one possible output string. Since ambiguity is a crucial property of natural language, it will be useful to have an extension of subsequential transducers that can deal with ambiguity but still retain the efficiency and other useful properties of sequential transducers. One such generalization of subsequential transducers is the *p*-**subsequential** transducer. A *p*-**subsequential** transducer allows for $p(p \geq 1)$ final output strings to be associated with each final state (Mohri, 1996). They can thus handle a finite amount of ambiguity, which is useful for many NLP tasks. Figure 3.11 shows an example of a 2-subsequential FST.

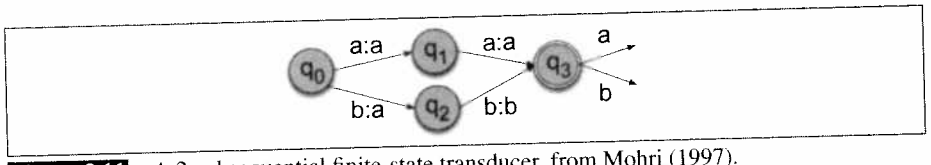

Figure 3.11 A 2-subsequential finite-state transducer, from Mohri (1997).

Mohri (1996, 1997) shows a number of tasks whose ambiguity can be limited in this way, including the representation of dictionaries, the compilation of morphological and phonological rules, and local syntactic constraints. For each of these kinds of problems, he and others have shown that they are **p-subsequentializable** and thus can be determinized and minimized. This class of transducers includes many, although not necessarily all, morphological rules.

3.5 FSTs for Morphological Parsing

Let's now turn to the task of morphological parsing. Given the input *cats*, for instance, we'd like to output *cat +N +Pl*, telling us that *cat* is a plural noun. Given the Spanish input *bebo* ("I drink"), we'd like *beber +V +PInd +1P +Sg*, telling us that *bebo* is the present indicative first person singular form of the Spanish verb *beber*, "to drink".

In the **finite-state morphology** paradigm that we use, we represent a word as a correspondence between a **lexical level**, which represents a concatenation of morphemes making up a word, and the **surface level**, which represents the concatenation of letters making up the actual spelling of the word. Figure 3.12 shows these two levels for (English) *cats*.

Surface level

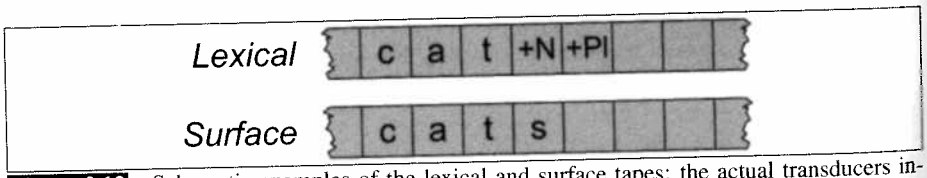

Figure 3.12 Schematic examples of the lexical and surface tapes; the actual transducers involve intermediate tapes as well.

For finite-state morphology, it's convenient to view an FST as having two tapes. The **upper** or **lexical tape** is composed from characters from one alphabet Σ. The

Lexical tape

lower or **surface tape** is composed of characters from another alphabet Δ. In the **two-level morphology** of Koskenniemi (1983), each arc is allowed to have a single symbol from each alphabet. We can then combine the two symbol alphabets Σ and Δ to create a new alphabet, Σ', which makes the relationship to FSAs quite clear. Σ' is a finite alphabet of complex symbols. Each complex symbol is composed of an input-output pair $i:o$, that has one symbol i from the input alphabet Σ, and one symbol o from an output alphabet Δ; thus, $\Sigma' \subseteq \Sigma \times \Delta$. Σ and Δ may each also include the epsilon symbol ϵ. Thus, whereas an FSA accepts a language stated over a finite alphabet of single symbols, such as the alphabet of our sheep language:

$$\Sigma = \{b, a, !\} \tag{3.2}$$

an FST defined this way accepts a language stated over *pairs* of symbols, as in

$$\Sigma' = \{a:a,\ b:b,\ !:!,\ a:!,\ a:\epsilon,\ \epsilon:!\} \tag{3.3}$$

Feasible pair

In two-level morphology, the pairs of symbols in Σ' are also called **feasible pairs**. Thus, each feasible pair symbol $a:b$ in the transducer alphabet Σ' expresses how the symbol a from one tape is mapped to the symbol b on the other tape. For example, $a:\epsilon$ means that an a on the upper tape will correspond to *nothing* on the lower tape. Just as for an FSA, we can write regular expressions in the complex alphabet Σ'. Since it's most common for symbols to map to themselves, in two-level morphology we call pairs like $a:a$ **default pairs** and just refer to them by the single letter a.

Default pair

We are now ready to build an FST morphological parser out of our earlier morphotactic FSAs and lexica by adding an extra "lexical" tape and the appropriate morphological features. Figure 3.13 shows an augmentation of Fig. 3.3 with the nominal morphological features (+Sg and +Pl) that correspond to each morpheme. The symbol ^ indicates a **morpheme boundary**, and the symbol # indicates a **word boundary**. The morphological features map to the empty string ϵ or the boundary symbols since no segment on the output tape corresponds to them.

Morpheme boundary

#

Word boundary

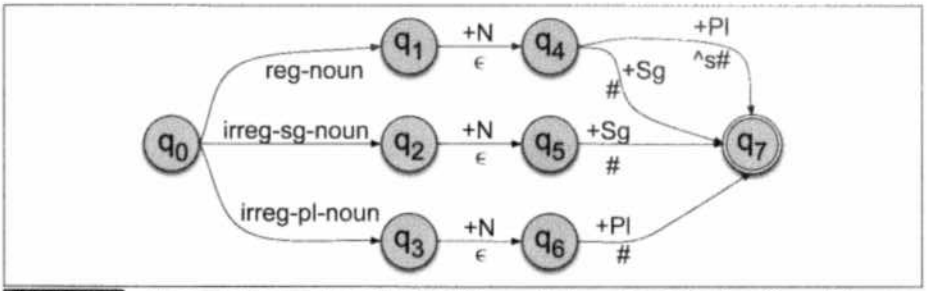

Figure 3.13 A schematic transducer for English nominal number inflection T_{num}. The symbols above each arc represent elements of the morphological parse in the lexical tape; the symbols below each arc represent the surface tape (or the intermediate tape, described later), using the morpheme-boundary symbol ^ and word-boundary marker #. The labels on the arcs leaving q_0 are schematic and must be expanded by individual words in the lexicon.

In order for us to use Fig. 3.13 as a morphological noun parser, it needs to be expanded with all the individual regular and irregular noun stems, replacing the labels

reg-noun, etc. To do this, we need to update the lexicon for this transducer so that irregular plurals like *geese* will parse into the correct stem `goose +N +Pl`. We do this by allowing the lexicon to also have two levels. Since surface *geese* maps to lexical `goose`, the new lexical entry will be "`g:g o:e o:e s:s e:e`". Regular forms are simpler; the two-level entry for *fox* will now be "`f:f o:o x:x`", but by relying on the orthographic convention that `f` stands for `f:f` and so on, we can simply refer to it as `fox` and the form for *geese* as "`g o:e o:e s e`". Thus, the lexicon looks only slightly more complex:

reg-noun	irreg-pl-noun	irreg-sg-noun
fox	g o:e o:e s e	goose
cat	sheep	sheep
aardvark	m o:i u:ϵ s:c e	mouse

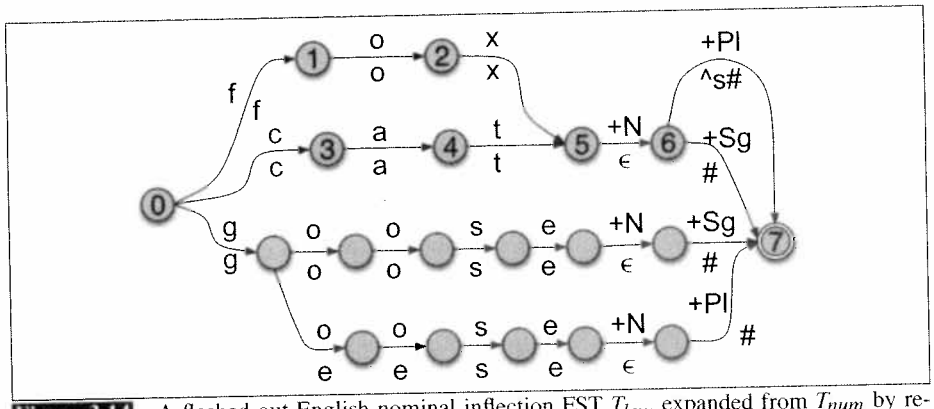

Figure 3.14 A fleshed-out English nominal inflection FST T_{lex}, expanded from T_{num} by replacing the three arcs with individual word stems (only a few sample word stems are shown).

The resulting transducer, shown in Fig. 3.14, will map plural nouns into the stem plus the morphological marker `+Pl`, and singular nouns into the stem plus the morphological marker `+Sg`. Thus, a surface *cats* will map to `cat +N +Pl`. This can be viewed in feasible-pair format as follows:

```
c:c a:a t:t +N:ϵ +Pl:^s#
```

Since the output symbols include the morpheme- and word-boundary markers ^ and #, the lower labels in Fig. 3.14 do not correspond exactly to the surface level. Hence, we refer to tapes with these morpheme boundary markers in Fig. 3.15 as **intermediate** tapes; the next section shows how the boundary marker is removed.

3.6 Transducers and Orthographic Rules

The method described in the previous section will successfully recognize words like *aardvarks* and *mice*. But just concatenating the morphemes won't work for cases in