0041630, OGP0147224 and was done during the visit of the first author to the Department of Computer Science, The University of Western Ontario, Canada.

$\alpha \geq 0$. Informally, δ is said to be additive if its measurement distributes over concatenation, and regularity-preserving if $E(R, \delta, \alpha)$ is regular for every regular language R and radius $\alpha \geq 0$.

In this paper, we prove that every additive distance is finite. We also show, as our main result, that every additive distance (or quasi-distance) is regularity-preserving. Examples of various additive and non-additive distance measures are also given in the paper.

As an application of the main result, we construct a very simple algorithm that transforms a given lexical analyzer to a metric (fault-tolerant) lexical analyzer for an arbitrary radius (fault-tolerance index).

The paper is organized as follows: In the next section we introduce the basic notation. In Section 3, we define distances and quasi-distances. Our main results concerning finite, additive, and regularity-preserving distance measures are presented in Section 4. In the last section we define metric lexical analyzers and describe a simple algorithm that constructs a metric lexical analyzer for a given lexical analyzer and desired radius.

2 Preliminaries

We assume that the reader is familiar with the basics of formal languages and finite automata in particular, cf. [4,10,11]. Here we introduce the notation we will use in the later sections.

The symbol Σ denotes a finite alphabet and Σ^* the set of finite words over Σ. The empty word is denoted by λ and the length of a word $w \in \Sigma^*$ by $|w|$. The shuffle of words $u, v \in \Sigma^*$,

$$\omega(u, v) \subseteq \Sigma^*$$

is the set of all words $x_1 y_1 x_2 \ldots x_m y_m$ such that $u = x_1 \cdots x_m$, $v = y_1 \cdots y_m$, $x_i, y_i \in \Sigma^*$, $i = 1, \ldots, m$, $m > 0$. The catenation of languages $S, T \subseteq \Sigma^*$ is denoted by ST.

A deterministic finite automaton (DFA) is a five-tuple $A = (Q, \Sigma, \gamma, s, F)$ where Q is the finite set of states, Σ is the finite alphabet, $s \in Q$ is the initial state, $F \subseteq Q$ is the set of final states, and $\gamma : Q \times \Sigma \to Q$ is the state-transition function. If A is defined as above except that γ is a function $Q \times \Sigma \to \mathcal{P}(Q)$ then we say that A is a nondeterministic finite automaton (NFA). (Here $\mathcal{P}(Q)$ is the set of subsets of Q.)

The state-transition relation γ of an NFA is extended in the natural way to a function $\hat{\gamma} : Q \times \Sigma^* \to \mathcal{P}(Q)$. We denote also $\hat{\gamma}$ simply by γ and the language accepted by A is $L(A) = \{w \in \Sigma^* \mid \gamma(s, w) \cap F \neq \emptyset\}$.

3 Distances and Quasi-distances

We want to measure the distance between distinct words of Σ^*. Let S be a set. We say that a function $\delta : S \times S \to [0, \infty)$ is a *distance* if it satisfies the following three conditions:

(D1) $\delta(x, y) = 0$ iff $x = y$, for all $x, y \in S$,
(D2) $\delta(x, y) = \delta(y, x)$, for all $x, y \in S$,
(D3) $\delta(x, z) \leq \delta(x, y) + \delta(y, z)$, for all $x, y, z \in S$.

Condition (D3) is called the triangle-inequality. A function $\delta : S \times S \to [0, \infty)$ that satisfies (D2) and (D3) and the weaker condition

(D1') $\delta(x, x) = 0$, for all $x \in S$,

is called a *quasi-distance* on S. A quasi-distance allows the possibility that $\delta(x, y) = 0$, for $x \neq y$.

Note that if δ is a quasi-distance on S we can define an equivalence relation $\sim_\delta$ on S by setting $x \sim_\delta y$ iff $\delta(x, y) = 0$. Then the mapping δ' defined by $\delta'([x]_{\sim_\delta}, [y]_{\sim_\delta}) = \delta(x, y)$ is a distance on $S/\sim_\delta$. (Since δ satisfies the condition (D3) it follows that the value of $\delta'([x]_{\sim_\delta}, [y]_{\sim_\delta})$ does not depend on the representatives x and y.)

Let δ be a (quasi-) distance on S, $K \subseteq S$ and $\alpha \geq 0$. The *neighborhood of K of radius α* (with respect to δ) is

$$E(K, \delta, \alpha) = \{x \in S \mid (\exists y \in K)\ \delta(x, y) \leq \alpha\}.$$

A natural distance between words of the same length is the so called *Hamming distance.* Since we need to compare also words of different lengths, there is more than one natural way to extend Hamming distance.

Let # be a symbol not appearing in Σ and put $\Gamma = \Sigma \cup \{\#\}$. For $a, b \in \Gamma$ define

$$\Delta(a, b) = \begin{cases} 1, & \text{if } a \neq b, \\ 0, & \text{if } a = b. \end{cases}$$

Define $\Delta_n : \Gamma^n \times \Gamma^n \to \mathbb{N}$ by setting

$$\Delta_n(x_1 \cdots x_n, y_1 \cdots y_n) = \sum_{i=1}^{n} \Delta(x_i, y_i).$$

The *prefix-Hamming distance* $\delta_{\rm pH}$ on Σ^* is defined as follows. Let $u, v \in \Sigma^*$. Then

$$\delta_{\rm pH}(u, v) = \begin{cases} \Delta_{|v|}(u\#^k, v), & \text{if } k = |v| - |u| \geq 0, \\ \Delta_{|u|}(u, v\#^k), & \text{if } k = |u| - |v| > 0. \end{cases}$$

The prefix-Hamming distance counts the number of distinct symbols in the first $\min\{|u|, |v|\}$ positions of the words u and v and adds to the result the length of the remaining suffix. It is easy to verify that $\delta_{\rm pH}$ satisfies the triangle-inequality and, thus, it is a distance. On the other hand, this distance is not very useful from a practical point of view because inserting or deleting one letter can change the distance of given words by an arbitrary amount (depending on the length of the words).

A better extension is the function which considers all possible ways to pad both words and then takes the minimum of the obtained distances. Let $u, v \in \Sigma^*$. Then we define

$$\begin{aligned}\delta_{\mathrm{H}}(u,v) = \min\{\Delta_k(x,y) \mid \\ k \geq \max\{|u|,|v|\}, x \in \omega(u, \#^{k-|u|}), y \in \omega(v, \#^{k-|v|})\}.\end{aligned} \tag{1}$$

Notice that for all $u, v \in \Sigma^*$, $\delta_H(u,v) \leq \max\{|u|,|v|\}$, and $\delta_{\mathrm{H}}(u,v) = \min\{\Delta_{|uv|}(x,y) \mid x \in \omega(u, \#^{|v|}),\ y \in \omega(v, \#^{|u|})\}$.

In general, $\delta_{\mathrm{H}}(u,v) \neq \Delta_{\max\{|u|,|v|\}}(x,y)$, for every $x \in \omega(u, \#^{\max\{|u|,|v|\}-|u|})$, $y \in \omega(v, \#^{\max\{|u|,|v|\}-|v|})$. For example, take $u = abab, v = baba$ and observe that $\omega(u, \#^0) = \{u\}, \omega(v, \#^0) = \{v\}$, $\Delta_4(u,v) = 4 > \delta_{\mathrm{H}}(u,v) = \Delta_5(u\#, \#v) = 2$.

It is convenient to look at (1) as a process. Consider changing a word into another word by means of the following three types of *edit steps* ([6]): a) *insert*—insert a character into a word, b) *delete*—delete a character from a word, c) *replace*—replace one character with a different character. Edit steps can be applied in any order. For example, to change the word *abab* into *baba* we can use rule c) (replace) four times and we get *bbab*, *baab*, *babb*, *baba*. We can be more efficient by deleting the first character of *abab* to get *bab*, then insert *a* at the end, so with only two edit steps we obtain *baba*. As we have seen below, $\delta_{\mathrm{H}}(abab, baba) = 2$; it can be obtained by first constructing the extended words $abab\#$ and $\#baba$ and then computing their Δ_5 distance. In fact, we have:

Lemma 1. *For all words u, v, $\delta_{\mathrm{H}}(u,v)$ coincides with the minimal number of edit steps necessary to change u into v.*[1]

Corollary 1. *The function δ_{H} satisfies (D1)–(D3).*

The function δ_{H} is a distance by Corollary 1; as it extends Hamming's distance it is appropriate to call it the *shuffle-Hamming distance.*

An immediate property of the shuffle-Hamming distance follows: insertions and deletions of the special symbol # do not count.

Lemma 2. *For all $u, v \in \Sigma^*$, and $i \geq 0$, $\delta_{\mathrm{H}}(u,v) = \delta_{\mathrm{H}}(\bar{u}, \bar{v})$, for all $\bar{u} \in \omega(u, \#^i), \bar{v} \in \omega(v, \#^i)$.*

Other possible distances can be obtained by varying edit steps (e.g., allowing adjacent characters in one word to be interchanged while copied to the other word) or by assigning cost functions to edit steps (e.g., capturing the idea that the cost of replacing a character is less than the combined costs of deletion and insertion). See [2] for more examples of discrete distances.

4 Neighborhoods of Regular Languages

Let L be a regular language over Σ. We are interested in the following question: Which conditions the distance δ should satisfy in order to guarantee that all the

[1] This number is called the *edit-distance* in [3], pp 325–326; it has been suggested by Ulam [12].

languages $E(L, \delta, \alpha)$, $\alpha \geq 0$, are regular? We say that a distance δ is *regularity-preserving* if $E(L, \delta, \alpha)$ is a regular language for all regular languages L and $\alpha \geq 0$.

It is fairly straightforward to construct examples of distances on Σ^* that are not regularity-preserving. Here is such an example.

Example 1. Let $\Sigma = \{a, b\}$. Construct the distance δ by

$$\delta(u,v) = \begin{cases} 0, & \text{if } u = v, \\ 1/2, & \text{if } u = a^n b^n, v = a^m b^m, \text{ for some } n, m \geq 0, n \neq m, \\ 1, & \text{otherwise,} \end{cases}$$

and notice that $E(\{ab\}, \delta, 1/2) = \{a^n b^n \mid n \geq 0\}$. □

Clearly we need to impose some additional conditions on the distance δ. Note that the distance in Example 1 has the property that for $n \geq 0$ and $\alpha \geq 1/2$, the inequality $\delta(u, a^n b^n) \leq \alpha$ has infinitely many solutions. Hence, the following finiteness requirement seems to be a suitable candidate to guarantee that a distance is regularity-preserving.

We say that a (quasi-) distance δ on Σ^* is *finite* if for all $w \in \Sigma^*$ and $\alpha \geq 0$, the set $E(\{w\}, \delta, \alpha)$ is finite.

Both the shuffle-Hamming distance and the prefix-Hamming distance considered above are clearly finite. The following example shows that finiteness of a distance δ is, unfortunately, not sufficient to guarantee that δ is regularity-preserving.

Example 2. Let $\Sigma = \{a, b, c\}$. By slightly modifying the prefix-Hamming distance δ_{pH} we construct a finite distance δ on Σ^* that is not regularity-preserving.

For $u, v \in \Sigma^*$ we define

$$\delta(u,v) = \begin{cases} 3/2, & \text{if } u = a^n b a^n, v = a^n c a^n, n \geq 0, \text{ or vice versa,} \\ \delta_{\mathrm{pH}}(u,v), & \text{otherwise.} \end{cases}$$

Clearly δ satisfies the conditions (D1) and (D2), so in order to show that it is a distance it is sufficient to verify the triangle-inequality. Assuming that (D3) does not hold, we must have $x, y, z \in \Sigma^*$ such that

$$\delta(x,z) > \delta(x,y) + \delta(y,z). \tag{2}$$

Since for all $u, v \in \Sigma^*$, $\delta(u,v) \geq \delta_{\mathrm{pH}}(u,v)$ and δ_{pH} is a distance, it follows that if (2) holds, then necessarily $\delta(x,z) \neq \delta_{\mathrm{pH}}(x,z)$, that is, $x = a^n b a^n$, $z = a^n c a^n$, $n \geq 0$, or vice versa. Thus $\delta(x,z) = 3/2$, and (2) implies that $\delta(x,y) = 0$ or $\delta(y,z) = 0$. Both possibilities directly yield a contradiction.

Also, δ is finite since for any $\alpha \geq 2$ and $w \in \Sigma^*$ we have $E(\{w\}, \delta, \alpha) = E(\{w\}, \delta_{\mathrm{pH}}, \alpha)$.

To see that δ is not regularity-preserving choose $L = a^* b a^*$. Then

$$E(L, \delta, 3/2) - E(L, \delta, 1) = \{a^n c a^n \mid n \geq 0\},$$

which implies that at least one of the languages $E(L, \delta, 3/2)$ and $E(L, \delta, 1)$ is not regular. □

The above example shows that we need to look for stronger restrictions for regularity-preserving distances. Since elements of Σ^* have a unique decomposition into subwords (of given length) it is perhaps reasonable to assume that the distances should "respect" such decompositions. Thus we say that a (quasi-) distance δ on Σ^* is *additive* if always when $w = w_1 w_2$ ($w_1, w_2 \in \Sigma^*$) we have for all $\alpha \geq 0$,

$$E(\{w\}, \delta, \alpha) = \bigcup_{\beta_1+\beta_2=\alpha} E(\{w_1\}, \delta, \beta_1) E(\{w_2\}, \delta, \beta_2). \tag{3}$$

First we observe that an additive distance is always finite. Note that an additive quasi-distance δ need not be finite. If, for some $b \in \Sigma$, $\delta(b, \lambda) = 0$, then any δ-neighborhood is necessarily infinite.

Lemma 3. *Every additive distance is finite.*

Proof. Let δ be an additive distance on Σ^*. By (3), for any $w = b_1 \cdots b_k$, $b_i \in \Sigma$, $i = 1, \ldots, k$, $E(\{w\}, \delta, \alpha)$ is contained in the catenation of the languages $E(\{b_1\}, \delta, \alpha)$, …, $E(\{b_k\}, \delta, \alpha)$. Thus, it is sufficient to show that $E(\{b\}, \delta, \alpha)$ is finite for $b \in \Sigma$ and $\alpha \geq 0$.

Let $u = c_1 \cdots c_m$, $c_i \in \Sigma$, be an arbitrary word of Σ^*. The additivity condition implies that $u \in E(\{b\}, \delta, \alpha)$ iff there exists $i \in \{1, \ldots, m\}$ such that

$$\delta(b, c_i) + \sum_{j \in \{1,\ldots,m\},\ j \neq i} \delta(\lambda, c_j) \leq \alpha. \tag{4}$$

There exist only a finite number of words $u = c_1 \cdots c_m$ that satisfy the above inequality. □

Both the prefix-Hamming distance and the shuffle-Hamming distance are additive.

Proposition 1. *The distances δ_{pH} and δ_{H} defined on an alphabet Σ are additive.*

Proof. We show that δ_{H} is additive as the proof for the distance δ_{pH} is simpler.

Let $w = w_1 w_2$ be an arbitrary decomposition of a word $w \in \Sigma^*$. We show that for every $u \in \Sigma^*$,

$$u \in E(\{w_1 w_2\}, \delta_{\mathrm{H}}, \alpha) \text{ iff } u \in \bigcup_{\beta_1+\beta_2=\alpha} E(\{w_1\}, \delta_{\mathrm{H}}, \beta_1) E(\{w_2\}, \delta_{\mathrm{H}}, \beta_2).$$

Assume $\delta_{\mathrm{H}}(u, w_1 w_2) \leq \alpha$. As edit steps (in the process of changing a word into another word) can be applied in any order, we can start the process of changing u into $w_1 w_2$ in such a way to obtain first w_1 from a prefix u_1 of u, and then w_2 (from the remaining suffix u_2 of u). Consequently, $\delta_{\mathrm{H}}(u_1, w_1) + \delta_{\mathrm{H}}(u_2, w_2) = \delta_{\mathrm{H}}(u, w_1 w_2) \leq \alpha$. Conversely, if $u_i \in E(\{w_i\}, \delta_{\mathrm{H}}, \beta_i)$, $i = 1, 2$, $\beta_1 + \beta_2 \leq \alpha$, then we have $\delta_{\mathrm{H}}(u_1 u_2, w_1 w_2) \leq \delta_{\mathrm{H}}(u_1, w_1) + \delta_{\mathrm{H}}(u_2, w_2) \leq \alpha$. □

From Example 2 we know that a finite distance need not preserve regularity. Below we show that, on the other hand, additivity is a sufficient condition to guarantee that even a quasi-distance preserves regularity. Note that, as observed above, an additive quasi-distance need not be finite. First we prove the following lemma.

Lemma 4. *Assume that δ is an additive quasi-distance on Σ^*.*

(i) *For each $b \in \Sigma$ and $\alpha \geq 0$, $E(b, \delta, \alpha)$ is regular.*
(ii) *Let $b \in \Sigma$ and $\alpha \geq 0$ be fixed. There exists an integer k and numbers $0 = \alpha_1 < \ldots < \alpha_k = \alpha$ such that*

$$E(b, \delta, \alpha_i), \quad i = 1, \ldots, k,$$

are all the distinct neighborhoods of b having radius at most α.

Proof. (i) Let $u = c_1 \cdots c_m$, $m \geq 0$, $c_i \in \Sigma$, $i = 1, \ldots, m$. As in the proof of Lemma 3 it follows that $u \in E(b, \delta, \alpha)$ iff the inequality (4) holds. (Note that, in contrast to Lemma 3, δ is now only a quasi-distance, so this does not imply the finiteness of the neighborhood.)

Denote

$$\Theta = \{d \in \Sigma \mid \delta(d, \lambda) = 0\}.$$

Let Ψ be the set of finite multisets of elements of Σ,

$$\{c_i, c_{j_1}, \ldots, c_{j_r}\}$$

such that $\delta(\lambda, c_{j_l}) \neq 0$, $l = 1, \ldots, r$ and

$$\delta(b, c_i) + \sum_{l=1}^{r} \delta(\lambda, c_{j_l}) \leq \alpha.$$

Then $u = c_1 \cdots c_m$ satisfies the inequality (4) iff u is the shuffle of a sequence obtained by listing the elements of a multiset belonging to Ψ (in arbitrary order) and a word in Θ^*. The shuffle of a finite language and a regular language is always regular.

(ii) In the construction above the elements of the multisets belonging to Ψ completely determine the neighborhoods of radius at most α around b. Thus as the radii α_s, $s = 1, \ldots, k$, we can simply take all the (distinct) sums $\delta(b, c_i) + \sum_{l=1}^{r} \delta(\lambda, c_{j_l})$ where the multiset $\{c_i, c_{j_1}, \ldots, c_{j_r}\}$ belongs to Ψ. (Note that Ψ is a finite collection of multisets.) □

The above construction implies that Lemma 4 (ii) can be written in the following stronger form:

Corollary 2. *Assume that δ is an additive quasi-distance on Σ^* and let $b \in \Sigma$ and $\alpha \geq 0$ be fixed. Then we can write*

$$E(b, \delta, \alpha) = R_1 \cup \ldots \cup R_k$$

where $R_i = \{w \in \Sigma^* \mid \delta(b, w) = \alpha_i\}$, $i = 1, \ldots, k$, is regular.

Proof. Without loss of generality we can assume that the numbers α_i in Lemma 4 (ii) are chosen so that there exists $w_i \in \Sigma^*$ with $\delta(b, w_i) = \alpha_i$, $i = 1, \ldots, k$. Let R_i, $i = 1, \ldots, k$, be as above. By Lemma 4 (ii), $R_i = E(b, \delta, \alpha_i) - E(b, \delta, \alpha_{i-1})$, $i = 2, \ldots, k$, and $R_1 = E(b, \delta, 0)$. By Lemma 4 (i), these sets are regular. □

Now we are ready to prove the main result of this section.

Theorem 1. *Assume that δ is an additive quasi-distance on Σ^* and let $L \subseteq \Sigma^*$ be regular. Then $E(L, \delta, \alpha)$ is regular for all $\alpha \geq 0$.*

Proof. Let $\alpha \geq 0$ be fixed and let $A = (Q, \Sigma, \gamma, s, F)$ be a DFA such that $L = L(A)$. Without loss of generality we can assume that the initial state s is not reachable from any other state.

By Corollary 2, for each $b \in \Sigma$ we can write

$$E(b, \delta, \alpha) = R_1^b \cup \ldots \cup R_{k(b)}^b,$$

where

$$R_j^b = \{w \in \Sigma^* \mid \delta(w, b) = \alpha_j^b\}, \quad 0 \leq \alpha_j^b \leq \alpha,$$

is regular, $j = 1, \ldots, k(b)$. Denote $D' = \{\alpha_j^b \mid b \in \Sigma, 1 \leq j \leq k(b)\}$ and

$$D = \{\beta \leq \alpha \mid \beta = \beta_1 + \ldots + \beta_r, \beta_i \in D', 1 \leq i \leq r\}.$$

We construct an NFA $B = (Q_B, \Sigma, \gamma_B, s_B, F_B)$ such that

$$L(B) = E(L(A), \delta, \alpha).$$

Choose $Q_B = Q \times D$, $s_B = (s, 0)$ and

$$F_B = \begin{cases} F \times D \cup \{s_B\} & \text{if } \lambda \in E(L(A), \delta, \alpha) \\ F \times D & \text{otherwise.} \end{cases}$$

The transition relation γ_B is defined as follows. Let $q \in Q$, $\beta \in D$ and $b \in \Sigma$. Then

$$(q', \beta + \alpha_j^b) \in \gamma_B((q, \beta), b) \tag{5}$$

for every $q' \in \gamma(q, R_j^b)$, $1 \leq j \leq k(b)$, such that $\beta + \alpha_j^b \leq \alpha$. (Here $\gamma(q, R_j^b) = \{\gamma(q, v) \mid v \in R_j^b\}$.) Since R_j^b is regular, the set $\gamma(q, R_j^b)$ ($\subseteq Q$) can even be effectively determined.

Let $w = b_1 \cdots b_m$, $m \geq 1$, $b_i \in \Sigma$, $i = 1, \ldots, m$. Since δ is additive

$$\begin{aligned} & w \in E(L(A), \delta, \alpha) \text{ iff } (\exists u \in L(A)) \text{ such that} \\ & u \in \bigcup_{\beta_1 + \ldots + \beta_m = \alpha} E(b_1, \delta, \beta_1) \cdots E(b_m, \delta, \beta_m). \end{aligned} \tag{6}$$

In the transitions (5), on input b the first component of the states of B simulates the computation of A on an arbitrary (nondeterministically chosen) word of $v \in R_j^b$, and in the second component we correspondingly increment the distance by $\alpha_j^b = \delta(b, v)$. By observation (6), some sequence of the nondeterministic choices on input $w = b_1 \cdots b_m$ leads to an accepting state of F_B iff w is in $E(L(A), \delta, \alpha)$. By the choice of the set F_B, the NFA B accepts λ if and only if $\lambda \in E(L(A), \delta, \alpha)$. □

5 A Metric Lexical Analyzer

The major difference between a lexical analyzer and a (traditionally-defined) finite automaton is that, in a lexical analyzer, each final state is linked to an action (or a set of actions). Because of this difference, the algorithms that are designed for finite automata may not directly apply to lexical analyzers. The equivalence of two final states in a deterministic lexical analyzer requires that not only the states are equivalent in the sense of a DFA, but also that they have the same action (or actions).

There are many other features which are associated with certain types of lexical analyzers. For example, some lexical analyzers assume that each input word has an end-of-word symbol. Also, many practical lexical analyzers are implemented using a data structure called *trie* [1]. However, those features are not considered to be common or essential to general lexical analyzers.

A lexical analyzer can be considered as a special type of Moore machine [4] with all nonfinal states having the empty output (action).

For notational convenience, we formally define a lexical analyzer to be a 7-tuple

$$A = (Q, \Sigma, \Gamma, \gamma, s, F, \tau)$$

where $(Q, \Sigma, \gamma, s, F)$ is a finite automaton; Γ is a set of actions; and, $\tau : F \to \Gamma$ is an action-function. Whether A is deterministic or nondeterministic depends on whether its underlying finite automaton is a DFA or an NFA.

Let $A = (Q, \Sigma, \Gamma, \gamma, s, F, \tau)$ be a deterministic (nondeterministic) lexical analyzer. Denote by $L(A)$ the set of all words recognized by the underlying finite automaton $(Q, \Sigma, \gamma, s, F)$. For $w \in L(A)$, denote by $\hat{\tau}(w)$ the action $\tau(\hat{\gamma}(s, w))$ (the set of actions $\{\tau(f) \mid f \in F \cap \hat{\gamma}(s, w)\}$). We simply write $\tau(w)$ instead of $\hat{\tau}(w)$ if there is no confusion. The above definition implies that if $w \in L(A)$ and the (an) accepting path for w goes through several final states, only the action associated to the last final state is activated.

A simple lexical analyzer A_u is shown in Figure 1.

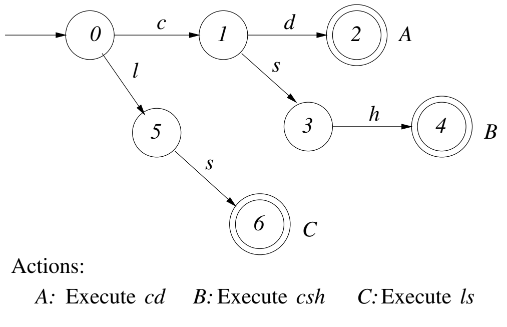

Fig. 1. A lexical analyzer A_u

Let $A = (Q, \Sigma, \Gamma, \gamma, s, F, \tau)$ be a lexical analyzer, δ a regularity-preserving distance and $\alpha \geq 0$ a radius. A lexical analyzer $A' = (Q', \Sigma, \Gamma', \gamma', s', F', \tau')$ is called a *metric lexical analyzer* of A with respect to δ and α if

(M1) $L(A') = E(L(A), \delta, \alpha)$, and
(M2) for each $w \in L(A)$, $\tau'(w) = \tau(w)$.

The proof of Theorem 1 gives a general guideline for constructing a metric lexical analyzer from a given lexical analyzer, an additive quasi-distance, and a radius. In the following, however, we consider only the shuffle-Hamming distance δ_{H}. The idea below can be easily generalized to all additive quasi-distances.

Construction: A Deterministic Metric Lexical Analyzer

Given: A deterministic lexical analyzer (DLA) $A = (Q, \Sigma, \Gamma, \gamma, s, F, \tau)$ and an integer $k > 0$.
Result: A deterministic metric lexical analyzer (DMLA)

$$A' = (Q', \Sigma, \Gamma', \gamma', s', F', \tau')$$

of A with radius k.

Construction steps:

i) Construct a nondeterministic lexical analyzer (NLA)

$$A'' = (Q'', \Sigma, \Gamma'', \gamma'', s'', F'', \tau'')$$

such that
$Q'' = Q \times \{0, \ldots, k\}$,
$\Gamma'' = \Gamma \cup \{\tilde{e} \mid e \in \Gamma\}$,
$s'' = (s, 0)$,
$F'' = \{(f, i) \mid f \in F \ \& \ i = 0, \ldots, k\}$,

$$\gamma'' : \begin{cases} (q,i) \in \gamma''((p,i),a), & \text{for } i = 0, \ldots, k, \text{ if } q = \gamma(p,a), \\ (q,i+1) \in \gamma''((p,i),b), & \text{for } i < k \text{ and } b \in \Sigma, b \neq a \text{ if } q = \gamma(p,a), \\ (q,i+1) \in \gamma''((q,i),a), & \text{for all } a \in \Sigma \text{ and } (q,i) \in Q'' \text{ where } i < k, \\ (q,i+1) \in \gamma''((p,i),\lambda), & \text{for } i < k \text{ if } q = \gamma(p,a), \text{ for some } a \in \Sigma, \end{cases}$$

$$\tau'' : \begin{cases} \tau''((f,0)) = \tau(f), & \text{for } f \in F, \\ \tau''((f,i)) = \tilde{e}, & \text{for } f \in F \text{ and } i = 1, \ldots, k, \text{ if } \tau(f) = e. \end{cases}$$

ii) Reduce A'' by deleting those states that are not reachable from s'' or that cannot reach a final state.

iii) Construct A' using the subset construction method [4] such that Q', γ', s', and F' are defined as in a standard subset construction, except that if a new state $r \in Q'$ contains both (q,i) and (q,j) for some $q \in Q$ and $i < j$ then delete (q,j) from r; $\Gamma' \subseteq \mathcal{P}(\Gamma'')$, and $\tau'(f') = \tau(f)$ if $(f,0) \in f'$, for some $(f,0) \in F''$, or $\tau'(f') = \{\tau''(f'') \mid f'' \in f' \ \& \ f'' \in F''\}$, otherwise.

iv) Simplify A' by merging all the equivalent states (that have the same actions if they are final states).

Note that the above construction uses two copies of the original set of actions ($\Gamma \cup \{\tilde{e} \mid e \in \Gamma\}$) in order to guarantee that the property (M2) holds.

A nondeterministic metric lexical analyzer A''_u of A_u with radius 1 is constructed following Step i) and Step ii) and shown in Figure 2, where $\Gamma'' = \{A, B, C, \tilde{A}, \tilde{B}, \tilde{C}\}$ and $\tau''((2,0)) = A$, $\tau''((2,1)) = \tilde{A}$, $\tau''((4,0)) = B$, $\tau''((4,1)) = \tilde{B}$, $\tau''((6,0)) = C$, $\tau''((6,1)) = \tilde{C}$. We use $[\hat{\ } a_1 \cdots a_t]$ to denote all letters in $\Sigma - \{a_1, \ldots, a_t\}$, and $\cdot$ to denote all letters in Σ.

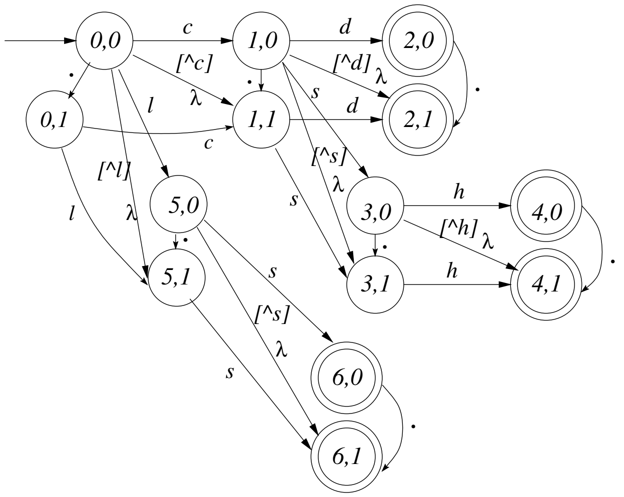

Fig. 2. A nondeterministic metric lexical analyzer A''_u

The resulting DMLA A' for A is shown in the following table, where $t1, \ldots, t6$ are terminating states which have no transitions:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
c	1	5	t1	X	10	X	X	X	t2	X	X	X	X	16	t5	X	X	18	X	X	18	18
d	17	2	t1	X	9	t1	t1	t1	t2	X	X	X	X	16	t5	X	t1	t1	t1	X	t1	t1
h	21	6	t3	t2	8	t2	t2	t2	t2	t2	t2	t2	t2	t4	t4	X	X	X	X	X	t2	X
l	13	7	t1	X	10	X	X	X	t2	X	X	X	X	15	t5	X	X	19	X	X	19	19
s	20	4	3	X	10	12	12	11	t2	X	X	X	X	14	t5	t5	12	11	12	t5	11	11
ˆ	21	5	t1	X	10	X	X	X	t2	X	X	X	X	t5	t5	X	X	X	X	X	X	X
τ'			A	$\tilde{A}$	$\tilde{A}$ $\tilde{B}$ $\tilde{C}$	$\tilde{A}$	$\tilde{A}$ $\tilde{B}$	$\tilde{A}$	B	$\tilde{A}$ $\tilde{B}$	$\tilde{B}$	$\tilde{C}$		$\tilde{A}$ $\tilde{C}$	C	$\tilde{C}$	$\tilde{C}$	$\tilde{A}$			$\tilde{C}$	

$$\tau'(t1) = \{\tilde{A}\},\ \tau'(t2) = \{\tilde{B}\},\ \tau'(t3) = \{\tilde{A}, \tilde{B}\},\ \tau'(t4) = \{\tilde{B}, \tilde{C}\},\ \tau'(t5) = \{\tilde{C}\},$$
$$\tau'(t6) = \{\tilde{A}, \tilde{C}\}.$$

Acknowledgment. We wish to thank Dr. Kaizhong Zhang for helpful discussions and suggestions.

References

1. A.V. Aho, R. Sethi, J.D. Ullman: *Compilers Principles, Techniques, and Tools*, Addison-Wesley, Reading, MA, 1988.
2. C.S. Calude, E. Calude: On some discrete metrics, *Bull. Math. Soc. Sci. Math. R. S. Roumanie (N. S.)* 27 (75) (1983), 213–216.
3. T.H. Cormen, C.E. Leiserson, R.L. Rivest: *Introduction to Algorithms,* MIT Press, Cambridge, MA, 1990.
4. J.E. Hopcroft, J.D. Ullman: *Introduction to Automata Theory, Languages, and Computation,* Addison-Wesley, Reading, MA, 1979.
5. R.L. Kashyap, B.J. Oommen: Spelling correction using probabilistic methods, *Pattern Recognition Letters* 2, 3 (1984), 147–154.
6. U. Manber: *Introduction to Algorithms—A Creative Approach*, Addison-Wesley, Reading, MA, 1989.
7. D.J. Margoliash: *CSpell—A Bloom Filter-Based Spelling Correction Program*, Masters Thesis, Dept. of Computer Science, Univ. of Western Ontario, London, Canada, 1987.
8. A. Mateescu, A. Salomaa, K. Salomaa, S. Yu: Lexical analysis with a simple finite-fuzzy-automaton model, *Journal of Universal Computer Science*, 1, 5 (1995), 292–311.
9. M. Mor, A.S. Fraenkel: A hash code method for detecting and correcting spelling errors, *Comm. ACM* 25, 12 (1982), 935–938.
10. A. Salomaa: *Formal Languages,* Academic Press, New York, 1973.
11. S. Yu: Regular languages. In: *Handbook of Formal Languages, Vol. I.* (G. Rozenberg, A. Salomaa, eds.) pp. 41–110, Springer-Verlag, Berlin, 1997.
12. S. Ulam: Some ideas and prospects in biomathematics, *The Annual Review of Biophysics and Bioengineering*, 1 (1972), 277–292. Reprinted in S. Ulam: *Science, Computers, and People* (M. C. Reynolds, G.- C. Rota, eds.), Birkhäuser, Boston, 1986, 115-136.