CSME 206A Natural Language & Speech Processing **Spring Semester**

Lecture 4: Finite State Transducers for Morphological Analysis

Lecturer: K.R. Chowdhary *: Professor of CS*

4.1 Morphological Parsing with Finite-state Transducers

The objective of the morphological parsing is to produce output lexicons for a single input lexicon, e.g., like it is given in table 4.1. The second column in the table contains the stem of the corresponding word (lexicon) in first column, along with its morphological features, like, +N means word is noun, +SG means it is singular, +PL means it is plural, +V for verb, and pres-part for present participle. We achieve it through two level morphology, which represents a word as a correspondence between lexical level - a simple concatenation of lexicons, as shown in column 2 of table 4.1, and a surface level as shown in column 1. These are shown using two tapes of finite state transducer.

Table 4.1: Lexical Transformation table.

Input	**Parsed output**
cat	cat +N +SG
cats	cat +N +PL
geese	goose +N +PL
reading	read +V +Pres-part

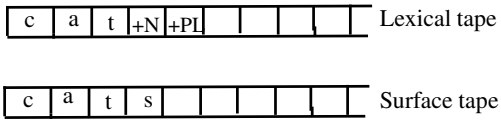

Figure 4.1: A FST.

The FST is a multi-function device, and can be viewed in the following ways:

- *Translator*: It reads one string on one tape and outputs another string,
- *Recognizer*: It takes a pair of strings as two tapes and accepts/rejects based on their matching.
- *Generator*: It outputs a pair of strings on two tapes along with yes/no result based on whether they are matching or not.
- *Relater*: It compares the relation between two sets of strings available on two tapes.

We define here, the FST as *Mealy machine*, which is an extension of normal finite state (FS) machine. The formal representation of Mealy machine is given by,

$$M = (Q, \Sigma, q_0, \delta, F) \tag{4.1}$$

where,

$$
\begin{aligned}
&Q = \{q_0, q_1, \ldots, q_{N-1}\}, \text{is finite set of states,} \\
&\Sigma \text{ is finite alphabet of complex symbols, and} \\
&\Sigma \subseteq I \times O, \\
&q_0 \text{ is start state,} \\
&\delta : Q \times \Sigma \rightarrow Q, \text{for example, } \delta(q', i : o) = q_j.
\end{aligned}
$$

The I and O are input and output symbols, and both include the symbol ε. For $\Sigma = \{a, b, !\}$, corresponding to the language discussed earlier, the FST has $i : o$ set as, $\{a{:}a,\ b{:}b,\ !{:}!,\ a{:}!,\ a{:}\varepsilon,\ \varepsilon{:}!\}$.

Like FSA (Finite State Automata) are isomorphic to regular expressions, the FSTs are isomorphic to *regular relations*. The FSTs are closed on the following relations:

1. *Union*: If R_1 and R_2 are relations on FST, then $R_1 \cup R_2$ is also a relation on FST.
2. *Composition:* If T_1 is FST from I_1 to O_1, and T_2 is FST from I_2 to O_2, then $T_1 \circ T_2$ is FST from I_1 to O_2.
3. *Inversion:* The FSTs are closed on *inversion*. A transducer T (or T^{-1}) simply switches the input and output labels.

The composition operation is useful because it replaces two FST running in series by a single FST. The composition works as in algebra. Applying $T_1 \circ T_2$ to input sequence S is equal to applying T_1 to S, and then T_2 to result $T_1(S)$, i.e.,

$$T_1 \circ T_2(S) = T_2(T_1(S)) \tag{4.2}$$

Similarly, the composition is useful to convert a FST as *parser* to FST as a *generator*.

In two level morphology, the lexical tape is composed of symbols from a in $a : b$ pairs, and the surface tape comprises the symbols from b in this pair. Hence, each symbol pair $a : b$ gives mapping from one tape to other tape. The symbols $a : a$ are called *default pairs* and written simply as a as shown in figure 4.2.

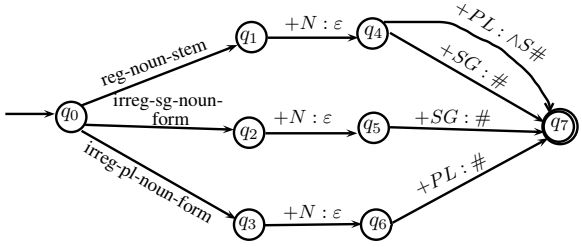

Figure 4.2: Morphological Parsing using FST.

The figure shows the transition diagram for FST with additional symbols $+SG$ (singular), $+PL$ (plural), corresponding to each morpheme. These symbols map to empty string (ε), as there are no corresponding symbols on output (surface) tape.

The symbol # stands for boundary symbol. Typical example of mapping, e.g., in case of word "geese" (irregular noun) on surface will parse into *goose +N +PL* on lexical tape, and symbols on the arc joining states $q_0 - q_2$ are "g:g o:e o:e s:s e:e", which is written as "g o:e o:e s e". Since, there are five letters in the

word, there will be five state transitions between $q_0 - q_2$. For regular noun, like fox, it will be "f:f o:o x:x". The surface form "geese" is mapped to lexical form "goose +N +SG" through *cascading* the FSTs, where two automata are run in series, i.e. output of first becomes input to next.

Instead of cascading two transducers, we perform this job using *composition* operator. Composing the transducers in this way helps in taking many different levels of input and outputs, and converting them into a single two level transducer with one input and one output tape. A typical FST, which results for morphological parsing of "cat" is shown in figure 4.3, producing a mapping *c:c a:a t:t +N:ε +PL:^S#*. The *+PL* maps to ^{S}. Symbol ^ indicates the morpheme boundary, and # indicates the word boundary.

4.1.1 Orthographic Rules

We note that concatenating the morphemes can work to parse the words like "dog", "cat", "fox", but this simple method does not work when there is spelling change, like "foxes" is to be parsed into lexicons "fox +N +PL" or "cats" is to be parsed into "cat +N +3SG", etc. This requires introduction of spelling rules (also called orthographic rules).

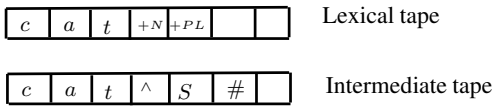

Figure 4.3: Morphological Parsing (Lexical and Intermediate tapes).

To account for the spelling rules, we introduce another tape, called *intermediate tape*, which produces the output slightly modified, thus going fom 2-level to 3-level morphology. Such a rule maps from intermediate tape to surface tape. For plural nouns, the rule states, "insert *e* on the surface tape just when lexical tape has a morpheme ending in *x* or *z* or *s* and next morpheme is -*s*". The examples are *ox* to *oxes*, and *fox* to *foxes*. The rule is stated as,

$$\varepsilon \rightarrow e / \left\{ \begin{array}{c} x \\ s \\ z \end{array} \right\} \hat{} \; - - - \; S\# \tag{4.3}$$

The equation 4.3 is called *Chomsky and Hall* notation. A rule of the form $a \rightarrow b/c - d$ means rewrite a as b, when it occurs between c and d. Since symbol ε is null, replacing it means inserting some thing. The symbol ^ indicates morpheme boundary. These boundaries are deleted by including the symbol $\hat{} : \varepsilon$ in the default pairs for the transducer. The ':' symbol separates the symbols on intermediate and surface tape. The mapping of symbols is shown in figure 4.4, called *morphological parsing*.

These multi-level FSTs in sequence between different tapes, as well as through parallel transducers for spelling checks, we are able to parse those words whose morphological analysis is simple. However, considering the sentence "The police books the right culprit", here it is not clear as per above rules that whether the lexical parser's output is "book +N +PL" or it is"book +V +3SG" ! However, to human it is not difficult to infer that it is the second. This is due to the ambiguity in the word, which may be a noun or a verb, depending on its position in a sentence. This type of ambiguity is called

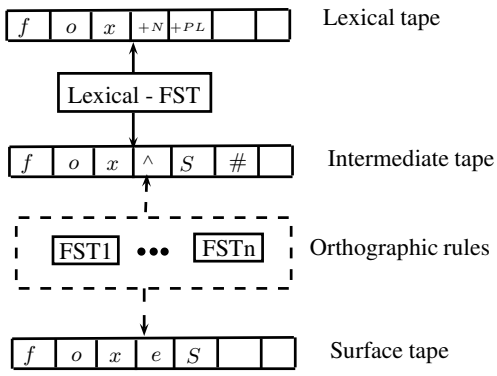

Figure 4.4: Morphological Parsing using 3-tape FSTs.

lexical ambiguity, and is the subject of later discussions.

Exercises

1. Demonstrate the morphological parsing for any five English language words.
2. Give your idea, as how you will modify the spelling rules of English language, to work in Hindi.
3. The alphabet symbols of *Devnaagri* as well as English have definite pattern/order on the keyboard and typewriters. Is this order important ? Give your arguments:
   (a) Is the letters' arrangement on the basis of similarity of sounds they produce?
   (b) Is this arrangement on the basis of their suitability for typing English words? That is, the symbols which come adjacent in words, are also placed closer on keyboards?
   (c) Is it on the frequency of occurrence of letters in words, so that frequently occurring letters are placed close to each other on the keyboard also?
4. Give some idea to resolve the morphemes to similarly pronounced words, like "bread", "dread", "red", "read", "raid."
5. Write a program /algorithm to produce your own version of stemer, which produces stem words from some nouns only.
6. What are the roles of lexical tape and surface tape in a finite state transducer? Explain.
7. Explain the *Morphological parsing* of following words using two level tapes FSTs: horse, donkey, house, cat, men.
8. What is difference between two-level and three level FSTs, used for morphological parsing? For the following words, which type of FST is required?

   book, booking, eat, ate, eaten
9. (a) What is Chomsky-and-Hall(CH) equation. How is is useful for defining orthographic rules? Explain.

   (b) Which of the following words' morphological parsing is generated by CH equation?

   cats, dogs, goose, foxes, oxes.

References

[1] D. Jurafsky and J. Martin, "Speech and Language Processing," *Pearson India*, 2002, Chapter 3.