Lecture 10

Beyond Phonology

At the very beginning of this course, we split language into the subareas phonetics, phonology, morphology, syntax, semantics, and pragmatics. Now that we have concluded our investigation of phonology, it is time to move on to a new area. We will start with morphology, which is very similar to phonology on a computational level, before moving on syntax, which will occupy us for the rest of the course.

1 Morphology

1.1 Morphology in Natural Language

Morphology is concerned with word formation, which can be further divided into two distinct subsystems. *Inflectional* morphology regulates the overt marking of agreement with other words in a sentence. For example, English has a very limited form of person and number agreement between the subject and the finite verb of a sentence.

(1) a. The man like**s** the children.
b. * The man like the children.
c. The men like the children.
d. * The men like**s** the children.

Other languages have a much richer system of inflectional morphology. In Icelandic, for example, adjectives agree with the noun they modify in number, gender, and case. In addition, they also agree in definiteness with the determiner. And Icelandic is still relatively tame from a typological perspective, thanks to its restriction to only two numbers, three person, three genders, and four cases. Other languages have over ten distinct genders, three numbers (singular-dual/paucal-plural), possibly four distinct persons, and a myriad of cases.

Derivational morphology consists of the rules for deriving new words from existing ones. This includes compounding — the adjective *smart* and the noun *phone* combine into the compound *smartphone*, which didn't exist until a few years ago — as well as changes in POS, such as turning the noun *sea lion* into the verb *to sea lion* (another neologism that refers to polite yet intrusive attempts to engage somebody in a debate). Derivational morphology thus provides a system for dynamically extending the lexicon of a language.

The term *to sea lion* was coined in 2014 following a popular Wondermark comic strip: http://wondermark.com/1k62/.

Note that the distinction between inflectional and derivational morphology has nothing to say about how these processes are marked. Number agreement in German, for instance, can be indicated via a suffix (*Frau* 'woman', *Frauen* 'women'), via a sound change (*Laden* 'store' and *Läden* 'stores'), a suffix and a sound change (*Haus* 'house', *Häuser* 'houses'), or neither (*Schlüssel* 'key' and 'keys'). Other languages may use prefixes, circumfixes, or infixes instead. The same goes for derivational morphology. English often uses a suffix (*quick* and *quickly*, *the haste* and *to hasten*), sometimes a shift in stress (*the export* and *to export*), and sometimes no marker at all (*the anger* and *to anger*). In addition, compounding has no overt marking in many languages beyond the adjacency of the words, while many languages use *reduplication* to create new words (Marshallese *kagir* 'belt' and *kagirgir* 'to wear a belt'). In sum, morphological processes can be realized overtly via affixation, compounding, reduplication, or some phonological process.

The interaction of morphology and phonology is symmetric, though, in the sense that morphological structure can also suspend or trigger phonological constrains and processes. For instance, consonant clusters can have greater complexity if they span a morpheme boundary. This can be seen in German, where /tstpR/ cannot occur in any monomorphemic word but is perfectly acceptable in the compound *Arztpraxis* 'doctor's office'. The interaction of morphology and phonology is also known as *morphophonology* or, slightly shortened, *morphonology*. The close interaction of those two domains, with morphology sometimes relying phonological processes and phonology being sensitive to morphological information, means that the two are often treated by one and the same mechanism in real-life applications.

1.2 Two-Level Morphology

Just like phonology, morphology has been analyzed in terms of rewrite rules for the largest part of recent history. To the best of my knowledge, morphologists instinctively followed the same ban against a rule rewriting its own output that guaranteed the regularity of phonology. This makes it very likely that morphology, or at least a large part of it, is regular, too. So the finite state methods we used for phonology can be used just as well for morphology, which makes it possible to have one big rewrite grammar that intersperses phonological and morphological rewriting to capture the interaction of the two. This collection of rewrite rules can then be converted into a single FST thanks to the closure of finite state transductions under composition.

Contrary to what one might expect, though, this is not quite the approach taken in most industrial applications. This is mostly for historical reasons. Composing a large number of transducers, many of which may have dozens of states, simply wasn't computationally feasible before the 90s. In addition, people had not figured out yet how such a transducer can be used efficiently for morphological analysis, rather than generation.

Suppose you have a phonological surface form *s* for a word that sounds like *wipeboard*. Then *s* could be simply a compound of *wipe* and *board*, or a compound of *white* and *board* where /t/ was replaced by /p/ due to the following /b/. Now if you take the FST for your rewrite grammar and construct its inverse, which maps surface forms to underlying forms, it can return either option as a possible underlying form (since the FST is non-deterministic, even getting this set of all possible underlying forms is not trivial). In order to determine the correct underlying form, you then have to lookup each form in the lexicon, where you will eventually find an entry for *white board* but none for *wipe board*. Keep in mind that a lexicon can contain hundreds of thousands of entries, so this search can also take a long time unless one uses a hashtable, which wasn't feasible back then due to memory limitations.

Nowadays the solution is obvious: take the identity function over the lexicon and compose it with the FST for the grammar. If this composed FST is run in reverse, it only outputs items in the lexicon, so that no search is needed. But this simple trick wasn't known until the mid 90s, and instead *two-level morphology* was developed as a more practical tool for morphological analysis (Koskenniemi 1983).

From a formal perspective, two-level morphology is a very simple modification of the rewrite paradigm. Rather than applying rewrite rules sequentially, one after another, they are all applied in parallel. Consequently, the grammar isn't a cascade of FSTs but runs all FSTs in parallel instead. The overall transduction is the intersection of these FSTs. Recall that the intersection of arbitrary FSTs is not guaranteed to be an FST. However, the class of ε-free FSTs is closed under intersection, and two-level morphology exploits this fact by using a special symbol 0 that is used to indicate unpronounced nodes. That way, deletion of a symbol σ amounts to relabeling it as 0, and insertion of σ is emulated by relabeling 0 as σ. A rather inelegant trick, but it gets the job done.

As is implied by the name, two-level morphology posits only two levels, one for underlying forms, one for surface forms. For example, the word *moved* may be analyzed as the underlying form *move+ed* and the surface form *mov00ed*. Since the two forms always have the same length, they can be analyzed as a single string of pair symbols (similar to how we formalized strictly local languages with a hidden alphabet).

The rewrite rules operate over such pair symbols $u : s$, where u is the underlying segment and s the surface segment. The rewrite rules are of the form $u : s \diamond \alpha\_\beta$, where α and β are strings over pair symbols (and may also include SPE-style notation like brackets for optionality and + for iteration). The symbol $\diamond$ is a placeholder for two different types of rewrite arrows: $\Rightarrow$ and $\Leftarrow$. If $\diamond$ is replaced by $\Rightarrow$, then the rule states $u : s$ can occur only in the specified context. This requirement is weakened with $\Leftarrow$, which states that if both $u : s$ occurs in the specified context and the surface form has an s, then the underlying form must have u. This is also called *surface coercion*. Sometimes $\Leftrightarrow$ is used as a combination of the two to express that $u : s$ occurs only in the given context and no distinct form $u' : s$ may occur there.

With a little bit of ingenuity, each rewrite rules can be converted into a regular language of pair symbols, and the whole grammar is simply the intersection of these regular languages. Analyzing a surface form is tantamount to finding a pair string whose second component matches the surface form, whereas generation instead looks for a pair string whose first component matches the desired surface form. Since a language of pair strings can be converted into an ε-free FST, this search is simply a matter of running the FST over the underlying form, or its reverse over the surface form.

In sum, two-level morphology may look very different from SPE or OT, yet it is just another way of defining finite state transductions. While it can generate all regular languages, just like SPE and OT, it is weaker than those two in the sense that it cannot handle deletion and insertion in an elegant way and must instead rely on padding out strings via the special symbol 0.

1.3 Complexity of Morphology

Two-level morphology has been successfully applied to a variety of typologically diverse languages, including English, Finnish, Turkish, and Japanese. Just like the overwhelming descriptive success of SPE is strong evidence that phonology is at most regular, the wide usage of two-level morphology suggests that morphology is regular, too.

This is actually not all that surprising, considering the local and finitely bounded nature of most morphological processes. In almost all cases it is sufficient to know the modified stem and which morphological process took place most recently. In addition, some morphemes can only be instantiated exactly once. All of these things fall under the purview of regular languages.

Only two aspects of morphology might be problematic. First, circumfixion requires the presence of both a prefix and a suffix of a specific type. This is illustrated by the German past participle, which consists of the stem of the verb and the circumfix *ge- -t* as in *ge-kauf-t* 'bought'. If this process were unbounded, then German would contain words of the form *ge*^{n}*-kauf-t*^{n}, but not, say, *ge-ge-kauft-t-t-t*. In our discussion of the complexity of syntax later on we will see that such a pattern is not regular. Intuitively, that's because an FST generating this pattern would have to first inserts n instances of *ge-*, followed by n instances of *-t*, but since the FST has only a finite number of states it can't keep track of the exact number n past a certain threshold and may end up inserting too few or too many suffixes. As you might expect, though, German past participle formation is not an unbounded process (probably because additional circumfixes would serve no function). To the best of my knowledge, unbounded circumfixion is universally unattested, lending strong support to the hypothesis that morphology is regular and thus incapable of generating such patterns.

The only remaining challenge to the regularity hypothesis is reduplication. If there is no upper bound on the size of the reduplicant, we run into a similar memory problem for the FST as above: with n states, the FST can memorize only a fixed amount of information, so if the material that should be reduplicated is so long that it does not fully fit in the state memory, the FST cannot insert an exact copy. Reduplication data has proven very difficult to analyze, and at this point it is not clear whether there are any cases of unbounded reduplication where the reduplicant must be an exact copy. In the cases where reduplication seems to be unbounded, it usually interacts with phonology in various ways that make it difficult to discern what the morphological well-formedness criterion is. Putting aside reduplication, though, all of known morphology seems to fall within the class of finite state transductions. And just like in phonology, the overwhelming majority of processes do not come close to exploiting the full power FSTs.

2 Syntax