The kind of phonology we learned about in the previous activities is called *derivational phonology*. It focuses on how we can explain rules that *derive* surface pronunciations (e.g., if a phoneme /k/ occurs after a nasal consonant in Maasai, it should be pronounced as [g] instead; this is a description of how the pronunciation [g] is *derived* based on the underlying phoneme and the context). But there is another approach to phonology, called *Optimality Theory*. In this reading you will learn about Optimality Theory. But first, let's consider why another approach to phonology is needed—what are the limitations of derivational phonology?

Phonological conspiracies, and the need for explanation

Example one: Sprite

Think about the soda called *Sprite* (雪碧). This English word is difficult for speakers of many languages to pronounce. Do you know how *Sprite* is pronounced by speakers of other languages?

When native Spanish speakers speak English, they often pronounce *Sprite* more like "*Esprite*". Spanish speakers have a hard time with consonant clusters (more than one consonant together) at the beginning of a word[1]; for example, they also often pronounce *special* as "*especial*".

[1] Technically the issue is about a consonant cluster at the beginning of a syllable; it doesn't really matter whether it's at the beginning of a word. But we haven't learned about syllable structure in this subject, so to keep

What about speakers of other languages? Take a look at this video to see how *Sprite* is pronounced by someone with an Arabic accent[2]: https://www.youtube.com/watch?v=lqJDuZIcQ34. He mentions Sprite around 2 minutes 15 seconds into the video. Is his pronunciation of *Sprite* different from a Spanish speaker's pronunciation of *Sprite*?

You should have noticed that while Spanish speakers pronounce it like "*Esprite*", Arabic speakers may pronounce it like "*Suprite*". (Korean speakers also often pronounce this like "*Suprite*".) The big question is, why?

It's clear that speakers of both these languages face the same underlying problem: it's difficult to pronounce a consonant cluster like *spr* at the beginning of a word. In their languages, consonant clusters like that never occur at the beginning of the word. So, to pronounce *Sprite*, they end up making some change to avoid having to pronounce this difficult thing. Spanish speakers solve the problem by adding an extra vowel at the beginning of the word; now *spr* is not at the beginning of the word anymore, so it's not so hard to pronounce. On the other hand, Arabic (and Korean) speakers solve the problem by adding a vowel in the middle of the cluster to break it up, so there no longer is a cluster *spr* at the beginning of the word; the only consonant at the beginning of the word is *s*,

---

things simple we can just think about this as an issue about consonant clusters at the beginning of a word.

[2] This comedian isn't actually an Arabic native speaker, I'm pretty sure he's an English native speaker (he's American, and his parents are Lebanese and Iraqi). Here he is just mimicking a typical Arabic accent.

which is not hard to pronounce. Both the Spanish speakers' solution and the Arabic speakers' solution resolves the problem. But why do speakers of these different languages resolve the problem in different ways? If we want to understand how language works, and how and why languages differ, then we need an answer to that. With derivational phonology, we could write a rule describing what happens in each language, as shown below (very roughly; here I use "#" to indicate the beginning or end of a word, and "C" to indicate any consonant):

Spanish: #CC --> #eCC
"When there are two consonants at the beginning of a word, an 'e' is inserted before them."

Arabic: #CC --> #CuC

"When there are two consonants at the beginning of a word, a 'u' is inserted between them."

These rules can accurately describe what happens in each language but it's not clear how we could explain *why* these happen, or why the languages are different.

Example two: Yawelmani

For another example, let's look at Yawelmani, an Amerindian language spoken around California. This language has several interesting phonological changes. Here are three of them. My description of these phonological patterns is based on the analysis by Kisseberth (1970).

Vowel insertion: CC# --> CiC#

"If there would be two consonants together at the end of a word, then it will be pronounced with a vowel 'i' between them."

Consonant deletion: CC+C --> CC+

"If putting two morphemes together [here the '+' indicates a boundary between two morphemes, such as a prefix and a main word] would cause there to be three consonants next to each other, then the last consonant will not be pronounced."

Vowel deletion: VCV# --> VC#

"If there would be a vowel-consonant-vowel sequence at the end of a word, the last vowel is not pronounced." For example, if a speaker has the word *taxa* and then adds the suffix *-ka*, making a word *taxaka*, they will actually pronounce it as *taxak* (notice that the last vowel is not pronounced). Importantly, this does not occur if there would be a *consonant*-consonant-vowel sequence at the end of the word. For example, if a speaker has the word *xat* and then

adds the suffix *–ka*, making a word *xatka*, they will NOT pronounce it as *xatk*. Instead, they will pronounce the full word *xatka*, including the final vowel.

These are three patterns of phonological change that happen when speakers of Yawelmani speak their language. At first glance, these three patterns may appear quite different. But if you look more closely, you may see that they share something in common. All of them are meant to make sure that there is never a cluster of two adjacent consonants in the same syllable. Let's look at these more closely.

The first rule, vowel insertion, puts a vowel in between two consonants at the end of a word. If two consonants are at the end of the word, they have to be in the same syllable. (Think of an English word like *tent*; it is not possible to break the *n* and the final *t* into different syllables.) It is difficult to pronounce a cluster of two consonants in the same syllable, so Yawelmani speakers put a vowel in between them, which allows them to be broken into two syllables. Note that this is only necessary if they're at the edge of a word. If there are two consonants in the middle of a word, they could be broken into two syllables. Think of an English word like *doorpost*; *r* and *p* are next to each other, but they're in different syllables, since the first syllable of the word is *door* and the second syllable is *post*. In fact, even Chinese languages can have two consonants next to each other, as long as they're in different syllables. Consider the Mandarin phrase 很棒, *hen bang*. *n* and *b* are

adjacent to each other, but they're in different syllables so it's possible to pronounce. But Mandarin can never have two adjacent consonants within the same syllable (Mandarin doesn't have any syllables like *stu*). So this all explains why Yawelmani needs to break up two consonants (by putting a vowel between) if they're at the edge of a word, but it's fine having two consonants together if they're in the middle of a word, where they may be in different syllables.

What about consonant deletion? That rule states that if there are three consonants together in Yawelmani, one of them gets deleted. This, again, happens because Yawelmani speakers do not want to have a cluster of two consonants next to each other in the same syllable. If you have three consonants, this is unavoidable. Imagine that I have a sequence of a vowel, three consonants, and another vowel: VCCCV. Since every syllable needs a vowel, there are two syllables here. Here are the possible ways this sequence could be divided into syllables:

- [V] [CCCV]
- [VC] [CCV]
- [VCC] [CV]
- [VCCC] [V]

As you can see, no matter how we divide it, one of the syllables will always have two or more consonants next to each other. And that is difficult to pronounce. So Yawelmani speakers solve the problem by deleting one of those consonants. After that, they can be spread out across different syllables: [VC] [CV]. So we see that

consonant deletion and vowel insertion, even though they look quite different, are actually ways of solving the same problem.

What about the last pattern, vowel deletion at the end of a word? This is also related to the same issue. Yawelmani speakers don't need to pronounce the vowel at the end of *taxaka*, because even if they don't pronounce it, they will have *taxak*, a word where there are never two consonants adjacent to one another in the same syllable. (The syllables are [ta] [xak].) But for *xatka*, they need to pronounce that last vowel. If they don't pronounce it, they would be left with *xatk*, and that word has *t* and *k* next to each other in the same syllable, which would be hard to pronounce. Therefore, they have to pronounce the vowel, so that the word can be pronounced [xat] [ka], without any consonants adjacent to one another in the same word.

What we see, then, is that all three patterns we saw in Yawelmani are aimed at the same goal: avoiding having a consonant cluster within a syllable. They are just three different ways to accomplish that goal. This should look similar to the case we saw with *Sprite*, where there were two different ways to avoid pronouncing *spr* at the beginning of a word. In that case, Spanish and Arabic speakers avoid the consonant cluster in different ways. In the case of Yawelmani, speakers of the same language avoid consonant clusters in different ways, depending on the context (i.e., depending on where in a word they are happening).

This situation is called a ***phonological conspiracy***. Several different processes (e.g., vowel insertion, consonant deletion, and

vowel deletion) are teaming up to ensure the same result—in this case, these three processes are teaming up against consonant clusters. In other contexts, "conspiracy" also refers to a lot of people or groups secretly coming together for some goal. (e.g., a "government conspiracy" is when lots of people in the government are trying to secretly do something bad. You may have heard of "conspiracy theorists", who believe in weird things—e.g., believing that the earth is flat, or believing that the government is controlled by lizard people who are disguised as humans. These people always believe that there are lots of secret organizations working together to hide the truth from us.) Phonological conspiracy is the same idea: so many different phonological processes working together to suppress the poor consonant clusters.

Phonological conspiracies are a case where derivational phonology does not do a good job explaining what is happening in language. The three phonological rules I wrote above for Yawelmani are based on derivational phonology. Looking at those rules, it is hard to see how they are connected, and hard to notice that they are all motivated by the same thing. It would be better if we had a way of looking at phonology that was focused not on how sounds change, but on what goals speakers are trying to accomplish. This is one of the main reasons that *Optimality Theory* was developed.

Conflicting goals: markedness and faithfulness

The core of Optimality Theory is the idea that when we speak language, we always have to compromise between two goals. It is impossible to satisfy both of them, so when we speak we strike what we consider the best balance (the "optimal" balance) between the two goals.

The first goal is to make speaking as easy as possible. Some sounds, or combinations of sounds, are pretty hard to pronounce. For example, the retroflex sound in Mandarin is hard to pronounce; click sounds (present in some languages of southern Africa, such as the Xhosa language) are also hard to pronounce. Furthermore, a string of a lot of consonants together is also hard to pronounce—we saw that above, with examples like *Sprite*. (Another good example is *strengths* and *eighths*; it is pretty difficult to pronounce these.) Your tongue will get tired pronouncing that stuff a lot. Therefore, speakers often want to be "lazy": to avoid pronouncing difficult things, or to simplify them. This can make speech more efficient and faster. In Optimality Theory, this concern is called *markedness*. Sounds that are difficult to pronounce are considered "marked" (meaning unusual), and speakers try to avoid pronouncing things that are marked.

For a concrete example, consider a word like *bat*. Technically, *t* is a stop, which should normally be released (it should have a small puff of air at the end). If you want to pronounce *bat* very carefully, what you end up saying is kind of like "*batuh*", because you have to let the air puff out after releasing the stop. In reality, though, we often do not do this. We often pronounce *bat* with an *unreleased stop*: when pronouncing the *t*, we put our tongue against

the top of our mouth to stop the air going out, but we don't open it again to let the little burst of air out. If you speak any Cantonese, this should sound familiar; *p*, *t*, and *k* at the end of Cantonese syllables (such as 落) are also pronounced in this unreleased way. This happens because of markedness: pronouncing *bat* very clearly, like "*batuh*", would take too much effort, so usually we don't bother to pronounce the release.

Markedness cannot be speakers' only concern, however. If it were, all language would just be the simplest possible sounds; every language would be nothing but "babababababababa", which is easy to pronounce. Clearly this is not really how language works. So, if speakers are lazy and avoid pronouncing marked sounds, why haven't all languages turned into "babababababa" by now?

This is where the second motivation comes in. Speakers want to be lazy, but speakers also want to be understood. To be understood, we need to pronounce things in pretty much the way that our listener expects. If we change the pronunciation too much, people won't understand us. Think of the *bat* example from above. As described above, because of the desire to avoid markedness, we usually pronounce this word a bit lazily, with an unreleased *t* instead of a released *t*. But if we simplified the word even more and pronounced it as "*ba*", the person we're talking to might not even understand what we were trying to say. If we pronounce it as "*ba*", we have simply changed it too much for our listener to recognize the word we were pronouncing. Therefore, in order to be understood, we have to avoid changing a word's pronunciation too much. In

Optimality Theory this concern is called *faithfulness*: when we speak, we have to try to be faithful to the way people expect the word to be pronounced.

These two goals usually come into conflict. If we concern ourselves completely with faithfulness, we will end up speaking pretty slowly and effortfully as we take care to pronounce everything exactly as others expect it to be pronounced. On the other hand, if we concern ourselves completely with markedness, then we will end up pronouncing everything as "babababababa" (or something like that) and nobody will have any idea what we're trying to say. So in reality, in order to communicate, we have to strike some balance between these.

The key idea of Optimality Theory is that speakers of different languages strike the balance in different ways. In some languages, faithfulness ends up being slightly more important than markedness; in other languages, markedness ends up being slightly more important. Of course, the balance may be struck differently for different aspects of languages. In a given language system (i.e., a given system of grammar), markedness might be very important for consonants, but not so important for vowels. Or a language might value faithfulness very highly when it comes to sounds at the end of a word, but not when it comes to sounds at the beginning of a word. There are many, many possible combinations of faithfulness-markedness balances that could be struck. The proposal of Optimality Theory is that all differences between languages (or even between speakers) are differences in how the languages rank (or weigh) the

conflicting demands of faithfulness and markedness in different situations.

Example analyses

While Optimality Theory was initially developed as a way of explaining phonological systems, and is still used in phonology more than it is in other fields, it actually is a general analysis that could apply to everything. Optimality Theory has also been used to explain syntax patterns, etc. We could even use Optimality Theory to describe how we choose a restaurant to go to or a city to travel to.

Example one: Travel

Imagine that you and your friends want to go traveling, and you can go to either Beijing, Mumbai, or New York (maybe there are some travel deals for cheap tickets for these cities). How will you decide which place to go to? You have to consider what desires or goals you and your friends have. Let's assume you and your friends have three main concerns:

1. You want to go somewhere within 5 hours' flight of Hong Kong, because you don't want a long plane flight.
2. You want to go somewhere where the food is not very spicy, since some of your friends 怕辣.

3. You want to go somewhere that's not in China, because you want some new experience.

It should be obvious that no choice will satisfy all your needs. Beijing is ruled out because it's in China. New York and Mumbai are ruled out because they're far. And Mumbai is also ruled out because most of the food there is spicy. So you can't have any perfect choice; instead you need to choose which place is the *optimal* balance between your desires.

What is optimal will depend on which desire is the most important. If the most important desire is that you don't want to fly more than 5 hours, then you will have to go to Beijing; New York and Mumbai are too far away, and then there's no need to even consider the other issues (spiciness and Chinese-ness) since those cities are already off the table.

When doing an analysis with Optimality Theory, we usually represent our choices (called *candidates*) and our desires (called *constraints*) in a table, like the one below:

	NotFar	NotSpicy	NotChinese
Beijing			
Mumbai			
New York			

The list of cities on the left is the cities we are considering going to. The list of constraints along the top is the things that we want in our travel. I have arranged the constraints from the most important (NotFar, on the left) to the least important.

The process of choosing which city to go to, then, is based on looking at each constraint one at a time and seeing which cities are ruled out by it. Once there's only one city left, that's what we choose. In this case, first we look at "NotFar" and decide which cities meet the constraint. If a city is NotFar, we don't write down anything; but if a city is not NotFar (i.e., if the city is more than 5 hours away), we mark down a "*" to show that we are kicking that city out:

	NotFar	NotSpicy	NotChinese
Beijing			
Mumbai	*		
New York	*		

In this case, both Mumbai and New York are more than 5 hours' flight away from Hong Kong, so they are going to be kicked out. Beijing is fine.

Now actually our problem is solved: Beijing is the only city left, so that's where we are going to. Usually we mark this with a ☞ symbol to indicate that it is our "optimal choice":

	NotFar	NotSpicy	NotChinese
☞ Beijing			
Mumbai	*		
New York	*		

In fact, Beijing is also not a perfect place to go; if we look at the rest of the constraints, we will see that Beijing violates the

"NotChinese" constraint, as shown below. But that doesn't matter, since we already decided (based on the NotFar constraint) that Beijing is the only place to go. Usually we shade in these later constraints to indicate that they don't matter; after looking at the NotFar constraint, the decision has already been made.

	NotFar	NotSpicy	NotChinese
☞ Beijing			*
Mumbai	*	*	
New York	*		

As you may have noticed, this conclusion totally depends on our subjective judgment of which constraint is most important. If I'm planning travel with my friends and we decide that the most important thing is that we don't want to fly far, we will choose Beijing. But what if I'm planning travel with a different group of friends and they have different priorities? What if their top priority is that they don't want to go somewhere in China, and the next priority is that they don't want too much spicy food, and the last priority is that they don't want to fly far? Then the analysis would work in the same way, but we'd start out with a different table, reflecting the different ranking (or weighting) of our priorities:

	NotChinese	NotSpicy	NotFar
Beijing			
Mumbai			
New York			

Using this table, can you do an Optimality Theory analysis and figure out which city the group will go to? Check the next page for the answer.

	NotChinese	NotSpicy	NotFar
Beijing	*		
Mumbai		*	*
☞ New York			*

As shown above, in this situation we would choose to go to New York. The most important thing is we don't want to go to China, and that rules out Beijing. Next, we don't want to go somewhere with mostly spicy food, so that rules out Mumbai. After that, all that's left is New York, so we don't really care that it's far away.

Let's make one last finishing touch to the table. Usually in Optimality Theory, we put a "!" next to the place where each city got ruled out (this is called the "fatal violation"). For example, for Mumbai, the reason we're not going to Mumbai is because it violates the NotSpicy constraint; that's the first (most important) constraint that Mumbai goes against. Mumbai happens to also be too far away, but we don't really care about that; Mumbai was already off the table when we realized that it has a lot of spicy food. Putting a "!" there, as shown in the table below, helps us see that.

	NotChinese	NotSpicy	NotFar
Beijing	*!		
Mumbai		*!	*
☞ New York			*

It could be possible that a city might get a * but not be ruled out, because the other cities are just as bad. Try doing an Optimality Theory analysis with the constraints ranked NotChinese > NotFar > NotSpicy (i.e., NotChinese is the most important, and NotSpicy is the least important) and you will see. In that case, the first place where Mumbai gets a "*" won't be the fatal violation (i.e., it won't get a "!"), because New York also has a "*" there. Then the decision will come down to the next constraint.

Of course, when you really make a decision with your friends, you probably don't sit down and make a table like this. But the claim of Optimality Theory is that you actually do think like this, using this sort of logic; it's something that you do unconsciously without being aware of it. Optimality Theory is just a way for us to make this thought process explicit.

Hopefully it is clear by now that the thought process for choosing a city to travel to is analogous to the thought process for choosing how to pronounce a word. In this example, the "candidates" we were choosing between were different cities we might go to; in phonology, the "candidates" are different ways you might pronounce a word. In this example, the "constraints" were what we want or don't want in our travel experience; in phonology, the "constraints" are various kinds of markedness (difficult-to-pronounce things that you want to avoid) and faithfulness (ways that you want to avoid changing the pronunciation, so that people don't misunderstand you). Let's see how this can work with the *Sprite* example we discussed above.

Example two: Sprite

Recall that Spanish speakers pronounce *Sprite* as *Esprite*, and Arabic speakers pronounce it as *Suprite*. Both groups of people don't like saying *spr* at the beginning of a word, but they use different strategies to get around that problem. How can we look at this situation using Optimality Theory?

First let's think of the ways a speaker could pronounce the word. They could pronounce it as *Sprite*, like it's pronounced in English. Or they could pronounce it as *Suprite* or *Esprite*. (Of course, there are infinite other ways they could pronounce it, but most of them we don't need to pay attention to. They could pronounce *Sprite* as *furglededoopyakkawut*, but that would be such a serious violation of faithfulness that we won't even bother considering it.) So we can make an Optimality Theory table like before, using possible pronunciations in the place where we had cities:

Sprite			
Esprite			
Suprite			

Next we need to think about what constraints people might be dealing with when they want to pronounce this. There are hundreds or thousands of constraints that influence how people pronounce words, but here we only need to think about the ones that are directly relevant to the pronunciation of this word. Just like in the

above example I listed three goals for our travel, here let me propose three constraints (goals) governing our pronunciation:

1. Let's not add extra sounds at the beginning of the word (because of faithfulness: adding extra sounds may make people misunderstand us).
2. Let's not add extra sounds in the middle of the word, either.
3. Let's not pronounce *spr* at the beginning of a word (because of markedness: it's too hard to pronounce).

Once again, it's impossible to satisfy all of these constraints. Any pronunciation will violate at least one of these. So we have to decide which constraint is most important to us. Let's see what happens if we put the constraints in the order above (not adding sounds at the beginning of a word is the most important concern, not adding vowels in the middle of the word is next, and not saying *spr* is the least important concern):

	NoAddBeginning	NoAddMiddle	NoSPR
Sprite			
Esprite			
Suprite			

If we go through the table and do an analysis like we did for cities, we end up with the following:

	NoAddBeginning	NoAddMiddle	NoSPR
☞ *Sprite*			*
Esprite	*!		
Suprite		*!	

Esprite is ruled out because it violates the most important constraint; we won't say *Esprite*, because we really do not want to add another sound at the beginning of a word. Next, *Suprite* is ruled out because it violates the second most important constraint. After that, only *Sprite* is left, so that is what we will pronounce—we don't care that it has an *spr* which is hard to pronounce (violating the last constraint), because it's still better than any of the other options. Thus, organizing the constraints in this way gives us the English pattern: apparently when it comes to these constraints, English speakers care about faithfulness more than markedness (at least for these particular sounds and these particular places within a word).

Like we have seen above, Optimality Theory predicts that if we have the same constraints but rank or weight them differently, we will make a different choice. If we decide we care more about avoiding China than we do about keeping a short flight, then we will go to New York instead of Beijing; likewise, if we care more about not saying *spr* than we care about not adding extra sounds, we might end up pronouncing *Sprite* differently. See if you can figure out how to rearrange the order of the three constraints to get the Arabic pattern and the Spanish pattern for how people pronounce *Sprite*.

References

Kisseberth, C. (1970). On the functional unity of phonological rules. *Linguistic Inquiry, 1*, 291-306.