Chapter Two
Syllable Structure in Taifi Arabic from an Optimality Theory Perspective

2.1 Introduction

In the previous chapter, we saw that the syllable is recognised as an essential unit of the theory of grammar. And, we looked at the main principles and the essence of Optimality Theory as a potential framework that might provide an adequate analysis of the various grammatical properties in general, and of syllable structure in particular. Our main objective in this chapter is to investigate this potentiality and see whether or not we may depend on OT to generate syllable structure cross-linguistically and apply that general perspective on a particular language. Consequently, this chapter will be divided into two sections: one involves a mere general theoretical presentation, and the other involves an application of that theory on a particular language, Taifi Arabic.

In the first section we will present syllable theory as viewed within OT. Yet, before that, we ought to have a clear understanding of what is called Jakobson Typology that presents the possible core syllables of the languages of the word and of the basic syllable structure universal constraints that have to be language-particularly ranked to determine the optimal syllable types of a given language.

In the second section, however, we will apply syllable theory to facts known about Arabic in general that are quite applicable to Taifi Arabic. Through this application, we will be able to see whether our main claim about OT as a potential framework for the analysis of syllable structure stands or not. That is, can we generate or determine, if I should say, the main syllable types of this language by ranking the basic syllable structure constraints in an order of dominance that suites this particular language, or can we not? This will be the

first challenge, in this study, with which OT has to deal rather formally due to the fact that it is a very fundamental issue.

2.2 Syllable Theory

2.2.1 Jakobson Typology

According to Clements and Keyser (1983), the primary set of core syllable types, cross-linguistically, contains the following sequences:

(1) a. cv
b. v
c. cvc
d. vc

(Clements and Keyser 1983, p.28)

They reached such a conclusion by considering a claim made earlier by Jakobson (1962) that languages neither forbid onsets nor require codas for all their core syllable types. This means that all languages have onsetful syllables, open ones, and may allow nothing else, which means that languages may have optional consonant-initial syllables but never ban them, and optional vowel-final ones but never require them. So, if we consider the inventory in (1) above along with Jakobson's claim, we will find that (1) a. cv is the least marked syllable type and (1) d. vc is the most highly marked one cross-linguistically. To summarise, consider the following cv syllable structure typology:

(2)

		Onsets	
		Required	Optional
Codas	Forbidden	Σcv	Σ(c)v
	Optional	Σcv(c)	Σ(c)v(c)

(Prince and Smolensky, 1993, p.85)

Apparently, there are two operations, according to Clements are Keyser (1983), that are responsible for producing the three less natural core syllables from the most natural one cv. The first involves deleting the initial consonant of the cv syllable type to have v, or to be the first of two steps towards producing vc. The second one involves adding a final consonant to the cv syllable type to have cvc or to be the second step of the two that yield vc. This means that the process of basic syllabification can involve deletion and/or insertion, that we will later call "underparsing" and "overparsing" respectively.

By this, it became rather clear that there are four possible core syllable types containing the two extremes of markedness cross-linguistically. The question to address now is: what are the basic syllable structure constraints that will generate this typology? This is a question related to the very heart of the discussion in this chapter and is worthy of consideration. So, the next section will be capitalising on the issue of the constraints involved in the syllable structure determination.

2.2.2 Basic Syllable Structure Constraints

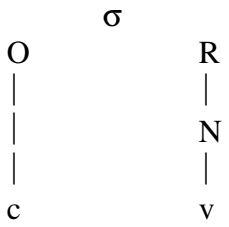
In this section, we will focus on the set of constraints that are employed in the process of determining the set of possible core syllables, presented above. Following Prince and Smolensky (1993) we will present these constraints as members of two families: Structural constraints and Faithfulness constraints. And, we will also find that some of these constraints are put in a superordinate position while the others may be relatively ranked to meet language-particular needs.

Due to the fact that these constraints are proposed to help us determine the set of possible core syllables, they will aim at the least marked core syllable, cv, and try merely to describe it. What I mean to say is that since cv is at the very positive extreme of optimality, markednesswise, the constraints, specially the structural ones, will be telling

us that every syllable ought to be cv. Of course, this is not the case as we have, in natural languages, other syllable types. Here comes one of the main principles of OT, i.e., violability, to solve this conflict. We are not aiming to achieve the ideal or the perfect, i.e., every syllable must be of the type cv; we are only trying to have the optimal that may always minimally violate some of the constraints that generate the perfect form.

Based on what we have said, let us now try to figure out the structural constraints that will generate cv. If we consider this syllable type, we will notice some peculiar structural properties. The first thing we notice is that it has an onset and a nucleus, but it does not have a coda. Another thing we notice is the lack of complexity in the association lines between the structural tier and the cv tier which means that only one c or v associates to a syllable position node:

(3)



Finally, we notice that the nucleus is a vowel, while the onset is a consonant. Out of these observations, we may postulate the following structural constraints:

(4)

a. NUC: Syllables must have nuclei
b. * COMPLEX: No more than one C or V may associate to any syllable position node[1].
c. * M/V: V may not associate to margin nodes (onsets and codas)
d. * P/C: C may not associate to peak nodes (nuclei)[2]

[1]We may need to parameterize this constraint when we analyse the heavy syllable cvv in Arabic which obviously has a complex nucleus, and when we discuss cvvc, cvcc and cvvcc which all show at least a violation of *COMPLEX.

e. ONS: Syllables must have onsets
f. -COD: Syllables must not have codas[3]

(Prince and Smolensky, 1993, ch. 6)

In addition to this family of constraints, however, there are the faithfulness constraints: namely PARSE and FILL. This family of constraints aim at restricting the well-formed syllable structures to those that exhibit a one-to-one correspondence of input segments with syllable positions (Prince and Smolensky, 1993). In other words, the perfect syllable structure, from the viewpoint of this family of constraints, is the one that can occupy all and only those segments that appear in the input which means that addition or deletion are not preferred. These faithfulness constraints are as follows:

(5) a. PARSE:
underlying segments must be parsed into syllable structure

b. FILL:
Syllable positions must be filled with underlying segments.

(Prince and Smolensky, 1993, p.85)

Consider the following syllable structure analyses of the input cvc to see how they may satisfy or violate a faithfulness constraint:

(6) a. .cvc.
b. .cv.<c>
c. .cv.c∇ .

While (6) a, where input segments are in one-to-one correspondence with syllable positions, satisfies both faithfulness constraints, (6) b-c do not. In (6) b., the syllable

---

[2]I do not think that it is always the case to have a consonant not occupying a peak position. Dell and Elmedlaoui, when they discussed core syllabification in Berber, have demonstrated that consonants can be nuclei.

[3]McCarthy and Prince (1993) used ONSET and NoCODA for ONS and -COD respectively.

structure suggested could not occupy all segments of the input yielding the underparsing of the final c (represented by putting it between angled brackets < >). On the other hand, (6) c. is a case of over parsing as it may occupy more than the input's segments by virtue of having an empty position (represented by ∇). Phonetically, these two processes, viz., underparsing and overparsing, will be realised as deletion and epenthesis, respectively. And, it should be made clear that we are dealing with the structure not with the input. This means that a structure that contains a nucleus position, for example, whether empty or filled like (6) c., is taken as one that has a nucleus, but an underparsed segment will be treated as totally absent even if it appears in the input, because epenthesis will fill that empty position while underparsed segments will not be phonetically realised by virtue of deletion. So, they are not like their structural counterparts, faithfulness constraints, instead of investigating whether syllables have onsets, lack codas, etc., are interested in the faithfulness of the relation between input segments and syllable positions.

According to Prince and Smolensky (1993), only these two faithfulness constraints along with two structural ones: ONS and -COD, may be relatively ranked in an order of dominance to satisfy language-particular needs while the remaining structural constraints, NUC, *COMPLEX, *M/V, and *P/V are "fixed in superordinate position" (Prince and Smolensky, 1993 p.88). So, our task in the next section is to see how language-particular ranking of the basic syllable structure constraints may lead us from input to optimal phonological output.

2.2.3 Ranking and Language-particular Satisfaction

In chapter one, when we discussed OT, we said that the constraints provided by UG are ranked in a language-particular order of dominance to help us determine the true outputs of a certain language. And, in the previous section, we presented the basic constraints

related to syllable structure. Our objective in this section is to try and understand the techniques of ranking of these constraints.

Before analysing the various methods or different rankings of syllable structure constraints, let us agree on a rather general but very essential point related to low-ranked constraints. It might cross one's mind that low-ranked constraints are not important and can be ignored or simply omitted. Yet, this is far from being correct. Violation of low-ranked constraints can prove to be very decisive in determining the optimal analysis, and their existence is felt when the higher ones fail to do the job. Consider the evaluation of two candidate analyses of the string cvcv in the following tableau:

(7) Two analyses of /cvcv/

cvcv	PARSE	FILL	ONS
→ .cv.cv.			
.cvc.v			*!

Due to the fact that analyses are faithful to their input, they do not violate neither PARSE nor FILL, but ONS, the low-marked constraint, comes to determine the optimal analysis stating a basic principle of the theory of ranking: "when all else is equal a subordinate constraint can emerge decisively" (Prince and Smolensky, 1993 p.86). This principle will be heavily depended on in the coming discussion about constraint ranking.

To see how we may rank syllable structure constraints to serve our language-particular purposes, we must know that there are two angles from which we should view this ranking. First, the relative ranking of structural constraints, on the one hand, and faithfulness ones, on the other, will determine whether onsets are required and/or whether codas are forbidden. Secondly, the relative ranking of faithfulness constraints with respect to each other will reveal how onset requirement and/or coda banning are enforced.

This will not tempt us, however, to designate a separate section for each ranking since they are closely related. What we will divide into two sections is the discussion of onsets and codas. We will do this because the structural constraints ONS and -COD do not directly interact. This means violating one of them does not mean satisfying the other and not vice-versa. So, they ought to be discussed in two distinct sections in which either of them will be confronted with the faithfulness pair (McCarthy and Prince, 1993).

2.2.3.1 Onsets and Faithfulness

Our main aim at this subsection, and the one about codas, that will follow shortly, is to show the effects of the interaction of structural and faithfulness constraints. For reasons, discussed above, we will focus, at the moment, on ONS, PARSE, and FILL. We will basically try to demonstrate that ranking ONS with respect to faithfulness constraints will determine if onsets are required or not, and ranking faithfulness constraints, PARSE and FILL, with respect to each other will show how the onset requirement, if it is required, is enforced. To do so, let us take the simple input /v/ and evaluate a set of three candidate analyses of this input each of which exhibits a single violation of one of the three constraints in question, against different constraint hierarchies showing the possible relative rankings. Consider these analyses:

(8)	a.	.v.	violates	ONS
	b.	<v>	"	PARSE
	c.	.∇v.	"	FILL

Let us take .v. as the optimal analysis. This means that the constraint that is violated by .v. must appear very low in the constraint hierarchy:

(9) Onset Not Required

/v/	FILL	PARSE	ONS
→ .v.			*
<v>		*!	
.∇v.	*!		

(Prince & Smolensky, 1993, p.90)

(The relative ranking of FILL and PARSE is not decisive, and this was indicated by the dotted line separating them).

We conclude from this demonstration that the language that will consider .v. which violates ONS as the optimal analysis, as a result of ranking ONS lower than both faithfulness constraints, is one that does not require onsets because the other two analyses, <v> and .∇v., are not optimal though they both satisfy ONS, either vacuously by having no syllable structure at all in the case of <v>, or by creating an empty position that can be filled by an onset in the case of .∇v. So, this means that if both faithfulness constraints dominate ONS in a language-particular hierarchy of constraints, onsets in that language are optional.

Yet, what would happen if the ranking was done the other way around? I mean what would happen if ONS dominates both or one of the faithfulness constraints? Of course, we will have a different result concerning onset requirement. Consider the following two tableaux taken from Prince and Smolensky 1993:

(10) Onset Required

(a) Enforcement of the Requirement by Underparsing

/v/	ONS	FILL	PARSE
.v.	*!		
→ <v>			*
.∇v.		*!	

(b) Enforcement of the requirement by Overparsing

/v/	PARSE	ONS	FILL
.v.		*!	
<v>	*!		
.∇v.			*

These two tableaux are indicative of two main points. First, both show that onsets are required, in languages that apply by these two rankings, since .v. is never optimal as a result of ranking ONS higher than both (10) a. or one (10) b. of the faithfulness constraints. So, again the relative ranking of ONS, which is a structural constraint, and the faithfulness constraints proved decisive of whether onsets are required or not. The other thing that is revealed to us by the tableaux in (10) above is the relation between the relative ranking of PARSE and FILL and the manner in which onset requirement is enforced. By ranking FILL higher than PARSE, in (10) a., the onset requirement in the optimal output is enforced by deletion. This means that the optimal analysis <v> satisfies ONS by having no structure at all. Also, by ranking PARSE higher than FILL, in (10) b., the onset requirement in the optimal analysis, .∇v., is enforced by epenthesis, i.e., by creating an empty position that can later be filled by an onset to satisfy ONS.

2.2.3.2 Codas and Faithfulness

Like we did with onsets, above, we will try to prove the same claim with codas. We will endeavour to show that the relative ranking of -COD and faithfulness constraints will determine whether codas are banned, and that the relative ranking of FILL and PARSE will show how we may enforce such a banning. This time we will consider different candidate analyses of the input /cvc/ such as:

(11)

a.	.cvc.	violates	-COD
b.	.cv.<c>	"	PARSE
c.	.cv.c∇.	"	FILL

Again, whenever both faithfulness constraints dominate a structural one, the analysis representing the very same input faithfully will be designated the victor, even if such an analysis violates the low-ranked structural constraint. Consider the following tableau:

(12) Coda not Banned

/cvc/	PARSE	FILL	-COD
→.cvc.			*
.cv.<c>	*!		
.cv.c∇.		*!	

Again, and if we pursue the same line of argument, we will reach the inevitable conclusion: when PARSE and FILL dominate -COD in any language-particular ranking of constraints, codas in that language will be considered optional.

Also, we would discover, like we did above with onsets, that if -COD dominates both or either of the pair PARSE, FILL, codas will be forbidden, and this banning will be enforced by either deletion or epenthesis that will be decided by the relative ranking of PARSE and FILL.

(13) Coda Banning

(a) Enforcement of the Banning by Underparsing

/cvc/	FILL	-COD	PARSE
.cvc.		*!	
→.cv.<c>			*
.cv.c∇.	*!		

(b) Enforcement of the Banning by Overparsing

/cvc/	-COD	PARSE	FILL
.cvc.	*!		
.cv.<c>		*!	
→.cv.c∇.			*

Both tableaux in (13) show that codas are forbidden by virtue of ranking -COD higher than both or one of the faithfulness pair, and they also show that this is enforced by deletion of the final c (13) a., or by epenthesis that creates an empty position which can be filled by a vowel to make a new syllable the onset of which is the final c of the input.

To summarise, we say that the process of ranking the basic syllable structure constraints involves two major steps that are decided differently to suit different languages. We, first of all, need to decide whether onsets are required and/or whether codas are forbidden by considering the syllable types of the language for which we would like to set a constraint hierarchy. By this we would rank the structural constraints with respect to their faithfulness counterparts to satisfy our language-particular demands. Then, we will have to decide the way in which requirement of onsets, if they are required, is enforced and the way in which banning of codas, if they are forbidden, is enforced to be able to determine the relative ranking of PARSE and FILL. This is the goal that we will try to achieve in the very next section when we talk about the basic Syllable Structure Constraints Ranking in Taifi Arabic.

2.3 Optimality Theory and the Basic Syllable Structure of Taifi Arabic (T.A)

This variety of Arabic is a dialect spoken in Taif city and the surrounding areas in Hijaz province of Saudi Arabia (For an elaborate discussion of the dialect and a thorough presentation of its phonetic system, consult the appendix!). This dialect will be the object of discussion in what remains of this study. I will try to apply the theoretical framework of OT to analyse some phenomena related to the syllable and the syllable structure. So, in this section, we will start with the foundations on which all future analyses rely. This means that we will discuss the ranking of basic syllable structure constraints in a hierarchy of dominance that will help us determine the core syllable, and consequently the syllable types of this dialect. Yet before getting involved in such a task, it is quite a good idea to know what the syllable types, that we are aiming to generate within OT are. So, in what remains of this chapter, we will first review the syllable types of Arabic in general and TA in particular. Then, we will start the discussion of constraint ranking that will be applicable to the syllable types in question.

2.3.1 Syllable Types in TA

All the scholars who have studied the phonology of Arabic, either classical Arabic or various dialects like Al-ani (1970), Al-ani and May (1978), McCarthy (1979a, 1979b), Selkirk (1981), Abu-salim (1982), Jarrah (1993) and many others, have a unanimous agreement that at least the underlying inventory of syllable types of Arabic is as follows:

(14)						
	light	a.	cv	in words like	.ki.tab.	`a book'
	heavy	b.	cvc	"	.ba.Har	`a sea'
		c.	cvv	"	.saa.9ah	`an hour'
	super-heavy	d.	cvvc	"	.faa.nuus.	`a lantern'
		e.	cvcc	"	.bint.	`a girl'
	extra-heavy	f.	cvvcc	"	Haadd	`sharp'

All the syllable types appear phonetically in TA. I chose to emphasise this point because it was argued by Abu-salim (1982) that the syllable type cvvcc is not phonetically realised, and that it is changed to other syllable types by nucleus shortening, for example, to surface as cvcc. Jarrah, however, disagrees with Abu-salim and says that this syllable type is phonetically realised, though there are some techniques to avoid it that are employed in some but not all cases. We will take the latter viewpoint and regard this syllable type as a phonetic entity (cf. Abu-salim 1982 and Jarrah 1993 for some fruitful discussion about this issue).

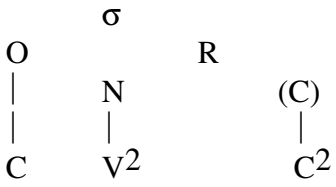
If we look at the list of syllable types in (14) above, which by the way is in itself a representation of the markedness scale of those syllables with cv (14) a. as the least marked and cvvcc (14) f. as the most highly marked as far as distribution is concerned, we will notice some properties that may help us in the discussion of constraint ranking. Firstly, all the syllables begin with a single consonant which means that we do not have onsetless syllables, and that we do not have consonant clusters, in the onset position. Secondly, only some of the syllables in question are open while others have codas that can either be simple or complex consisting of two consonants; (14 a-c) are open syllables where the others are not. Another thing that we notice about this list of syllable types is related to the nuclei. The nuclei can be simple as in (14 a,b,e) or complex as in (14, c,d,f). We can put it differently by saying that the nuclei can either be short or long to establish the fact that the complexity is in the sense of having one vowel occupying two timing slots not two vowels, due to the fact that this dialect, as many modern Arabic dialects, lack diphthongs. So, from these observations, we can conclude that onsets are always present, codas are not always represented and complexity i.e., the existence of more than a c or a v under a syllable position node, is only absent in onsets.

So, after considering the list of syllable types, can we have a core syllable that captures all these properties? This means that on which of the four core syllable types, discussed above in section 2, may we decide to represent these properties of T.A syllable types? To answer this, we say, employing a technique presented by Clements and Keyser (1983), that the primary core syllable of T.A can be represented as in (15) below:

(15) $cv^2(c^2)$

This representation says that we have an onset, up to two segments in the nucleus and optionally up to two segments in the coda. This core syllable will represent all the six syllable types of TA that all have the following internal structure[4].

(16)



O = Onset
R = Rhyme
N = Nucleus
C = Coda

2.3.2 Onsets and Codas

Before we start using the above mentioned properties of the primary core syllable of TA to determine the ranking of the syllable structure constraints that will generate our syllable types, let us discuss the issue of complexity exhibited by some syllable position nodes, nuclei and codas in particular. This is because when presenting the basic syllable structure constraints, and especially the structural ones, we saw that there were some constraints that are ranked in superordinate position with respect to the others. One of

[4] It has been poseted by many researchers that the core syllable in such cases is cv(x), where x stands for either a vowel or a consonant, treating the superheavy and the extraheavy syllables using other techniques. Fot the purposes of this study, however, the main point is to capture the optionality of the Coda which is obvious using either of the representations of the core syllable.

these constraints is *COMPLEX, which prohibits associations of more than one C or V to any syllable position node, in this respect, to any onset, nucleus or coda. This particular constraint is obviously violated by four out of the six syllable types in question, which makes it by no means an exception. When we discussed this particular constraint above, we mentioned the need for some kind of parameterisation to suit our syllable types. So, how may we parameterise *COMPLEX? We can do this by restricting the application of this constraint to onsets only. This claim is supported by the fact, that is quite obvious as one goes through the list in (14) above, that the onset is the only syllable position node that never permits the association of more than one segment, i.e., only one C. On the other hand, we do have syllables with complex nuclei (cvv, cvvc), with complex codas (cvcc), and with complex nuclei and codas (cvvcc). However, for the time being, we may not need such a parameterization, yet we will surely employ it in the next chapter, when we talk about epenthesis and syncope.

Now, let us turn to the heart issue of ranking the basic syllable structure constraint, ONS, -COD, PARSE and FILL, in a dominance order suitable for TA. As expected, we will need to decide two points. First, are the onsets of TA required and/or the codas forbidden? Secondly, if either are, how is that enforced? The answers to these two questions will tell us how to rank structural constraints with respect to their faithfulness counterparts and will tell us how to rank the faithfulness constraints, FILL and PARSE, with respect to each other.

Let us start by looking at the first question that is related to the onsets and codas. All the syllable types in (14) above have onsets, which means that onsets are not optional; they are required in TA and Arabic in general. This means that ONS must dominate both, or at least one, of the faithfulness pair. This relation of dominance is represented as follows:

(17) ONS >> PARSE, FILL
(`>>' means dominance and `,' no relative dominance)

On the other hand, only two out of the six syllable types of TA are open hinting at the fact that codas are optional, though not forbidden. This, and in accordance with what has been discussed in section (2) above, will result in ranking -COD lower than both faithfulness constraints.

(18) PARSE, FILL >> -COD

So, from (17) and (18) above, we can say that the relative order of the structural constraints, on the one hand, and the faithfulness ones, on the other, in TA is represented as in (19) below:

(19) ONS >> PARSE, FILL >> -COD

Now. let's examine this partially complete ranking to see whether it succeeds in determining the true output if it were to evaluate particular inputs. We will take, as our first example, the input cvc of which we will evaluate the best candidate analyses in the following tableau:

(20)

/cvc/	ONS	PARSE	FILL	-COD
→.cvc.				*
.cv.<c>		*!		
.cv.c∇.			*!	

Obviously, the postulated ranking of the constraints is successful in determining the true output of the input cvc. Yet, what if this particular ranking, in which the relative order of dominance of PARSE and FILL is not specified, is confronted with the input v of which it must choose the right output. Will this ranking be able to do the job now? Consider the following tableau:

(21)

/v/	ONS	FILL	PARSE	-COD
.v.	*!			
.∇v.		*		
<v>			*	

Apparently, we may not determine the true output of the input v depending on this partially complete ranking. We can not choose, as the optimal analysis, between .∇v. and <v> due to the fact that PARSE and FILL are not relatively ranked.

From the examples discussed above, we feel an urgent need for deciding on a relative order of dominance between PARSE and FILL. This dominance will help us choose either .∇v. or <v> as the optimal analysis. This is done by ranking the constraints violated by our desired analysis lower. So, if our true output, in (21) above, is .∇v., FILL must be ranked lower than PARSE. Now comes the time to answer the real question: Is it .∇v. or <v> that stands as the optimal output of /v/? In other words, we are asking about the way in which onset requirement in TA is enforced, which means that we want to know how ONS is satisfied: Is it satisfied vacuously by having no structure at all, or is it satisfied by creating an empty position that can be filled by a potential onset?

There is an overwhelming amount of evidence demonstrating that the onset requirement in Arabic, in general, is enforced by epenthesis. If we go through the cases that call for either epenthesis or syncope to solve a problem like having superheavy syllables at a non-final position, violation of the sonority sequencing generalisation, having three or five consonants clustering medially, and the like, which we will look into with more depth in the next chapter, we shall see that it is almost always that epenthesis takes precedence over syncope, demonstrating the need for ranking FILL, that is violated by epenthesis, lower than PARSE, which is violated by syncopating. Nevertheless, the most striking

evidence for the enforcement of onset requirement by epenthesis which results from ranking PARSE higher than FILL comes from the fact that a consonant like the glottal stop in words like /?in katab/, `it was written', is an epenthetic one inserted to avoid onsetless, or vowel-initial syllables. The seventh binyan, out of the fifteen binyanim of the triliteral verbs of Arabic (cf. McCarthy 1979a, 1981), is an example of a binyan having a cluster of two consonants in the onset of its penultimate syllable, /nkabat/. This consonant clustering is not allowed by the language. So, as a result, a vowel is usually epenthesized before that onset cluster. This epenthesis will break up this clustering as it will take the initial consonant, /-n-/ in our example, as its coda leaving the second consonant, /-k-/, occupying the onset of the following syllable[5]. This process will yield an onsetless antepenultimate syllable, violating a well-established constraint in Arabic, ONS. To avoid such a fatal violation, we either delete the initial vowel that we epenthesized and simply go back to where we have started or to epenthesize a consonant that will serve as the onset and satisfy ONS. Obviously, the second choice is better, and consequently we may say that it is epenthesis that does enforce onset requirement in Arabic.

According to what we have said above, the relative ranking of the faithfulness pair should show PARSE higher than FILL, which means that the overall ranking of the basic syllable structure constraints for TA is as follows:

(22) ONS, PARSE >> FILL >> -COD

This shows that the relative ranking of ONS and PARSE is not specified due to the fact it is enough for ONS to dominate a faithfulness constraint, FILL in our case, to convey

[5] One may argue for the epenthesis of such a vowel between the two consonants in the onset to serve the same purpose. Yet, this is not what usually happens in reality. And, to ensure that the vowel is epenthesized where we want it to, i.e., initially, an appropriate constraint, that will be discussed in chapter 3, is employed.

onset requirement. Now, let us try this ranking of the constraints to see whether we can discriminate between .∇v. and <v> in (21) above. Consider the following tableau:

(23)

/v/	ONS	PARSE	FILL	-COD
.v.	*!			
<v>		*!		
→.∇v.			*	

So, determining the relative ranking of PARSE and FILL results in distinguishing .∇v. as the optimal analysis.

In conclusion, we say that the framework of O.T. showed high potentiality to serve as an analytical power of grammar after what we have experienced with basic syllabification. I do not want to appear as an advocate of the perfectness of this theory, however. This is because when we tackle more complicated issues, in the next chapter, we will find ourselves, sometimes, not able to describe a certain phenomenon or, at best, suggesting an *ad-hoc* solution.