2020

PMKNS for PIE: Parsed Morphological KATR Networks of Sanskrit for Proto-Indo-European

Ryan Mark McDonald
University of Kentucky, rmc257@uky.edu
Digital Object Identifier: https://doi.org/10.13023/etd.2020.498

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Ryan Mark McDonald, Student

Dr. Andrew M. Byrd, Major Professor

Dr. Allison Burkette, Director of Graduate Studies

PMKNS for PIE: Parsed Morphological KATR Networks of Sanskrit for Proto-Indo-European

THESIS

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Arts in Linguistic Theory and Typology in the College of Arts and Sciences at the University of Kentucky

By
Ryan Mark McDonald
Lexington, Kentucky

Director: Dr. Andrew M. Byrd, Professor of Linguistics
Lexington, Kentucky

2020

ABSTRACT OF THESIS

PMKNS for PIE: Parsed Morphological KATR
Networks of Sanskrit for Proto-Indo-European

In this thesis, I construct two computational networks for Sanskrit to test theories of nominal accentuation as a way of examining the simplicity of each theory. I will be examining the Paradigmatic Approach and the Compositional Approach to nominal accentuation. For the Paradigmatic Approach, nominals are categorized into mobile and static categories based on how the accent appears in the paradigm [For10]. For the Compositional Approach, accent mobility is a result of the combination of morphemes and their inherent accent states [Kip10]. To construct these networks, I use the KATR extension to the DATR language for lexical knowledge representation [Fin+02].

In Chapter 1, I give an overview of Proto-Indo-European (PIE) accentuation and KATR. Chapter 2 presents my methods and connects the hypothetical nature of PIE to the well-documented Indo-European (IE) language Sanskrit. In Chapters 3 and 4, I use a guided derivation of a Sanskrit r-stem nominal *pitr̥-* and a Sanskrit a-stem nominal *sukha-* to walk us through each step. Chapter 5 is an analysis of my results for the two networks from Chapters 3 and 4 and then the overall conclusions I have drawn from the project and suggests further areas of expansion.

KEYWORDS: KATR, Proto-Indo-European, Sanskrit, Network Morphology, nominal morphology, morphophonology

Ryan Mark McDonald

December 3, 2020

PMKNS for PIE: Parsed Morphological KATR Networks of Sanskrit for Proto-Indo-European

By
Ryan Mark McDonald

Andrew M. Byrd
Director of Thesis

Allison Burkette
Director of Graduate Studies

December 3, 2020

Dedicated to my parents.

ACKNOWLEDGMENTS

I want to thank all my friends and family for they have been an everlasting source of support. Without them, I would never have been able to complete this thesis. My family pushed me to complete my work and my friends provided emotional support when I needed it most. I want to especially thank Avery Bagget, Taha Husain, Aleah Combs, Steven Gerencser, and Kyler Laycock for their friendship and support during our shared time at the University of Kentucky. I also want to thank Samuel Cook for everything he does for me and Shawn Hyde for constantly checking on me during the final stages of the thesis writing.

I want to give special thanks to Phillip Barnett. He provided the title for my thesis as well as provided the initial instruction for one of the core components of the project. Without his help, it would have been a more laborious experience.

I want to thank my committee, Andrew Byrd, Mark Lauersdorf, and Raphael Finkel, for their numerous comments. I have learned a great deal about how to express myself more effectively. I want to thank Andrew Byrd for giving me the idea for the thesis and guiding my studies during my time at the University of Kentucky. I will also consider him to my mentor. Mark Lauersdorf provided many comments that helped me see my writing from new angles and challenge my thinking in helpful ways. Raphael Finkel was crucial in the project for my thesis. Without him, I would not have been able to understand the computational aspects enough to complete the project. He also provided crucial technical and formal stylistics help.

TABLE OF CONTENTS

LIST OF TABLES

Chapter 1 Introduction

The field of Indo-European Studies has two competing theories on the accentuation process for Proto-Indo-European (PIE). The first theory describes the process as a product of the paradigm, where accent is assigned based on the position of the word in the paradigm [Sch72]; [For10]; [Cla07]. The second theory describes the process as a result of the combination of accented, unaccented, and pre-accenting morphemes [Kip10]; [LY18]. The purpose of this thesis is to computationally model the two competing theories, focusing on nominals, using Sanskrit as a test language. I modeled the theories using an extension of the DATR language called KATR, developed at the University of Kentucky by Gregory Stump and Raphael Finkel [Fin+02].

My main research questions are which theory of PIE accentuation is more elegant to code using KATR and what using KATR can tell us about the two theories. My definition of elegance is the fewest number lines needed to generate a full paradigm of a network, leaning on the idea of code coverage [Eri17].

Although a basic understanding of PIE phonology, PIE morphology, and DATR/ KATR and Network Morphology is beneficial for the reader, I plan to give a basic outline of each. My goal is to provide enough information that anyone should be able to follow my analysis without needing to dive deeply into the complex fields of PIE Studies and Network Morphology.

1.1 The PIE Nominal

In PIE phonology, every nominal is reconstructed with only a single accent, and the location of the accent affects the shape of the remaining syllables. For PIE the accented vowel is reconstructed as **é*[1], and this vowel is typically the only **e* in a PIE nominal. This process is known as **ablaut**, the change of an inherent vowel's quality (e.g. *e* to *o*) or quantity (e.g. *e* to *ē*) as a part of the languages morphology. The state of an inherent vowel is described as its ablaut **grade**. Table 1.1 exemplifies the various grades reconstructed for PIE [For10]; [Cla07]; [MA06].

Table 1.1: Ablaut grades with *sed- 'sit'. Capital C refers to any consonant. NE refers to New English or Modern English, and OE refers to Old English. Table is a recreation of the table in Mallory and Adams 2006:68.

Alaut Grade	**Shape**	**Example**
Full-Grade/E-Grade	*CeC	**sed-ye/o-* >NE. *sit*
O-Grade	*CoC	**-sódos* >OE. *gesæt* 'act of sitting'
Zero-Grade	*CC	**ni-sd-os* >NE. *nest*
Lengthened E-Grade	*CēC	**sēdeh*$_a$- >OE. *sǣt* 'lurking-place'
Lengthened O-Grade	*CōC	**sōdos* >NE. *soot*

[1] An asterisk marks a form as reconstructed.

The default grade is the **full-grade**. The vowel has the quality and quantity **e*, seen in the table with **sed-ye/o-* becoming English 'sit'. The **o-grade** is when the vowel has its quality changed while having the same quantity, surfacing in the shape **o*. Table 1.1 shows how the change of the vowel in **sed-* from the full-grade to the o-grade results in a different form semantically related word (i.e. the verb 'sit' to the nominal 'sitting'). The **zero-grade** is when the ablauting vowel is absent completely, exemplified by the word **ni-sd-os* 'nest' where **sed-* has become **-sd-*. The two remaining grades, the **lengthened e-grade** and the **lengthened o-grade**, have the same quantity as the full-grade and the o-grade respectively but the quantity of the vowels have changed from short to long. As with the other grades, this change results in the formation of differently shaped but semantically similar words[2].

These grades are distinct and morphologically salient in PIE; however, in Sanskrit a merger caused these the full-grade and o-grade to lose their distinction, resulting in only the full-grade surviving with the shape *a*, except in instances where it lengthens to *ā* as a result of Brugmann's Law [For10]; [Cla07].

PIE nominals contain three major components[3]: the **root**, the morpheme with the core lexical meaning; the **suffix**, various derivational morphemes; and the **ending** or **desinence**[4], the case-number inflectional morphemes. The full grade vowel can be present in any of these components, which means that any of the components can host the nominal's accent. When no suffix appears, these nominals are **root nominals** [For10]; [Cla07].

An inflectional stem refers to the root and any derivational suffixes. The boundary of the stem is immediately preceding the endings. When discussing the stem, we reference the strength of the stem. Strength refers to the ablaut grade of the inherent vowel. A strong stem has the ablauting vowel in the full grade while the weak stem as the ablauting vowel in the zero grade; the full grade vowel will appear in the ending when the stem is weak. One major exception is the **static** categories as the vowel never reaches the zero grade in the weak cases and instead the strong case has a different starting grade[5]. Stem strength is directly connected to case-number assignment in PIE and Sanskrit where the nominative and vocative of all numbers, the accusative in the singular and dual, and occasionally the locative singular[6] are all classified as strong case-number pairings as they exhibit the strong stem. All the other case-number pairings use the weak stem [For10]; [Cla07].

A visual representation of the concepts of ablaut and stem strength is helpful. The following table shows the full paradigm of *pitṛ-*, the Sanskrit word for 'father'. The different cells reflect ablaut and stem strength. A grey cell uses the strong stem and a white cell uses the weak stem. When the ablauting vowel is present, it is in

[2]Mallory and Adams give that 'soot' is 'what settles,' showing the initial semantic connection to **sed-* [MA06].

[3]The names of the components are traditional to PIE Studies [For10].

[4]Both terms can be found in the literature; however, I prefer ending as the case-number morphemes appear at the end of a nominal.

[5]Commonly static nominals have the o-grade in the strong stems with the full-grade in the weak stems (e.g. **dṓm, *déms*, but nominals with the lengthened e-grade exist as well

[6]Not every nominal uses the strong stem when constructing the locative singular, but some do.

bold. This table shows the interconnected nature of ablaut and stem strength.

Table 1.2: A full paradigm of *pitr̥-* with ablaut and stem strength marked.

	Singular	**Dual**	**Plural**
Nominative	*pit**ā́***	*pit**á**rau*	*pit**á**raḥ*
Vocative	*pít**a**ḥ*	*pít**a**rau*	*pít**a**raḥ*
Accusative	*pit**á**ram*	*pit**á**rau*	*pitr̥̄́n*
Instrumental	*pitrā́*	*pitŕ̥bhyām*	*pitŕ̥bhiḥ*
Dative	*pitré*	*pitŕ̥bhyām*	*pitŕ̥bhyaḥ*
Ablative	*pitúḥ*	*pitŕ̥bhyām*	*pitŕ̥bhyaḥ*
Genitive	*pitúḥ*	*pitróḥ*	*pitr̥̄ṇā́m*
Locative	*pit**á**ri*	*pitróḥ*	*pitŕ̥ṣu*

Looking at Table 1.2, we can see how in the strong cases the inherent vowel is present in the full-grade, but in the weak cases the vowel is in the zero-grade. This pattern shows the interconnected nature of ablaut and stem strength. Accent location is also a connected concept to ablaut. Table 1.2 also shows this as the accent is placed on the stem vowel when it is in the full grade but on the ending when the stem is in the zero grade and the ending begins with a vowel, showing an mobile accent pattern, which I explain in the following sections.

Now that we have examined how a PIE nominal is constructed, we can begin our dive into the two competing theories of PIE accent: the Paradigmatic Theory and the Compositional Theory.

The Paradigmatic Theory

In the paradigmatic theory of PIE accentuation, athematic nominals are traditionally separated into at least four different categories based on how the accent alternates throughout the paradigm or does not alternate in the case of static categories. Experts who want to continue the paradigmatic description have created additional categories to explain exceptions to the four traditional categories [MFM03]. The traditional four categories are acrostatic, proterokinetic, hysterokinetic, and amphikinetic. In acrostatic nominals, the accent is **static**, meaning that it remains in the same position throughout the paradigm, thus nók^{w}ts (nominative singular), nék^{w}ts (genitive singular). In the other three nominal classes the accent is **mobile**, which means that the accent "moves" throughout the paradigm, depending on the strength of the case/number[7]. Before we begin the description of each of the categories, Table 1.3 helps to illustrate the connection between ablaut, stem strength, and accent mobility.

[7] Mobility refers to how an accent may appear elsewhere in the word depending on various morphological conditions; here mobility relates to case-number assignment.

Table 1.3: The strong case is the nominative singular and the weak case is the genitive singular. The form of the words are before certain phonological laws have been applied, such as Szemerényi's Law. This table is found in Fortson 2004.

	acrostatic				**proterokinetic**		
	R	**S**	**E**		**R**	**S**	**E**
strong	**nók^w*	*t*	*s*		**mén*	*ti*	*s*
weak	**nék^w*	*t*	*s*		**mn̥*	*téi*	*s*
	'night'				'thought'		

	hysterokinetic				**amphikinetic**		
	R	**S**	**E**		**R**	**S**	**E**
strong	**ph_2*	*tér*	*s*		**$h_2$éu̯s*	*os*	*s*
weak	**ph_2*	*tr*	*és*		**$h_2$us*	*s*	*és*
	'father'				'dawn'		

Acrostatic

The acrostatic (AS) category is the only static category of the four traditional paradigmatic categories. For AS nominals, the location of the accent always surfaces on the root. Even if a vowel or syllable nucleus that could hold the accent appears later in the word, the accent will still remain on the root. This is what it means to be **static**. An example of this category is the PIE word for 'night' **nók^{w}ts*, **nék^{w}ts*[8]. We can see that the strong form contains /ó/ in the root while the weak form contains /é/; however, the accented vowel never ablauts to the zero grade in the root. The acro- part of the category name shows that the accent is "at the top", meaning that it appears on the root; thus, the accent is always realized on the root [For10]; [Cla07].

The AS category is broken into sub-types. The example above belongs to the normal subtype where the root ablauts between the o-grade and the full grade[9]. A second subtype is the lengthened type. For these AS nominals, the expected o-grade is realized instead with the lengthened e-grade in the strong forms and the expected full-grade in the weak forms. An example of this type of AS is the nominal for 'king' **h_1rḗḱs*, **$h_1$réḱs*. As can be seen, the nominative singular has the lengthened e-grade (*ḗ*) of the root vowel and the full-grade (*é*) in the genitive singular.

The final type of static athematic nominals are the root AS nominals. The previous types contain all three elements of a PIE nominal (the root, the suffix, and the ending); this type lacks the middle element. Root AS nominals have the endings attached directly to the root. An example of this type is the PIE word for 'cow' **g^{w}ṓu̯s*, **g^{w}éu̯s*[10]. Here the root **g^{w}óu̯-* has the case endings attached directly without any intervening suffixes. This category is the first of the traditional categories focused on by this thesis. While this category describes athematic categories and the

[8] In this thesis, I cite all nominals with their nominative singular forms followed by genitive singular forms.

[9] As a reminder, full grade indicates that the inherent vowel is present and in the shape /e/ while o-grade indicates the inherent vowel in PIE has ablauted from /e/ to /o/.

static nominal in the model is a thematic nominal, it can serves as an acceptable replacement as thematic nominals and acrostatic athematic nominals have a static accent.

Proterokinetic

The proterokinetic (PK) category is the first of the three traditional mobile categories, where the accent depends on the word's location in the paradigm. For PK nominals, the strong cases show root accentuation whereas the weak cases show suffix accentuation. This alternation of the accent location between the strong and weak cases is what separates the static categories (e.g. the acrostatic) from the mobile categories. Examples of this category can be seen in the neuter PIE word for 'fire' **péh$_2$u̥r̥*, **ph$_2$u̥éns* or the PIE word for bird **h$_2$éu̥is*, **h$_2$u̥éi̥s*. As shown in Table 1.3, the root is in the full grade while the suffix and ending are in the zero grade for the strong form, but the accent now appears on the suffix in the weak cases, ablauting the root to the zero grade, the suffix to the full grade and the ending remaining in the zero grade [For10]; [Cla07].

Hysterokinetic

The hysterokinetic (HK) category is the second of the traditional mobile categories. For HK nominals, the accent begins on the suffix in the strong cases whereas appearing on the ending in the weak cases. An example of this category includes the PIE word for 'father' **ph$_2$tḗr*, **ph$_2$trés*. As shown in Table 1.3, the suffix is in the full-grade whereas the root and ending are in the zero-grade for the strong cases, but the accent appears on the ending in the weak cases, resulting in the suffix ablauting to the zero-grade, the suffix to the full-grade, and the root remaining in the zero-grade. This category is the second of the two traditional categories focused on by this thesis [For10]; [Cla07].

Amphikinetic

The final mobile category consists of the amphikinetic (AK) nominals. For the strong cases, these nominals have the full-grade in the root, the o-grade in the suffix, and the zero-grade in the ending. This pattern ablauts to having the zero grade in the root and the suffix with full grade in the ending for the weak cases. This category is exemplified in the PIE word for 'dawn' **h$_2$áu̥sōs*, **h$_2$usés*. As shown in Table 1.3, the root is in the full grade and the suffix is in the o-grade, but the ending is in the zero grade for the strong cases. The accent appears on the ending in the weak cases, causing the root and suffix to ablaut to the zero grade and the ending to ablaut to the full grade [For10]; [Cla07].

[10] Manfred Mayrhofer states that the Sanskrit derivative shows the acrostatic categorization with the nominative singular being *gáuṣ*, the genitive and ablative singular being *góṣ*, and the dative singular being *gáve* [May86].

The Compositional Theory

In contrast to the Paradigmatic Theory explaining athematic accent assignment, the Compositional Theory [Kip10]; [LY18] combines various components in the phonology resulting in the final, surface forms. Morphemes in PIE have one of three inherent accent states: accented, unaccented, and pre-accenting. It is the combination of these states and various phonological rules that determines the surface accent. Every nominal has one stem throughout the entire paradigm, excluding suppletive forms (e.g. the heteroclites where the strong cases have -*r* in the suffix and the weak cases have -*n*, such as the word for 'fire' from above **péh*$_2$*u̯r̥*, **ph*$_2$*u̯éns*).

Morpheme States

As mentioned above, each morpheme in PIE can have one of three states. Some morpheme types have predictable states within the Compositional Theory because they are always in a particular state. For example, the endings may be in one of two states: accented or unaccented. The strong cases have inherently unaccented forms (e.g. the nominative singular *-*s*) whereas the weak cases have inherently accented forms (e.g. the genitive singular *-*és*). The derivational s-stem suffix -*e/os*- is always pre-accenting, meaning that this morpheme never carries the surface accent. This morpheme instead tries to force the accent to surface on the morpheme directly preceding it, exemplified in the word for 'mind' **ménos*, **méneses* where the root **men*- is inherently unaccented but receives its accent from the suffix -*e/os*. According to Lundquist and Yates, **men*- must be inherently unaccented due to the nature of verbal roots in PIE [LY18].

Phonological Rules

The combination of inherent accents is not enough to explain the reconstructed surface accent. It is possible for the combination to yield words with more than one accent or even no accent at all. In PIE, all free nominals must have one and only one surface accent. Only clitics are allowed to be unaccented in their surface realization because they require the use of an adjacent word's accent. To resolve this dilemma, several phonological rules are required. Below are the rules necessary for accent assignment as well as rules that affect the surface accent.

The Basic Accentuation Principle

The Basic Accentuation Principle (BAP) first proposed by Paul Kiparsky for Sanskrit [Kip84] and then for PIE [Kip10] is the most important phonological rule for the Compositional Theory. Unlike many phonological rules, the BAP is actually two complementary rules. The importance of the BAP is that it solves two problems that could arise with the combination of inherent accents; the BAP forces all non-clitic[11] PIE words to have only a single surface accent.

[11] A clitic is word that requires another word for its accent. When a clitic uses the accent of the preceding word, it is called a enclitic. When it uses the accent of a following word, it is called an

The first component to the BAP is that the leftmost syllable of an unaccented word is assigned an accent, explaining how unaccented roots and suffixes combined with the strong endings to receive their surface accent. To illustrate this concept, the root nominal **ḱu̥on-* has an unaccented root and lacks a suffix element. When it is then combined with any of the strong endings (e.g. the nominative plural *-es*), this combination would yield an unaccented, non-clitic nominal ^{x}*ḱu̥ones*. As non-clitic nominals must be accented, the BAP solves the problem. Here an accent would be assigned to the *o* in ^{x}*ḱu̥ones*[12] creating the correct **ḱu̥ónes*.

The second component to the BAP is that all except the leftmost accent is deleted when more than one accent is present, explaining how accented roots and suffixes combined with the weak endings to receive only a single accent. To illustrate this concept, let us look at the nominal *$nók^wt$-, which is inherently accented. If this nominal stem were combined with the genitive singular ending without the second component of the BAP applying, the output would be ^{x}$nék^wtés$. This double accent necessitates the second component. When it is successfully applied, the correct output form surfaces *$nék^wts$[13].

The Oxytone Rule

This rule was proposed by Kiparsky [Kip10] to resolve problems not addressed by the BAP. Although the BAP does prevent unwanted, unaccented nominals from arising, it does not help, however, with resolving accents that have been reconstructed on inherently unaccented morphemes. An example can be shown with the word for 'father'. In most scenarios, the BAP is enough to realize the reconstructed surface form. The stem *ph_2ter- is unaccented and with the strong endings and the weak endings that begin with a resonant, the BAP correctly outputs the reconstructed form: *ph_2ter- + nominative plural *-es* = *$ph_2téres$ and *ph_2ter- + genitive singular *-és* = *$ph_2trés$. However, a problem occurs with the weak endings that begin with an obstruent. The BAP would allow the stem to remain unaccented and the ending to be accented: *ph_2ter- + *-sú* = ^{x}ph_2*tr̥sú*. The Oxytone Rule solves this problem by assigning an accent to the rightmost syllable of an unaccented, polysyllabic, inflectional stem. As the stem *ph_2ter- is unaccented, the Oxytone Rule creates a new stem: *$ph_2tér$-. With the obstruent-initial weak ending, the formulation would now be this: *$ph_2tér$- + *-sú* = *ph_2*tŕ̥su* (The alternation of *$ph_2tér$- to *ph_2*tŕ̥-* is due to a new rule, addressed in the next section).

The Zero Grade Rule

The Zero Grade rule posited by Kiparsky [Kip10] is a process where any *e/o* is lost preceding an accented syllable. This rule deletes both accented and unaccented *e/o*

proclitic [SL12]. An example of an enclitic would be Latin *-que* in the phrase *senatus populus**que** Romana*. An example of a proclitic would be French contractions like the *s'* in *s'habiller* 'to get dressed'.

[12] The character ^{x} marks an ungrammatical form.

[13] The alternation from o-grade to the full grade is one of the ablaut patterns hypothesized for PIE [Byr18]; [For10]; [Cla07].

and is capable of applying targeting e/o with a syllable in between it and the accented syllable, as long as no intervening syllable contains a high vowel. This is the reason that the genitive singular of 'father' is *ph$_2$trés and the locative plural is *ph$_2$tŕ̥su. Here the e in the stem *ph$_2$tér- precedes the accented syllables -és and -sú, resulting in the loss of the vowel in the suffix.

The Vocative Rule

The Vocative Rule assigns an accent to the leftmost syllable of a PIE nominal inflected for the vocative case. This rule explains why the surface accent for some PIE nominals is on the leftmost syllable when the BAP, Oxytone Rule, and the natural combination of inherent accents should cause the surface accent to appear on another syllable, as seen with the vocative singular of the PIE word *pə́h$_2$ter. I have chosen to write out the schwa in this instance to give the accent a location to rest[14]).

Without the Vocative Rule, the Oxytone Rule places an accent on the e, creating *ph$_2$tér. When the BAP applies, no other accent is present, so that is where the accent surfaces. With the Vocative Rule, the next form would be *pə́h$_2$ter. The BAP would then apply and create the expected output *pə́h$_2$ter.

Order of Phonological Rules

Moving from a description of the individual phonological rules of the Compositional Theory leads us to how these rules are ordered in respect to each other. I imply above that the order of the rules is important for the correct output to surface. The following table shows a theoretical derivation from Kiparsky [Kip10], modified to include the Vocative Rule.

Table 1.4: Order of rules from Kiparsky 2010 including the Vocative Rule from Byrd 2018.

	Voc. Sg.	**Acc. Sg.**	**Instr. Sg.**	**Loc. Pl.**
Inflection	ph$_2$ter-∅	ph$_2$ter-m̥	ph$_2$ter-éh$_1$	ph$_2$ter-sú
Vocative Rule	pə́h$_2$ter-∅	—	—	—
Oxytone Rule	—	ph$_2$tér-m̥	ph$_2$tér-éh$_1$	ph$_2$tér-sú
Zero Grade Rule	—	—	ph$_2$tr-éh$_1$	ph$_2$tŕ̥-sú
BAP	—	—	—	ph$_2$tŕ̥-su
Sanskrit	pítaḥ	pitáram	pitr-ā́	pitŕ̥-ṣu

Looking at Table 1.4, we can see that after inflection, the Vocative and Oxytone Rules apply. The BAP must be last, or it would not properly affect the various forms of a nominal. Some forms would have multiple accents (e.g. the locative plural) while others would be lacking an accent all together. This is because the Vocative Rule only applies to the vocative case and the Oxytone Rule only applies to polysyllabic inflectional stems. It is possible for the order of the Vocative and Oxytone Rules to be

[14]See Byrd 2018 for more information.

flipped without any major differences. The biggest difference would be the vocative would have two accents, but this would be repaired by the BAP as with the locative plural in Table 1.4.

1.2 Computational Methods

In the past, historical linguistics relied heavily on philology and written documents [Cam13]. It is only now in recent years that we have had the technological ability to digitized large amount of texts, including ancient manuscripts, with enough care as to avoid damaging the texts [Uni19]. This digitization has drastically increased the tools available to researchers. There are benefits to complementing traditional methods with computational ones. The biggest benefit would be a computational examination of the theories posited through traditional methods. I want to stress that I do not believe the traditional methods should be abandoned; for some of the greatest work in historical linguistics not only comes from traditional philology but also predates computers. I believe that traditional philology is enhanced by computational testing. It is through these tests that our theories can be expanded and refined to reach the leanest theory for PIE.

DATR and KATR

This thesis uses an extension of the DATR language, which is a language for "lexical knowledge representation", meaning that it attempts to represent ways for generating the various forms of a particular language in a network of nodes and the various paths connecting these nodes [EG96]. DATR allows researchers to build theories based on the theory of Network Morphology (see the following section for a brief description and the following chapters for a more specific description) [EG96].

DATR encodes the nodes as labeled sections in the code and dictates the paths the code is supposed to take between the various nodes for generating the desired forms. These paths are what establish the network. A nominal's lexical entry may indicate the stem, gender, declension class, and a particular path needed for generating the final output forms. In another network with different goals, the lexical entry may instead be filled with various paths to other nodes for establishing various derivation processes, such as denominal verbs from a nominal lexeme.

For the purposes of this thesis, the biggest limitation of DATR is the way it handles phonological information. DATR only allows for local phonological modifications, meaning that any modification must involve sequential phonological segments. This is a problem as not all phonological processes require adjacency to occur. Improving the phonological capabilities of DATR was one of the goals KATR addresses. KATR is an extension to the DATR language developed at the University of Kentucky by Raphael Finkel, Lei Shen, Gregory Stump, and Suresh Thesayi [Fin+02]. KATR adds the capability to process nonlocal phonological modifications [Fin+02]. For more specific explanations for this thesis, see the following chapters.

KATR Syntax

This section provides a basic overview of KATR syntax for the rules in the following chapters. A KATR rule is broken into two sides: the left-hand side (LHS) and the right-hand side (RHS). The LHS contains the morphological information in the form of a set or path, and the RHS contains the data for that set or path. Let's look at some examples, which illustrate KATR syntax.

```
1 #vars $ruki: i í ī́ u ú ū ū́ r r̥ ŕ̥ k .[15]
2 #sandhi $ruki s => $1 ṣ .
3 R-Nouns:
4     $strong == "<stem1>" Endings
5     <> == Base-Nouns2
6 A-Neuter:
7     {nom sg} == A-Masculine:<acc sg>
8 FATHER:
9     <stem1> == p i t a r
```

Each of the various lines are extracted from different sections of the networks. I collected them to illustrate different aspects of KATR syntax. They do not appear in this order in the actual networks, and some of the information has been left out for brevity. Line 1 represents how variables are assigned in KATR. First the line begins with #var to indicate the following information will be the variable label and the value(s) assigned to the label. In this example, the variable assigns the various values that trigger the Ruki Rule in Sanskrit. This variable allows me to have a single rule for the Ruki Rule instead of one for every possible sound that triggers the rule.

Line 2 gives an example of a sandhi rule. The line begins with #sandhi to indicate a phonological rule will follow. This rule is the Ruki Rule from above. The rule has an LHS and an RHS where the LHS is converted into the RHS. The LHS instructs the computer to look for any of the values for the ruki variable from above, immediately followed by the letter *s*. When the computer sees this exact sequence, the RHS indicates that the computer should output exactly the value it read for $ruki. The $1 in the RHS represents this function. Then, the RHS instructs the computer to replace the *s* with *ṣ*.

Lines 3 through 5 gives only a small portion of how the r-stem nominals are formed within the KATR networks. I have only included these to exemplify certain syntactic elements. The full rule for r-stem formation will be addressed in Chapter 3. Line 3 has the node name that can be referenced by other nodes and marks the information to be contained by the node. Line 4 is the rule for how the strong cases are to be formed. The LHS has the variable $strong (defined as nominative, accusative, and vocative early in the networks), and this side is defined by the RHS, which states to use the information stored as stem1 and add on the information stored in the Endings node. I explain the information for stem1 below and cover the Endings node

[15] The periods at the end of lines 1 and 2 indicate that set of values for the variable and the sandhi rule is complete respectively.

in Chapter 3. Line 5 has the default path. Instead of having attributes on the LHS, the path is empty to allow this node to have a default out in the event information not otherwise specified is given to it. If this were how the r-stem nominals were formed, the weak cases would be instructed to follow the rules in the Base-Nouns2 node (See Chapter 3 for an explanation of the Base-Nouns2 node).

Lines 6 and 7 shows how the neuter a-stem nominals are formed. Just like the node above, only a potion of the full node has been included to exemplify the relevant information. The LHS of the rule on line 7 has a set of attributes that indicate the RHS only applies to the nominative singular. The RHS contains a path to have the RHS equal to the RHS of the rule for the accusative singular in the node A-Masculine. This path allows for connected concepts to have interconnected nodes. If a change occurs to the value for the masculine accusative singular, the neuter nominative singular would also change due to the interconnected nature of the nodes.

The final set of lines, 8 and 9, is a section of the lexical entry for 'father.' Just as for lines 3 and 6, this line begins with the name of the lexical node and contains all the information all the basic information, such as nominal class and the various stems, as well as the information unique to a particular lexeme. Line 9 is the rule for assigning stem information. The LHS has the label of stem1, and the RHS contains individual letters instead of nodes or paths to follow.

1.3 Outline

I organized the remainder of the thesis as follows. In chapter 2, I describe my methods and connect the hypothetical nature of PIE to the well-documented Indo-European (IE) language Sanskrit, which shows how conservatively Sanskrit has inherited certain features of PIE phonology, morphology, and morphophonology. Testing the theories on Sanskrit provides an empirical basis for applying the same theories on PIE.

In chapters 3 and 4, I use a guided derivation of a Sanskrit r-stem nominal (*pitṛ-*) and a Sanskrit a-stem nominal (*sukha-*) to walk us through each step. The purpose of these chapters is to explain the final output forms based on the shape of the networks. I have chosen these specific words to be representative of the mobile accent class and the static accent class. We see that networks for the Paradigmatic Theory and the Compositional Theory share many of the same features without compromising the basic tenets of each accentuation theory.

Chapter 5 is an analysis of my results for the two networks from chapters 3 and 4 and then the overall conclusions I have drawn from the project. It is in this chapter that we come to an answer about the elegance of each KATR network in a computational framework.

Chapter 2 Methodology

In order to look at the computational complexity[16] of the two major theories for PIE accent assignment, I have created the following KATR networks: the Paradigmatic Network (PN) and the Compositional Network (CN). Each network shared as many KATR rules as possible to limit the number of variables.

Although the goal of the computational modeling of nominal morphology is to look into PIE accent assignment, I have built the networks for Classical Sanskrit. As Sanskrit is a documented language, none of the cells of the paradigms need to be reconstructed or based on speculation. Every instance of speculation decreases the reliability of the results. The level of confidence in the reconstruction of PIE noun paradigms is not even across the nominal components: the root, suffix, or ending. Even with secure[17] roots and inflectional stems (the result of applying one or more derivational suffixes to a root), the endings are less secure: the strong endings are rather secure, the weak endings, especially those of the dual, are less secure.

The reasons Sanskrit was chosen over PIE are two-fold. Firstly, documented languages have actual data. As PIE is a reconstructed language, all of the features are hypothetical and are only as secure as the robustness of the data. In contrast, documented languages have features that experts can directly observe. For this reason, I chose to model one of the daughter languages.

Secondly, Sanskrit is one of the more conservative Indo-European languages when it comes to both morphology and morphophonology. Sanskrit, along with other ancient Indo-Iranian languages, preserves the 8 cases and 3 numbers traditionally reconstructed for PIE [For10]. Sanskrit allows for the KATR network to reflect any processes that may be unique to specific cases while also allowing for representations of the whole PIE paradigm. The major caveat to this is the dual: a number reconstructed for PIE but with insecure oblique endings due to the sparsity[18] of data in IE languages [Rup17].

I have constructed the KATR theories with as many commonalities as possible. The place with the greatest overlap between the theories is the morphological information (i.e. the case endings) and the words chosen. For this project, I have used the paradigms found Ruppel (2017) for ease of access and reference. I have created a lexicon with 6 entries: each entry either belongs to a different declension or a different gender when the declension type repeats. For one declension, each of the three genders have different enough forms that it warranted having all three genders represented in the lexicon; for the other declension, the masculine and feminine genders do not differ morphologically within the word itself [Rup17].

The basic structure of each of my KATR networks is the same: the sandhi rules, the case endings, the noun formation rules, and finally the lexicon. The sandhi section

[16]Please refer to Chapter 1 for the definition of complexity.

[17]Security refers to the robustness of the reconstruction.

[18]The main source for the dual is Greek, the Indo-Iranian languages, and the Balto-Slavic languages [LY18].

Table 2.1: A reminder of the paradigm for *pitṛ-* ‘father’.

	Singular	**Dual**	**Plural**
Nominative	*pitā́*	*pitárau*	*pitáraḥ*
Vocative	*pítaḥ*	*pítarau*	*pítaraḥ*
Accusative	*pitáram*	*pitárau*	*pitṝ́n*
Instrumental	*pitrā́*	*pitṛ́bhyām*	*pitṛ́bhiḥ*
Dative	*pitré*	*pitṛ́bhyām*	*pitṛ́bhyaḥ*
Ablative	*pitúḥ*	*pitṛ́bhyām*	*pitṛ́bhyaḥ*
Genitive	*pitúḥ*	*pitróḥ*	*pitṝā́m*
Locative	*pitári*	*pitróḥ*	*pitṛ́ṣu*

contains three sets of rules. The first set is the phonological rules for creating the output forms. There are rules based on Sanskrit phonology as well as rules involving morphophonemes as a way of differentiating different types of the same phoneme. For example, there are different ways that the stem final *i* operates: in the *ī*-stems, it becomes a *y* before a vowel; and in the *i*-stems, it becomes an *e* at the end of the word. Morphophonemes are a highly capable tool available to KATR; however, they were used sparingly to prevent abusing their capabilities. Whenever possible, I favored more specific sandhi rules as well as adding another stem. It is only when I have no choice do I use the morphophoneme. These are usually represented with capital letters instead of the lowercase letters used for actual segments.

```
#sandhi I $vowel => y $1 .
    #sandhi U $vowel => v $1 .
FIRE: %agníḥ
    <stem1> == a g n I
    <stem2> == a g n a y
    <> == I-Masculine-Nouns
```

The second set of rules in the sandhi section contains the accent rules. For the paradigmatic network, the only accent rules are those for removing the extra accents created by combining inherently accented stems with the inherently accented endings. Since it is important to have as much consistency as possible to prevent adding bias[19] to one network, all the endings have an inherent accentedness (i.e. accented or unaccented). If an accented stem ends up with an accented ending, the accent on the ending is removed with a sandhi rule. For the compositional network, this subsection contains more rules as there is only one stem that must handle all the different accent placements. This subsection also contains the benefit of KATR over DATR as most of these rules are nonlocal, meaning that rules may ignore intervening sounds in their search for target segments.

The final set of the sandhi section contains the rules for cleaning up the various remaining variables and relics of how the various rules combine with the different

[19]Bias refers to favoring one network over another and inadvertently or intentionally adding unnecessary lines to the network biased against. Examples of where attempts at avoiding bias appear below.

characters to be printed in the network. An example of this is when diacritics are applied as a result of various phonological rules. At times the macron marking length would appear after the acute marking the accented syllable had been applied, resulting in the diacritics out of order. To fix this, a rule was created to flip the diacritics around. Another rule in this section is the single graphemes used to represent the diphthongs. Other rules were used that would alter the output when *a* and *i* or *a* and *u* were placed next to each other. If I did not make the diphthongs different from the concatenated graphemes, the rules I had created would overapply, underapply, or require extremely specific rules that would only work for a particular environment.

The next section is the morphological information, specifically the endings. Both theories use the same set of endings to prevent any biases from entering at this point. The structure is broken down into three different sections: one for the base endings, another for the a-stem nouns, and a third for r-stems. These nouns are grouped based on how their stems end [Whi02]. Although there are other irregularities among the different declensions, these differences were handled without the need for extra endings. New endings were only added when no transparent phonological rule could be created.

The third section is devoted to the nominal formation process necessitated by KATR. This section contains subsections for each of the declensions included in the networks when a generic process cannot create the desired form. An example of a pattern not bound by a particular declension is the n-epenthesis in the genitive plural. Many of the declensions have an *n* before the genitive plural ending, even when the particular noun does not end in an *n*. Since this is rather systematic, it could be encoded into the networks directly without modifying any of the previous sections, and since the process is not restricted to a particular declension or lexical item, new subsections within this section were created specifically for this reason.

The final section of the networks is the lexicon of the nominals included. Here all the information specific to a single word is stored, which in the case of this project is just the various stems. This section is the spot for the greatest diversity between the two different KATR networks. The network for the paradigmatic theory often has multiple stems per lexeme. On the other hand, the network for the compositional theory almost always has one stem per lexeme.

2.1 Paradigmatic Theory

Phonological Rules

As mentioned above, the paradigmatic theory begins with the phonological rules. In order to combine as many of the endings as possible, I relied on synchronic Sanskrit phonology. I have rules for repairing scenarios where two vowels end up side by side.

```
#sandhi a a => ā .
#sandhi a i => e .
#sandhi a u => o .
```

In the above example, *a* followed by *i* always results in *e*, based on an internal sandhi rule of Sanskrit. To prevent overapplication, I temporarily had the two diphthongs mapped to the characters *ạ* for *ai* and *ḁ* for *au*. The choice for the remapping is completely arbitrary because KATR does not have a restriction on the symbols available for use, besides regular expression operators (e.g. *, +, and ?) [Fin+02]. Examples of the *ạ* and *ḁ* in code follow:

```
#sandhi a ạ => ạ .
Endings:
    ...
    {$strong du} == ḁ wordEnd
    ...

```

As illustrated above, I have encoded various internal vowel sandhi rules into the network; however, these rules are not the only phonological rules encoded into the network. I also designed rules for various consonantal rules as well, specifically the Ruki Rule [Beg17] and the Retroflexion of n [Rup17], illustrated below.

```
#sandhi $ruki s => $1 ṣ .
#sandhi $retro $noretro* n => $1 $2 ṇ .
```

The final set of KATR rules for the phonology handles situations where the networks incorrectly generate forms. For example, the correct genitive plural of nominals ending in -*nām* has a long vowel immediately preceding the ending; however, the network would only generate a long vowel there if the stem used for the genitive plural happened to end in a long vowel. To prevent creating extra stems, I created a specific rule that lengthened a short vowel, illustrated below.

```
#sandhi $short n ā m => $1 ¯ n ā m .
```

Morphophonological Rules

The morphophonological section of the sandhi rules are rather simple for the paradigmatic network and could even be considered a form of clean-up rules. I only considered them to be morphophonological as they deal with accents, and these same rules are considered morphophonological in the compositional network. These rules either ensure that only one accent is present in the final output or they handle situations when an accented vowel appears before another vowel illustrated below.

```
#sandhi á i => é .
#sandhi $accented $abc* á̄ => $1 $2 ā .
```

Clean-Up Rules

These rules clean up the various morphophonemes and other relics of the code that remain in the output, resulting in awkward, strange, or technically incorrect forms. Although the various phonological rules above deal with most of the morphophonemes, some of the them survive as illustrated below.

```
#sandhi ∅ => .
#sandhi wordStart => .
#sandhi wordEnd => .
```

Another set of clean-up rules are character replacements. As mentioned above, the Sanskrit diphthongs were indicated with single characters to prevent overapplication of various phonological rules. It is at this step that the characters are restored to their proper orthographic shape. In addition, there are the graphical errors where diacritics are doubled or out of order based on various preceding phonological and morphophonological rules. It is at this stage that the network is instructed to output the proper grapheme, which the following rules illustrate.

```
#sandhi ạ => a i .
#sandhi ḁ => a u .
```

2.2 Endings

The next section of the paradigmatic network is the groups of endings, grouped into two macro sections with the second being broken into smaller specific sections. The first section contains the base endings that a nominal would use if Sanskrit had a single declension, which is the first second of code below. These endings were taken from a chart in Ruppel's textbook for consistency [Rup17]. The second section contains all the declension-specific alternative endings. If a phonological rule or nominal formation process (explained in detail in the following subsection) could not be constructed satisfactorily, then an entry in this section was created with the appropriate information.

```
Endings:
    {nom sg} == ḥ wordEnd[20]
    {$strong du} == ḁ wordEnd
    {nom pl} == a ḥ wordEnd
    {voc sg} == wordEnd
    {voc pl} = <nom pl>
    {acc sg} == a m wordEnd
    {acc pl} == A ḥ wordEnd
    {ins sg} == ā́ wordEnd
    {ins du} == bh y ā m wordEnd
    {ins pl} == bh i ḥ wordEnd
    {dat sg} == é wordEnd
    {dat du} = <ins du>
    {dat pl} == bh y a ḥ wordEnd
    {abl sg} == ḥ wordEnd
    {abl du} = <ins du>
    {abl pl} = <dat pl>
    {gen sg} = <abl sg>
    {gen du} == ó ḥ wordEnd
```

```
    {gen pl} == ā́ m wordEnd
    {loc sg} == i wordEnd
    {loc du} = <gen du>
    {loc pl} == s u wordEnd
```

Because the KATR networks lack the historical developments of Sanskrit due to these developments not being programmed into the networks, the phonological and nominal formation rules were restricted to rules that a Sanskrit speaker speaker may have abstracted, meaning that analogy or diachronic relics would allow the creation of a new phonological or nominal formation rule. An untrained computational network cannot understand that consonants and vowels are phonologically different [Hay09]. This is simultaneously the power and caveat of computational modeling, which is why it is necessary to encode any necessary information in the beginning.

Whenever possible, the Endings list was linked to the other nodes, because part of the purpose of Network Morphology is the construction of interconnected nodes within a larger network [BH12]. The Endings list is the default list; however, neuter nouns are first linked to a base Neuter-Endings list to grab the appropriate neuter endings that follow the neuter rule in Indo-European languages (i.e. the nominative and accusative of any neuter noun are identical in all numbers regardless of the declension), and this Neuter-Endings list then links to the base Endings list so the neuter nouns are able to access the endings otherwise missing from the base Neuter-Endings list, as illustrated below.

```
Neuter-Endings:
    $strong sg == wordEnd
    $strong du == ī wordEnd
    $strong pl == i wordEnd
    <> == Endings
```

Nominal Formations

After defining all the necessary endings any given nominal would need, the rules for actually generating the forms to undergo the various phonological, morphophonological, and clean-up rules need to be defined. Similar to the Endings subsection, there is a base formation that all nominals would follow in the most ideal situation, and from there the various declension specific exceptions have been defined, as illustrated below.

```
Base-Nouns:
    <> == <stem1> Endings
```

The various declension specific exceptions to the base formation rule is not always connected to the deviation from the Endings list specific to the declensions because some of the formation deviations are the result of needing a different stem while still needing the base ending.

[20] The wordEnd is a morphophoneme that marks the end of the word in the event that I ever need to indicate that.

```
A-Masculine-Nouns:
    {ins du} == <stem1> a Endings
    {dat du} = <ins du>
    {dat pl} == <stem1> i Endings
    {abl du} = <dat du>
    {abl pl} = <dat pl>
    {gen du} == <stem1> y Endings
    {gen pl} = N-Gen
    {loc du} = <gen du>
    <> == <stem1> A-Masculine
```

The lines above are an example of an exception to the base nominal formation rule: the exceptions for the a-stem masculine nouns. This list of exceptions bypasses the base formation rule entirely due to needing to make sure that the default endings come from the list of deviations to the Endings list first. In addition to bypassing the base rule, a few other interesting components can be mentioned. First there are a total of 3 epenthetic rules where different characters are placed between the stem and the ending. These characters create the necessary environments for the phonological, morphophonological, and clean-up rules. The second component are the interconnections between the case-number nodes where the instrumental dual is linked with the dative dual, which is then linked with the ablative dual. The link was to simplify information in the network as well as show the syncretic nature of this portion of the Sanskrit nominal declension system. The ablative was linked to the dative instead of the instrumental directly based on the fact that the ablative also shares its endings with the dative in the plural, showing a stronger connection between these two cases. The final component of note is the node referred to as N-Gen, as illustrated below.

```
N-Gen:
    {gen pl} == <stem 1> n Endings
```

This node referenced in the a-stem masculine noun formation list is actually a formation rule for creating the genitive plural as shown above, where an *n* is placed between the stem and the ending. I chose to create a separate node for this instead of placing it in the a-stem masculine noun formation list because this process of *n*-epenthesis is found in several of the declensions. Since this commonality is present in the formation of the genitive plural in several declensions, I chose to make it an independent node the lists would link to.

I debated the rationale of the N-Gen node since the option to create a new ending specifically for the genitive plural was available to me. In the end, I decided that it made more sense to posit an N-Gen node, as its presence in most paradigms is due to analogical spread from the n-stem paradigm, and it is specifically not a new ending. If I wanted to create a new ending for the subset of declensions to use *-nām* instead of *-ām*, I would need to either place the new genitive plural in each of the declension lists (a source of unnecessary redundancy that I believe speakers would not initially create) or find the source of the *-nām* genitive that the other declensions reanalyzed the ending from and then have the other declensions link to that one for the genitive plural. The level of complexity for the latter was greater than I was willing to allow, but I do believe it is a step in the process of moving from epenthesis to complete ending replacement. These KATR networks just reflect a stage where this hypothetical speaker has yet to fully make the change underlyingly even if they would appear to do so on the surface level.

Lexicon

The final section of the network is the lexicon containing every word that I want a paradigm for. The code uses this information in the construction of each case-number form. For the paradigmatic network, several entries contain two or more stems. The choice between creating a new stem and allowing for a phonological, morphophonological, or clean-up rule or nominal formation rule to alter an existing stem came down to the complexity of the rule and the specificity. If it had been more computationally complicated[21] to create a series of additional rules for a single cell for a subset of words than creating a new stem, then I have created a new stem for those particular nouns. The computer, unless told to bypass information, does not inherently know the specificity of certain rules, so it would attempt to apply, for all intents and purposes, vacuous rules, because they were on the list of rules to use. Also if a new rule would have to be specific to each lexeme to undergo whatever process prompted the new rule, a new stem was favored as well.

2.3 Compositional Theory

Combined Phonological, Morphophonological, and Clean-Up Rules

Unlike with the Paradigmatic Network, the phonological and morphophonological rules are not as discrete in the Compositional Network. As a result, I have chosen to discuss all the sandhi rules at once for this network. There are minor differences and more rules for the compositional network to allow for accent assignment, but the overall set of rules is roughly the same. The biggest difference is the rules for constructing the vocative forms. There is a morphophonological rule as well a series of clean-up rules, as illustrated below.

```
#sandhi wordStart[22] $cons $cons* $vowel $abc* ∅ => wordStart $1 $2 $3 ’ $4 ∅ .
#sandhi a ’ => á .
#sandhi ā ’ => ā́ .
#sandhi e ’ => é .
#sandhi i ’ => í .
#sandhi u ’ => ú .
```

[21] Computational complexity refers to the amount of information in the network needed to generate a paradigm. The more information needed, the more complex the network. Complexity is discussed further in Chapter 5.

[22] The wordStart is a morphophoneme marks the start of the word in the event that I ever need to indicate that.

Nominal Formations and Lexicon

The nominal formation rules are nearly identical to the rules for the paradigmatic network. This was intentional to prevent as much bias as possible. Since the nouns in compositional lexicon only have one stem assigned, all of the nodes only list stem1. This means that if there are nodes in the paradigmatic network that look for stem1 or stem2, these have been consolidated into a single node.

```
Paradigmatic Network
N-Gen1:
    {gen pl} == "<stem1>" n Endings
N-Gen2:
    {gen pl} == "<stem2>" n Endings
```

```
Compositional Network
N-Gen:
    {gen pl} == "<stem1>" n Endings
```

The code above is how the genitive plural is constructed in the networks for when an *n* appears between the stem and the ending as a result of analogy from elsewhere in the system. For the paradigmatic network, certain noun classes have the genitive plural using the first stem (e.g. the a-stems) or the second stem (e.g. the r-stems). This is not possible for the compositional network, so only one N-Gen node even exists.

As mentioned before, the lexicon in the compositional network only contains a single lexical entry and the nominal formation rules necessary for generating the forms. This is the biggest difference between the two networks.

```
Paradigmatic Network
FATHER: %pitā
    <stem1> == p i t á r
    <stem2> == p i t ṛ́
    <nympha> == p í t a r
    <> == R-Nouns
```

```
Compositional Network
FATHER: %pitā
    <stem1> == wordStart p i t a r
    <> == R-Nouns
```

Chapter 3 Guided Derivations for R-Stems

In each chapter, we will see how different sets of endings are accessed during the derivation. The nominal chosen for the r-stem derivation is pitr- 'father', one of the hallmark words of PIE Studies [May86]. These nominal classes illustrate how both mobile and static accent placements work within both systems. I also wanted to illustrate how different sets of endings would be accessed during the derivation. The nominal that I have chosen to represent the r-stems is the Sanskrit word for 'father' *pitr̥-* because this word is one of the hallmark words of PIE Studies.

3.1 Characters

Before we begin, it would be good to explain the various characters found within the networks. In addition to the 26 individual minuscule letters in the Latin alphabet available, one also has the option to use the majuscule letters. By default the code assumes that majuscule letters are part of a node label, so using them for a particular purpose as data in the network causes an error. To bypass this problem, the only majuscule letters listed are the ones used as atoms.[23].

The only characters used that were not plain Latin letters were accented letters, and the symbols {}, <>, %, #, $, =, and *. The accented letters come in three types: those with acute accents, those with the underdot, and the character with the under-ring. The acute accented characters are used to represent the locations of the accent. The characters with the under-ring represent the vocalic *r* (*r̥*), the visarga (*ḥ*), the retroflex series of stops, or within the analysis, a singular character for the *ai* diphthong (*ạ*). The use of the *ạ* character for this function is unique to me as I needed a singular character is needed to represent this singular sound, normally transcribed as the digraph *ai*. The *ḁ* character likewise represents the *au* diphthong with the same reasoning as the *ạ* character.

The final characters are used for their computational functions. The curly brackets mark sets of features for a definition, and the angled brackets mark a path. The percent sign marks a comment, which the code ignores but the programmer can read. The pound sign marks a sandhi rule and the dollar sign marks a variable. The equal sign marks a fact definition or equates one side of a rule to another. The asterisk, called a Kleene star in computer science, indicates zero or more repetitions of the previous item.

3.2 Paradigmatic Derivation

The guided derivation begins with the Paradigmatic Network, following the steps taken by the athematic, mobile nominal *pitr̥-*. First, KATR calls up the lexicon,

[23]KATR normally sees the majuscule letters as being part of a label. An atom is a character not part of a label.

which designates *pitr̥-* as an r-stem nominal, meaning that its lexical entry contains a path to the R-Nouns nominal formation node[24]. The lexicon also contains the three stems needed by this network for nominal formation. Because there is nothing unique about *pitr̥-* that would be required to be stored in the lexicon and not later nodes, the stems and nominal type complete the lexical entry:

```
94 FATHER: %pita
95     <stem1> == p i t á r
96     <stem2> == p i t ṛ́
97     <nympha> == p í t a r
98     <> == R-Nouns
```

Above is the lexical entry for FATHER (lines 94 through 98) and below are the nominal formation rules for the r-stem nominals (lines 99 through 104). Line 94 marks the following information as being the lexical information that describes *pitr̥-*. For the purposes of this thesis, only the stems have been included, as well as the nominal class this word belongs to as that determines the various forms of its paradigm. Lines 95-97 contain the stems used. As this KATR network would have multiple stems for the r-stem nominals, each stem is referred to by the order they appear in the lexicon; however, the label 'nympha' in line 97 refers to the Greek word that exemplifies the vocative stress rule [Byr18]. There is no need to order the information in the lexicon specifically as Pāṇini's Principle[25] applies in KATR [Fin16]. It is the left-hand side (LHS) of a rule that determines the precedence of a rule's right-hand side (RHS). Since lines 95 through 97 all have paths in the LHS position, they are more specific than line 98 which lacks any specific path. This will be the path taken if the more specific paths can not apply. Table 3.1 is how the network generates the paradigm with just this information.

Table 3.1: The current paradigm only contains the information of just the lexical entry and not the nominal formation node. Although the network is able to access the three stems, it is unable to place them in the correct cells due to a lack of information thus far.

	Singular	**Dual**	**Plural**
Nominative			
Vocative			
Accusative			
Instrumental			
Dative			
Ablative			
Genitive			
Locative			

[24]The names of any node are completely arbitrary but have been given to aid in the identification.

[25]Pāṇini's Principle states that the most specific rule applies before less specific rules [Fin16].

```
99  R-Nouns:
100     {$strong} == <stem1> Endings
101     {voc} == <nympha> Endings
102     {acc pl} == <stem2> A-Masculine
103     {loc sg} == <stem1> Endings
104     <> = Base-Nouns2
```

Line 99 is equivalent to line 94 above in that it labels the following information as belonging to a particular node, in this case the R-Nouns node. Without any other nodes, we can start to fill in some of the cells, even if partially. Line 100 shows the generation of the strong case forms where ‘stem1’ is combined with the matching information in the Endings node. The appropriate endings will be defined by later nodes. The variable $strong has been defined as the nominative, vocative, and accusative for all numbers. The accusative plural is actually a weak form [For10]; however, the variables have been defined with only one set of values (case, number, gender, etc.) and not a combination. This is for more general statements with the exceptions more targeted and truly exceptional. With just the first line, the partially generated paradigm would look like Table 3.2:

Table 3.2: The partially generated paradigm for $strong only with the remaining cells omitted for brevity.

	Singular	**Dual**	**Plural**
Nominative	p i t á r	p i t á r	p i t á r
Vocative	p i t á r	p i t á r	p i t á r
Accusative	p i t á r	p i t á r	p i t á r

Line 101 contains the rule for generating the vocative forms. The LHS here is more specific than the previous line due to atoms having “slightly more precedence than variables;” thus this rule applies before the previous rule and overrides the conflicting values (i.e. line 100 generates p i t á r $\varnothing$ for the vocative singular but line 101 overrides this output and generates p í t a r $\varnothing$ due to line 101 having a higher precedence than line 100. The inclusion of line 101 is to accommodate the special accentuation pattern that the vocative follows, where the accent is static on the first syllable [Byr18]. Table 3.3 shows the updated partial paradigm.

Table 3.3: The partially generated paradigm for $strong only.

	Singular	**Dual**	**Plural**
Nominative	p i t á r	p i t á r	p i t á r
Vocative	p í t a r	p í t a r	p í t a r
Accusative	p i t á r	p i t á r	p i t á r

Lines 102 and 103 are more specific than line 101 as the LHSs contain the specific case-number pairings the rules apply to instead of just a specific case or class of cases. These rules are the most specific rules in the R-Nouns node and override any other rule that would attempt to generate a form for these specific case-number pairings. As mentioned above, the accusative plural is actually a weak form instead of an expected strong form like the accusative singular and dual. The evidence for this can be seen as the accusative plural will use the same stem as the weak cases whenever there are any ablauting stems. Instead of a form like ^{x}*pitr̥ān*, the correct form lacks a vowel after the *r* of the root. Table 3.4 is an update to the accusative cells only.

Table 3.4: The accusative cells only.

	Singular	**Dual**	**Plural**
Accusative	p i t á r	p i t á r	p i t ŕ̥

```
105 R-Nouns:
106     {$strong} == <stem1> Endings
107     {voc} == <nympha> Endings
108     {acc pl} == <stem2> A-Masculine
109     {loc sg} == <stem1> Endings
110     <> = Base-Nouns2
```

Just as line 108 takes precedence over line 106, line 109 takes precedence over line 110. This precedence marks how the locative singular needs to be formed differently than would otherwise be generated by line 110. Whereas the nominative, vocative, and accusative are the strong cases, the instrumental, dative, ablative, genitive, and locative are the weak cases. For some nominals, the locative singular behaves more similarly to the strong cases (i.e. possessing a strong stem instead of a weak stem). To accommodate these nominals, a rule like line 110 prevents the locative singular from being grouped with the rest of the weak cases, including the rest of the locative numbers. Table 3.5 shows the incorrect generation without line 109 while table 3.6 shows the correct generation with the rule in line 109 applying. Without line 109, table 3.5 is how the locative stem would generate. The dual and plural stems have not been defined yet in the guided derivation; thus, the cells are empty. Table 3.6 shows what happens with line 109.

Table 3.5: Locative singular without r-stem rule.

	Singular	**Dual**	**Plural**
Locative	p i t ŕ̥		

Table 3.6: Locative singular with locative r-stem rule.

	Singular	**Dual**	**Plural**
Locative	p i t á r		

So far, the stem for the strong cases, the accusative plural, and the locative singular have been generated; however, we also need to generate the remaining cases.

Line 110 has the least specific LHS, thus making it the rule that applies when no other constraint is at work. The rule in line 110 tells the network that how to generate a cell in the absence of any further instructions. If this connection were missing, the network would not be able to proceed because the network is expecting information for 8 cases and 3 numbers. The RHS of line 110 is a path, instructing the network to retrieve the missing rules from the Base-Nouns2 node.

```
111 Base-Nouns2:
112     <> == <stem2> Endings
```

The Base-Nouns2 node[26] has a single line that indicates the default formation process is to use stem2 and apply the appropriate endings. As this is the default, all nominals that have a stem2 and use it for the majority of the cells in their paradigms can be connected together. With this, the full paradigm of stems has been generated in table 3.7:

Table 3.7: The full paradigm of stems for *pitr̥-*.

	Singular	**Dual**	**Plural**
Nominative	p i t á r	p i t á r	p i t á r
Vocative	p í t a r	p í t a r	p í t a r
Accusative	p i t á r	p i t á r	p i t ŕ̥
Instrumental	p i t ŕ̥	p i t ŕ̥	p i t ŕ̥
Dative	p i t ŕ̥	p i t ŕ̥	p i t ŕ̥
Ablative	p i t ŕ̥	p i t ŕ̥	p i t ŕ̥
Genitive	p i t ŕ̥	p i t ŕ̥	p i t ŕ̥
Locative	p i t á r	p i t ŕ̥	p i t ŕ̥

The next explanation for the guided derivation covers the A-Masculine node, which is collected by the network during the application of the R-Nouns and Base-Nouns 2 nodes, in addition to the lexical entry. The network accesses the A-Masculine node and the Endings node while applying the rules of the R-Nouns node. We will first look at the A-Masculine node as R-Nouns only references it once. The Endings node is basal to the network, meaning meaning that it forms the basis to which all changes occur.

```
113 A-Masculine:
114     {acc sg} == m wordEnd
115     {acc pl} == A n wordEnd
116     {ins sg} == i n a wordEnd
117     {ins pl} == ạ ḥ wordEnd
118     {dat sg} == a y a wordEnd
119     {abl sg} == a t wordEnd
120     {gen sg} == s y a wordEnd
121     <> = Endings
```

[26] As the name of this node implies, there is indeed a Base-Nouns1 node in the network. It just is not connected to the R-Nouns node.

Line 113 is equivalent to lines 94 and 99 above in that it labels the information following as belonging to a particular node, in this case the A-Masculine node. This node is a set of endings, defining different case-number suffixes, and not nominal formation rules. All but one of the lines in this node are specific to certain case-number pairings, giving them equal precedence to each other and having precedence over line 121, which is the default path as with line 110 above. In this case, this node is an exception[27] node to the Endings node, just as R-Nouns was an exception node to Base-Nouns2. This node tells the network to ignore the specified case-number from the Endings node and use the defined information instead. For the derivation of *pitr̥-*, only line 115 is important because the R-Nouns node tells us that we need the information in the RHS to generate the accusative plural correctly. The A-Masculine node states that the ending needs to be A n wordEnd so the new paradigm would have this as the accusative line:

Table 3.8: The singular and dual are still incomplete, but the plural now has its ending provided.

	Singular	**Dual**	**Plural**
Accusative	p i t á r	p i t á r	p i t ŕ̥ A n wordEnd

The shape of the accusative plural ending is unexpected because it contains items that are used by sandhi rules or capable of being used by sandhi rules. These items are called morphophonemes; the ones referenced in the A-Masculine node are the 'A' in line 115 and 'wordEnd' in lines 114 through 120. Morphophonemes are underlying segments whose surface outputs are dependent on morphological rules and not solely on phonological rules [Hay09]. Morphophonemes[28] allow for sandhi rules to apply in seemingly arbitrary conditions when in actuality it is the situation where a morphophoneme and a morpheme share similar appearances but have differing outcomes. The morphophoneme 'A' is distinguished from the character 'a' by being present in certain situations and absent in others. Without morphophonemes, a sandhi rule would need to be created for each and every output where the *a* is missing. There are clean-up rules that convert this morphophoneme into the proper orthographic character if it ever survives all the other sandhi rules; however, the sandhi rules normally delete it first (the sandhi rules will be addressed later in the guided derivation).

The morphophoneme 'wordEnd' at the end of the ending is found in all endings in the network to mark the end of word, so that word-final sandhi rules can be created. Currently it is not necessary; however, there are times where it could be, at the very least, temporarily useful to target the word-final environment while better and more concrete phonological rules are formulated. The clean-up rules always delete

[27]Exception refers to a rule or set of rules that have precedence over another rule or set of rules. An entire node can be considered an exception if all the of information contained within takes precedence over a node it is connected to.

[28]Morphophonemes are either always a capital letter or a string of letters without spaces in this thesis. In the latter situation, KATR sees the entire string a singular character instead of individual characters.

the morphophoneme ‘wordEnd’ to prevent it from being part of the output by the network.

The next node discussed is referenced by the R-Nouns node, the Base-Nouns2 node, and the A-Masculine node: the Ending node. This node as indicated by the A-Masculine node is the basal node for all endings. Unless otherwise specified, this is the node that provides the network with the endings for every case-number pairing. Lines 122 through 144 below is the complete Endings node.

```
122 Endings:
123     {nom sg} == ḥ wordEnd
124     {$strong du} == ạ̥ wordEnd
125     {nom pl} == a ḥ wordEnd
126     {voc sg} == ∅ wordEnd
127     {voc pl} = <nom pl>
128     {acc sg} == a m wordEnd
129     {acc pl} == A ḥ wordEnd
130     {ins sg} == ā́ wordEnd
131     {ins du} == bh y ā́ m wordEnd
132     {ins pl} == bh í ḥ wordEnd
133     {dat sg} == é wordEnd
134     {dat du} = <ins du>
135     {dat pl} == bh y á ḥ wordEnd
136     {abl sg} == ḥ wordEnd
137     {abl du} = <dat du>
138     {abl pl} = <dat pl>
139     {gen sg} = <abl sg>
140     {gen du} == ó ḥ wordEnd
141     {gen pl} == ā́ m wordEnd
142     {loc sg} == i wordEnd
143     {loc du} = <gen du>
144     {loc pl} == s u wordEnd
```

Just as with the A-Masculine node, we can see the same morphophonemes ‘A’ and ‘wordEnd’ appearing. These have the same functions as described for the A-Masculine node. We can also see in-node connections where endings for one case-number combination are the same as another case-number combination. For example line 127 has an LHS with the set {voc pl}, indicating that the RHS applies only to the vocative plural. Whereas the RHS of line 127 is a path that indicates it is identical to the RHS of the nominative plural (i.e. line 125). Any changes to the RHS of line 125 automatically applies to the RHS of line 127 without any extra work. This is how I reflected the syncretic nature of Sanskrit nominals in the networks. Another note is the lack of another node referenced, showing that we have reached the end of the formation process. Finally there is no empty path, (e.g. line 121), further indicating that this node is the final node in the chain. There should be no place where a case-number pairing is missing once the network accesses the node Endings. Table 3.9 shows how the paradigm is generated without any of the sandhi rules applying.

Table 3.9: The fully generated paradigm for *pitr̥-* before the phonological, morphophonological, and clean-up rules have been applied.

	Singular	**Dual**	**Plural**
Nom.	p i t á r ḥ wordEnd	p i t á r ḁ wordEnd	p i t á r a ḥ wordEnd
Voc.	p í t a r Ø wordEnd	p it́ a r ḁ wordEnd	p í t a r a ḥ wordEnd
Acc.	p i t á r a m wordEnd	p i t á r ḁ wordEnd	p i t ŕ̥ A n wordEnd
Ins.	p i t ŕ̥ ā́ wordEnd	p i t ŕ̥ bh y ā́ m wordEnd	p i t ŕ̥ bh í ḥ wordEnd
Dat.	p i t ŕ̥ é wordEnd	p i t ŕ̥ bh y ā́ m wordEnd	p i t ŕ̥ bh y á ḥ wordEnd
Abl.	p i t ú ḥ wordEnd	p i t ŕ̥ bh y ā́ m wordEnd	p i t ŕ̥ bh y á ḥ wordEnd
Gen.	p i t ŕ̥ ḥ wordEnd	p i t ŕ̥ ó ḥ wordEnd	p i t ŕ̥ ā́ m wordEnd
Loc.	p i t á r i wordEnd	p i t ŕ̥ ó ḥ wordEnd	p i t ŕ̥ s ú wordEnd

The next step for the derivation is to explain the sandhi rules which govern the phonological, morphophonological, clean-up processes of the network. It is fitting to use Sanskrit with KATR as Sanskrit's native phonological rules are also called sandhi rules. For the rest of the guided derivation, sandhi will refer to the KATR rules that were created to reflect the various phonological and morphophonological processes in Sanskrit as well as to clean up the relics of the nominal formation process.

As the network applies each of the sandhi rules to each cell of the paradigm, it runs through each rule top to bottom as defined in the network, and each rule is only accessed once. If a rule needs to be applied a second time, the rule will need to be added to the list a second time. This makes ordering the rules difficult but not impossible. Feeding, the creation of an environment for another rule to apply, and bleeding, the removal of an environment for another rule to apply [Kip82], must always be considered when creating new rules and their placement in the networks list.

The first section of sandhi covers the Sanskrit-specific rules as well as any rule that is not concerned exclusively with accent assignment nor the rules that clean up the output into the accepted orthography. The following rules apply only to the r-stems in the network:

```
145 #sandhi $ruki s => $1 ṣ .
146 #sandhi á r ḥ => ā́ .
147 #sandhi ŕ̥ H => ú ḥ .
148 #sandhi ŕ̥ A => r̥̄́ .
149 #sandhi r ∅ => ḥ .
```

As mentioned in the KATR syntax section of Chapter 1, line 145 is the rule for how the Ruki Rule in Sanskrit is applied to the forms generated by the nominal formation processes as long as they contain at the environment on the LHS of the rule. Line 146 converts the generated form of the nominative singular of r-stem nominals into the proper surface form as a reflection of Szemerényi's Law[29] at work, where Sanskrit's final ḥ is lost at the end of the word when immediately preceded by a resonant. This loss causes the preceding vowel to lengthen as a result of compensatory lengthening [Byr18]; [For10]. As the networks were coded for the r-stem and a-stem nominals, it is not necessary to create more generalized environments. Line 147 reflects a phonological process in Sanskrit were *r̥* becomes *u* before the morphophoneme H

As this is the first time we are running into a KATR sandhi rule, it is necessary to break it down before talking about the function to the network. Each sandhi rule starts with #sandhi and ends with a period to mark that the information between the two represents a rule for telling the network how to output the word. The information in a KATR sandhi rule can be any of the characters (e.g. letters, Kleene stars, variable letters) present in the network and a variable defined at the beginning of the network. The left and right sides of the rule are separated by the symbol =>. The left side tells the network what to look for in the words, and if the sequence is present, the right side is what is printed instead. This sequence is not limited to any particular section of an input, which explains why the morphophoneme 'wordEnd' appears.

These rules are designed to transform the generated forms into the proper output forms. The first, second and last are known Sanskrit sandhi rules dealing with how r works within Sanskrit phonology. The first rule encodes the Ruki Rule [Beg17], where s becomes retroflex *ṣ* after *r*, *r̥*, *u*, *k*, or *i*. This rule accounts for and produces that effect. The second states that the sequence *á r ḥ* is not allowed, at this point in the code, and if it is detected, it must be replaced entirely with the character *ā́*. The last rule states that if it is the last letter in the vocative singular (marked by the morphophoneme ∅), replace the *r* with the visarga (*ḥ*). The fourth rule is a reflection of how many times there is a vowel or at least an extra mora in the ending that results in a long vowel preceding the rest of the ending. The rule effectively has the morphophoneme 'A' fusing with the *ŕ̥*, causing the code to print *r̥̄́*. The reason for having the morphophoneme 'A' instead of a standard *a* is to prevent the overapplication of this rule. Making this rule too specific by listing the entire ending would add more than the minimum amount of information necessary for the desired output. The third rule reflects a quirk of Sanskrit phonology where *ŕ̥* becomes *ú* before the visarga. Although three of these rules contain an accented character, none of the rules alter the placement or realization of the accent, which is why these are not placed with the other accent rules below.

Let us see what the paradigm looks like after the application of the currently defined sandhi rules. Remember that although all the entries have each sandhi rule applied, only a subset of the rules will actually make any changes. Here is the updated paradigm:

To balance the biases, the endings between the two networks have been defined in the same way. This means that the weak endings already have accents placed on them, implying they are underlyingly accented. For the paradigmatic approach, this means that these accents may have to be deleted depending on where the accent rests for the surface form of a particular word. Although this does require an extra step at times, it prevents the need for redundant layers of endings (i.e. a layer of accented endings parallel to a layer of unaccented endings). The rule for wiping away any unnecessary accents is the same for both the networks. For FATHER, there are only

[29] Szemerényi's Law normally applies to word final *-*s* but, because Sanskrit word final -*ḥ* descends from PIE word final *-*s*, the process is the same.

Table 3.10: The paradigm after 4 sandhi rules applied.

	Singular	**Dual**	**Plural**
Nom.	p i t ā́ wordEnd	p i t á r ḁ wordEnd	p i t á r a ḥ wordEnd
Voc.	p í t a r ∅ wordEnd	p í t a r ḁ wordEnd	p í t a r a ḥ wordEnd
Acc.	p i t á r a m wordEnd	p i t á r ḁ wordEnd	p i t r̥̄́ n wordEnd
Instr.	p i t ŕ̥ á wordEnd	p i t ŕ̥ bh y ā́ m wordEnd	p i t ŕ̥ bh í ḥ wordEnd
Dat.	p i t ŕ̥ é wordEnd	p i t ŕ̥ bh y ā́ m wordEnd	p i t ŕ̥ bh y á ḥ wordEnd
Abl.	p i t ú ḥ wordEnd	p i t ŕ̥ bh y ā́ m wordEnd	p i t ŕ̥ bh y á ḥ wordEnd
Gen.	p i t ú ḥ wordEnd	p i t ŕ̥ ó ḥ wordEnd	p i t ŕ̥ ā́ m wordEnd
Loc.	p i t á r i wordEnd	p i t ŕ̥ ó ḥ wordEnd	p i t ŕ̥ ṣ u wordEnd

two sandhi rules dealing with accent:

```
150 #sandhi ŕ̥ $accented => r $1 .
```

For *pitr̥*- 'father', we can see that if the KATR network sees the sequence ŕ̥ plus any accented character in that order, it will replace the ŕ̥ with *r* and print the accented character as is. As KATR sandhi rules are sequential, the output of one rule is the input of the next. This requires that the rules be ordered appropriately to prevent bleeding wanted changes and feeding unwanted ones. Although this rule does not conflict with the sandhi rules above, it was placed after the others to create the three sections. Table 3.11 is how the paradigm looks after the application of the single accent-related sandhi rule in this network.

Table 3.11: The application of the applicable accent-related rule.

	Singular	**Dual**	**Plural**
Nom.	p i t ā́ wordEnd	p i t á r ḁ wordEnd	p i t á r a ḥ wordEnd
Voc.	p í t a r ∅ wordEnd	p í t a r ḁ wordEnd	p í t a r a ḥ wordEnd
Acc.	p i t á r a m wordEnd	p i t á r ḁ wordEnd	p i t r̥̄́ n wordEnd
Instr.	p i t r ā́ wordEnd	p i t ŕ̥ bh y a m wordEnd	p i t ŕ̥ bh i h wordEnd
Dat.	p i t ŕ̥ e wordEnd	p i t ŕ̥ bh y ā́ m wordEnd	p i t ŕ̥ bh y á h wordEnd
Abl.	p i t ú h wordEnd	p i t ŕ̥ bh y ā́ m wordEnd	p i t ŕ̥ bh y á h wordEnd
Gen.	p i t ú h wordEnd	p i t ŕ̥ ó h wordEnd	p i t ŕ̥ a m wordEnd
Loc.	p i t á r i wordEnd	p i t ŕ̥ ó h wordEnd	p i t ŕ̥ s u wordEnd

The final set of sandhi rules clean up the relics of the code, including the morphophonemes and substitute characters. The rules in lines 151 through 154 below are only the ones that apply to FATHER.

```
151 #sandhi ạ => a i .
152 #sandhi ḁ => a u .
153 #sandhi ∅ => .
154 #sandhi wordEnd => .
```

The rules in lines 151 and 152 convert the single characters *ạ* and *ḁ*, used as single letter versions of the diphthongs, into the standard, orthographic representations. The other two rules in lines 153 and 154 delete the morphophonemes '∅' and 'wordEnd', used to represent the vocative singular ending and to mark the end of the word. These characters are no longer necessary once all the other sandhi rules have been applied. As such these rules have to be placed at the end of the list so the various characters and morphophonemes can be targeted by the rules in the sections above if necessary.

After the application of all the sandhi rules and the nominal formation process stated above, the final paradigm is generated as seen in Table 3.12.

Table 3.12: The final output of *pitr̥-* for the Paradigmatic Network.

	Singular	**Dual**	**Plural**
Nom.	pitā́	pitárau	pitáraḥ
Voc.	pítaḥ	pítarau	pítaraḥ
Acc.	pitáram	pitárau	pitr̥̄́n
Instr.	pitrā́	pitŕ̥bhyām	pitŕ̥bhiḥ
Dat.	pitré	pitŕ̥bhyām	pitŕ̥bhyaḥ
Abl.	pitúḥ	pitŕ̥bhyām	pitŕ̥bhyaḥ
Gen.	pitúḥ	pitróḥ	pitr̥̄ṇā́m
Loc.	pitári	pitróḥ	pitŕ̥ṣu

3.3 Compositional Derivation

The compositional derivation for FATHER was more complex[30] for the sandhi rules but less complex for the lexicon. As the Compositional Theory has only one stem, the entry in the Compositional Network's (CN) lexicon for FATHER only contains the following information in lines 155 through 157 below.

```
155 FATHER: %pitā
156     <stem1> == wordStart p i t a r
157     <> == R-Nouns
```

As we can see, there are two differences between this lexical entry and the lexical entry from the Paradigmatic Network (PN). The first is the number of stems. Since accent assignment throughout the paradigm is not stored as part of the lexicon in the compositional approach but instead the result of various phonological rules, the lexicon only needs the minimal amount of information to generate the full paradigm [Kip10]. Any deviations from this stem will be handled by the rules in the phonological section. The second difference is the lack of any accented characters in the stem.

If we generate the partial forms for the cells at this stage, we would get a paradigm filled with just the information in the lexicon. It would be better to wait for partial paradigm generation until we get to the sandhi. Unlike the PN, the sandhi rules are

[30] As a reminder, computational complexity refers to the amount of information in the network needed to generate a paradigm.

not able to be grouped into three separate sections; instead, the sandhi rules were organized based on their functions: ablaut and high-dominance accentuation rules were placed first; the majority of the non-ablaut, non-accentuation transformations come next; then the lower-dominance accentuation rules; and finally the clean-up rules. We will begin at looking at the first group of sandhi rules: the ablaut and high-dominance accentuation rules.

158 #sandhi a $cons[31] $cons* $accented[32] => $1 $2 $3 .
159 #sandhi t a r => t á r .
160 #sandhi wordStart $cons $cons* $vowel[33] $abc* ∅ => wordStart $1 $2 $3 $4 ∅ .

Line 158 accounts for the ablaut pattern of FATHER. The code scans through the input form looking for the sequence of the character *a*, one instance of any character defined as a consonant, zero or more instances of any character defined as a consonant, and one instance of any character defined as an accented letter in that exact order. If the code finds this sequence, it will remove the *a* and output the rest without any further modification. Just as with the PN, the output of a sandhi form becomes the input of the next form, making it imperative to consider feeding and bleeding.

After line 158 the rule in line 159 then tries to assign an accent. It will place an accent on any instance of *a* that is followed by an *r*. Although this rule is extremely specific and does not accurately express the process, it is sufficient for the time being as there are no other lexemes in the data set that have been assembled for this thesis have the sequence of *t a r*. If the data set were to expand and include others lexeme that contain this sequence in a position where the *a* should not be accented, then the rule would have to be updated. The purpose of this rule is to account for the accent that shows up on this vowel or the syllable nucleus that the *t* is the onset for. Kiparsky [Kip10] calls this the Oxytone Rule, where an accent is placed on the rightmost syllable of an inflectional stem. As KATR cannot easily accent a class of sounds with a variable, I have opted to use a specific rule, but only when I do not need to manipulate the specific sequence later. This is because KATR views á and a+ as different from each other as A is from Z due to the Unicode values being different.

Similarly to line 159, line 160 also attempts to assign an accent. According to the vocative accentuation pattern, the first syllable is accented regardless of the position of the accent elsewhere in the paradigm. This is why it is important to have the morphophoneme 'wordStart,' so I could target the first syllable. To prevent the rule from applying to every cell in the paradigm, the character for the vocative has been added. This explains why the Base-Nouns formation rule specifically states that the vocative is formed with the morphophoneme '∅' before the ending is added. If this morphophoneme were not there, it would be much harder to target the vocative in

[31] The values for $cons are the letters k, kh, g, gh, ṅ, c, ch, j, jh, ñ, ṭ, ṭh, ḍ, ḍh, ṇ, t, th, d, dh, n, p, ph, b, bh, m, y, r, l, v, ś, ṣ, s, and h and the morphophoneme H.

[32] The values for $accented are the accented morphophoneme Á; the characters á, ā́, á, é, í, ī́, ó, ú, ū́, ạ́, ḁ́, ṛ́, and r̥̄́; and the morphophoneme H.

[33] The values for $vowel are a, e, i, o, u, ā, ī, ū, ạ, ḁ, á, ā́, é, í, ī́, ó, ú, ū́, ạ́, ḁ́, ŕ̥, and r̥̄ and the morphophoneme Á

the dual and plural since the vocative is only distinct from the nominative in the singular.

With just these three rules, accentuation has been properly assigned for some of the cells. Below Table 3.13 shows the generation of a partial paradigm with only the lexicon, the same nominal formation processes and endings from PN, and the three sandhi rules from lines 158 through 160.

Table 3.13: A partial paradigm having undergone only three sandhi rules after nominal formation.

	Singular	**Dual**	**Plural**
Nom.	p i t á r ḥ wordEnd	p i t á r ḁ wordEnd	p i t á r a ḥ wordEnd
Voc.	p í t á r ∅ wordEnd	p í t á r ḁ wordEnd	p í t á r a ḥ wordEnd
Acc.	p i t á r a m wordEnd	p i t á r ḁ wordEnd	p i t r Á n wordEnd
Instr.	p i t r ā́ wordEnd	p i t ŕ̥ bh y ā m wordEnd	p i t ŕ̥ bh i ḥ wordEnd
Dat.	p i t r é wordEnd	p i t ŕ̥ bh y ā m wordEnd	p i t ŕ̥ bh y a ḥ wordEnd
Abl.	p i t ú ḥ wordEnd	p i t ŕ̥ bh y ā m wordEnd	p i t ŕ̥ bh y a ḥ wordEnd
Gen.	p i t ú ḥ wordEnd	p i t r ó ḥ wordEnd	p i t r n ā́ m wordEnd
Loc.	p i t á r i wordEnd	p i t r ó ḥ wordEnd	p i t r s ú wordEnd

As we can see, the vocative cells contain two accents as a result of the vocative accentuation rule and the highly specific strong-case accent-assignment rule in line 159. The second accent is a case of overapplication, but may also represent a stage where the vocative could either realize with the special vocative accentuation pattern or align with the rest of the strong cases. The accusative plural, the ablative singular, and the genitive singular contain KATR characters used by sandhi rules for controlling what the code outputs. We can also see that most of the weak cases have incorrect accentuation as the previous rules were not able to assign the proper accents. These rules will come in the third section of the sandhi rules.

The next set of sandhi rules are equivalent to the first set of sandhi rules from the PN. As this set of rules are the same as the previous network, I will skip over them and move to the second set of accent sandhi rules. As these rules are concerned with the wiping away excess accents, it is important to explain these rules:

```
161 #sandhi i ’ $abc* á => í $1 a .
162 #sandhi r̥̄́ $abc* ā́ => r̥̄ $1 ā́ .
163 #sandhi $accented $abc* ā́ => $1 $2 ā .
164 #sandhi $accented $abc* á => $1 $2 a .
165 #sandhi $accented $abc* í => $1 $2 i .
166 #sandhi $accented $abc* ó => $1 $2 o .
167 #sandhi $accented $abc* ú => $1 $2 u .
```

Although it appears there are many rules, there are only three processes. The first process is line 161 which cleans up the double accents on the vocatives by wiping the second accent while preserving the vowel. The second process is on line 162 which wipes the accent created by the Oxytone Rule in favor of the accent on the ending.

The final process on lines 163 though 167 is a series of rules for wiping the ending accent in favor of the accent created by the Oxytone Rule. Lines 162 through 167 are ordered as the environment for the genitive plural is similar to the environment for the other cases. To prevent overapplication, the genitive plural needs to be fixed before the other cases.

```
168 #sandhi ạ => a i .
169 #sandhi ḁ => a u .
170 #sandhi ∅ => .
171 #sandhi wordEnd => .
```

The final set of sandhi rules are the clean-up rules, which are identical to the rules found in PN. Lines 168 through 171 recapitulate the rules. The processes and functions remain the same: remove all the remaining relics of the code and print the orthographically correct forms for the cells in the paradigm. Let's look at the final forms of the paradigm. Table 3.14 is identical to Table 3.12 but is the result of differing sandhi rules and lexical entry.

Table 3.14: The final output of *pitr̥-* for the Computational Network.

	Singular	**Dual**	**Plural**
Nom.	pitā́	pitárau	pitáraḥ
Voc.	pítaḥ	pítarau	pítaraḥ
Acc.	pitáram	pitárau	pitr̥̄́n
Instr.	pitrā́	pitŕ̥bhyām	pitŕ̥bhiḥ
Dat.	pitré	pitŕ̥bhyām	pitŕ̥bhyaḥ
Abl.	pitúḥ	pitŕ̥bhyām	pitŕ̥bhyaḥ
Gen.	pitúḥ	pitróḥ	pitr̥̄ṇā́m
Loc.	pitári	pitróḥ	pitŕ̥ṣu

Chapter 4 Guided Derivations for A-Stems

This next chapter will follow the creation of full paradigms for the word *sukha-* 'happy' for PN and CN. Although this nominal represents a thematic noun, it has a static accent [May86]. Given that acrostatic athematic nominals (such as PIE *nóḱ^{w}t-* 'night') also have static accentuation, this analysis of Sanskrit *sukha-* is representative of this type.

4.1 Paradigmatic Derivation

After the step-by-step derivation of an athematic nominal in the previous chapter, this chapter focuses on the reasoning behind the choices made in this analysis. Certain rules omitted from the previous chapter are present in this chapter as they are now applicable. The amount of overlap is restricted to only the necessary elements.

Just like with FATHER in the previous chapter, the cells of the paradigm for HAPPY have the same number of cases and numbers; however, gender is now an important point of consideration because HAPPY is an adjective and not a noun. For consistency, each gender is considered a separate nominal and has a separate entry in the lexicon. This separation is not ideal and should be condensed for future versions of the networks. For the guided derivation, I go over each gender as there are differences for each gender to cover.

To begin, I have listed the lexical entries for each gender below.

```
172 HAPPY_M: %sukháḥ
173     <stem1> == s u kh á
174     <nympha> == s ú kh a
175     <> = A-Masculine-Nouns

176 HAPPY_N: %sukhám
177     <stem1> == s u kh á
178     <nympha> == s ú kh a
179     <> = A-Neuter-Nouns

180 HAPPY_F: %sukhā́
181     <stem1> == s u kh á
182     <stem2> == s u kh ā́
183     <nympha> == s ú kh a
184     <> = A-Feminine-Nouns
```

Just as with FATHER, each form of HAPPY has at least one stem and a path for nominal formation. For the masculine and neuter (lines 172 through 179), only one stem is necessary to generate all the forms in the cells of the paradigm without having to alter the nominal formation process too much. The deviation from the standard is more acceptable in the a-stems as they represent the Sanskrit descending from the PIE thematic nominals [For10] in contrast to the r-stems which descend from

PIE athematic nominals. These nominals operated differently from the athematic nominals with different case endings in the weak cases in the masculine and neuter genders; however, the feminine gender takes the athematic endings instead of the thematic endings [For10]. Since the feminine behaves slightly differently, it makes sense to allow it to behave differently from the masculine and neuter.

```
185 A-Masculine-Nouns:
186     {voc} == “<nympha> Endings
187     {ins du} == “<stem1>” a Endings
188     {dat du} = <ins du>
189     {dat pl} == “<stem1>” i Endings
190     {abl du} = <ins du>
191     {abl pl} = <dat pl>
192     {gen du} == “<stem1>” y Endings
193     {gen pl} = N-Gen
194     {loc du} = <gen du>
195     <> == “<stem1>” A-Masculine
```

Just as with the nodes in Chapter 3, line 185 labels the information in the lines that follow it as part of the A-Masculine-Nouns node. Line 186 uses the stem form labeled ‘nympha’ from the lexicon and applies the endings from the Ending node. This rule is the first where the LHS only indicates the case instead of a case-number pairing. The reason for only including the case and not the number is due to the Vocative Rule from Chapter 1. Lines 187, 189, and 192 all follow the same basic structure. A letter is inserted between the stem and the corresponding ending from the Endings node instead of a node specific for the a-stems. Line 187 has an epenthetic *a* placed after the stem, line 189 has *i* instead of *a* as an epenthetic vowel, and line 192 has the consonant *y* instead of a vowel. Lines 188, 190, 191, and 194 are all paths to rules within this node instead of rules on their own. Lines 187 and 189 indicate that the dative and ablative dual forms are identical to the instrumental dual and that any changes to the instrumental dual should also be reflected in the dative and ablative dual. Likewise line 191 indicates that the ablative plural is identical to the dative plural. Line 194 indicates that the locative dual is identical to the genitive plural. Line 193 is a path to a different node. Since the path contains only the node label, the network uses the RHS in the N-Gen node with an LHS that matches the genitive plural. Finally line 195 is the default path. When the network accesses the A-Masculine-Nouns, this line indicates what should always be done when no exception is given for the case-noun pairing requested, just as the empty paths in the previous chapter. Here the information in ‘stem1’ is applied directly to the node A-Masculine, a set of endings for the a-stem masculine nominals discussed below.

```
196 A-Neuter-Nouns:
197     {voc} == “<nympha>” A-Neuter
198     {$strong} == “<stem1>” A-Neuter
199     <> = A-Masculine-Nouns
```

The A-Neuter-Nouns node is far simpler than the A-Masculine-Nouns node above or the A-Feminine-Nouns node below due to the nature of neuter nominals in Sanskrit. They only differ from the masculine gender in the strong cases. Otherwise, they are identical in form. Lines 197 and 198 are the only exceptions to the default path defined on line 199. Line 197 follows the same structure as the rule in the A-Masculine-Nouns node with the only exception being that this rule uses the endings in the A-Neuter node instead of the endings in the Ending node. Line 197 applies to the strong cases as we can see with the variable $strong in the LHS. The rule has the ‘stem1’ attached to the endings in the A-Neuter node (discussed below). Line 199 is the default path, and the network follows the path to the A-Masculine-Nouns node for the weak cases.

```
200 A-Feminine-Nouns:
201     {nom sg} == “<stem2>” A-Feminine
202     {$strong du} = A-Neuter-Nouns
203     {voc} == “<nympha>” A-Feminine
204     {ins sg} == “<stem1>” y Endings
205     {ins du} = A-Masculine-Nouns
206     {ins pl} == “<stem2>” Endings
207     {dat sg} == “<stem2>” y A-Feminine
208     {dat du} = <ins du>
209     {dat pl} == “<stem2>” Endings
210     {abl sg} == “<stem2>” yā Endings
211     {abl du} = <ins du>
212     {abl pl} = <dat pl>
213     {gen sg} = <abl sg>
214     {gen du} = A-Masculine-Nouns
215     {gen pl} = A-Masculine-Nouns
216     {loc sg} == “<stem2>” y A-Feminine
217     {loc du} = <gen du>
218     {loc pl} == “<stem2>” Endings
219     <> == “<stem1>” Endings
```

As the feminine a-stems pattern more similarly to athematic nominals like the r-stems than the masculine or neuter thematic nominals, the A-Feminine-Nouns node contain quite a few exceptions to the thematic pattern. Lines 202, 205, 208, 211, 214, 215, and 217 all connect to the masculine and neuter thematic nodes while lines 201, 203. 207, and 216 are all connected to a node unique to the feminine a-stems, which is explained below. The syntax is the same as the previous sections where the LHS contains case-number pairings when a rule is for an exception or the empty path for the default rule. The RHS contains the formation rules or paths for the network to follow for more formation rules. Lines 204, 207, 210, and 216 are rules where an epenthetic consonant or syllable appears before the ending nodes. Most of the exceptions that are not paths and the default all connect to the Endings node, reflecting that this nominal patterns with more similarly to the athematic nominals than the thematic nominals. This node shows the two stems of the feminine a-stems. The rules that connect to the A-Masculine-Nouns and the A-Neuter-Nouns nodes as well as lines 204 and 219 all use ‘stem1’ while the rest use ‘stem2’.

The next few paragraphs look at the endings for the a-stems that are exceptions to the Endings node from the previous chapter, broken down by gender and following the same order as the rules above (lines 185 through 219). The Endings node is not restated for brevity.

```
220 A-Masculine:
221     {acc sg} == m wordEnd
222     {acc pl} == A n wordEnd
223     {ins sg} == i n a wordEnd
224     {ins pl} == ạ ḥ wordEnd
225     {dat sg} == a y a wordEnd
226     {abl sg} == a t wordEnd
227     {gen sg} == s y a
228     <> = Endings
229 A-Neuter:
230     {$strong sg} = A-Masculine:<acc sg>
231     {$strong du} == i wordEnd
232     {$strong pl} == a n i wordEnd
233     <> = A-Masculine
234 A-Feminine:
235     {nom sg} == wordEnd
236     {voc sg} == i ∅ wordEnd
237     {voc du} = A-Neuter
238     {dat sg} == ạ wordEnd
239     {loc sg} == ā m wordEnd
240     <> == Endings
```

Lines 220, 229, and 234 label each node as the masculine a-stem endings, the neuter a-stem endings, and the feminine a-stem endings that differ from the Endings node respectively. As lines 221 through 227 and lines 235 through 239 are exceptions to the Ending node, lines 228 and 240, which are the default paths of their respective nodes, connect back to the Endings node so the missing case-number pairings can be properly generated. Although lines 230 through 232 list exceptions to the Endings node, A-Neuter node does not connect directly back to the Endings node but instead to the A-Masculine node. The reason is because that neuter nominals share weak case endings with masculine nominals. If I were to link back to the Endings node, I would not be able to use the weak case exceptions[34] in the Endings node.

Table 4.1 shows the partial derivation before the sandhi rules have been applied. As we can see, the appropriate stems are attached to the various endings from the nominal formation rules above, including any epenthetic sounds necessary. This is not the final output, however, because the forms are not quite right yet. Some of the

[34]Remember that although the accusative plural is considered a weak case, the networks treat the accusative in all numbers as strong and exceptions have been added to allow for the accusative plural to behave more like a weak case instead of a strong case.

Table 4.1: Each cell contains up to three entries. The first line is the masculine, the second is the neuter, and the third is the feminine. When there are only two entries, the first is the masculine-neuter with the second being the feminine. The morphophonemes 'wordEnd' and 'wordStart' have been excluded for brevity but are technically present at this stage.

	Singular	**Dual**	**Plural**
Nom.	s u kh á ḥ s u kh á m s u kh ā́	s u kh á ḁ s u kh á i s u kh á i	s u kh á a ḥ s u kh á a n i s u kh á a ḥ
Voc.	s ú kh a ∅ s ú kh a m s ú kh a i ∅	s ú kh a ḁ s ú kh a i s ú kh a i	s ú kh a a ḥ s ú kh a a n i s ú kh a a ḥ
Acc.	s u kh á m s u kh á a m	s u kh á ḁ s u kh á i s u kh á i	s u kh á Á n s u kh á a n i s u kh á a ḥ
Instr.	s u kh á i n a s u kh á y ā́	s u kh á a bh y ā́ m	s u kh á ạ ḥ s u kh ā́ bh í ḥ
Dat.	s u kh á a y a s u kh ā́ y ạ	s u kh á a bh y ā́ m	s u kh á i bh y á ḥ s u kh ā́ bh y á ḥ
Abl.	s u kh á a t s u kh ā́ y ā H	s u kh á a bh y ā́ m	s u kh á i bh y á ḥ s u kh ā́ bh y á ḥ
Gen.	s u kh á s y a s u kh ā́ y ā H	s u kh á y ó ḥ	s u kh ā́ n ā́ m
Loc.	s u kh á i s u kh ā́ y ā m	s u kh á y ó ḥ	s u kh á s ú s u kh ā́ s ú

cells have multiple accents, and some cells have phonological or morphophonological rules that need to be applied before the final output.

```
241 #sandhi a a => ā .
242 #sandhi a i => e .
243 #sandhi a ạ => ạ .
244 #sandhi a ḁ => ḁ .
245 #sandhi á ḁ => áu .
246 #sandhi á a => ā́ .
247 #sandhi á Á => ā́ .
248 #sandhi á i => é .
249 #sandhi á ạ => ái .
```

Lines 241 through 244 deal with the vocative formation process and lines 246 through 249 deal with the other cases' formation processes as the vocative always ends in an unaccented vowel and the other cases end in accented vowels in the a-stems. Normally the accent rules would be discussed after the pure phonological rules; however, since the a-stems have a static accent, there is no separation between the phonological and morphophonological sections of the sandhi rules. These rules do not delete accents present elsewhere in the word, however. For the paradigmatic network, only one accent is present in the strong cases ever while two accents might be present in the weak cases as a result of using the same Endings node as used for the r-stems to prevent over biasing CN.

Table 4.2: This table is before morphophonological and clean-up sandhi rules have been applied. Just as with the previous table, each cell contains up to three entries. The first line is the masculine, the second is the neuter, and the third is the feminine. When there are only two entries, the first is the masculine-neuter with the second being the feminine.

	Singular	**Dual**	**Plural**
Nom.	s u kh á ḥ s u kh á m s u kh ā́	s u kh áu s u kh é s u kh é	s u kh ā́ ḥ s u kh ā́ n i s u kh ā́ ḥ
Voc.	s ú kh a ∅ s ú kh a m s ú kh e ∅	s ú kh au s ú kh e s ú kh e	s ú kh ā ḥ s ú kh ā n i s ú kh ā ḥ
Acc.	s u kh á m s u kh ā́ m	s u kh áu s u kh é s u kh é	s u kh ā́ n s u kh ā́ n i s u kh ā́ ḥ
Instr.	s u kh é n a s u kh á y ā́	s u kh ā́ bh y ā́ m	s u kh ái ḥ s u kh ā́ bh í ḥ
Dat.	s u kh ā́ y a s u kh ā́ y ạ	s u kh ā́ bh y ā́ m	s u kh é bh y á ḥ s u kh ā́ bh y á ḥ
Abl.	s u kh ā́ t s u kh ā́ y ā H	s u kh ā́ bh y ā́ m	s u kh é bh y á ḥ s u kh ā́ bh y á ḥ
Gen.	s u kh á s y a s u kh ā́ y ā H	s u kh á y ó ḥ	s u kh ā́ n ā́ m
Loc.	s u kh é s u kh ā́ y ā m	s u kh á y ó ḥ	s u kh á s ú s u kh ā́ s ú

Table 4.2 shows the output after having applied the rules above to the lexical entries at the beginning of the chapter. This table is the current state of the derivation and would be the output if the remaining sandhi rules did not exist. We can see that the dual and plural of the weak cases for some of the cells have multiple accents. This situation is when the same accent-removing rules used by the r-stems comes into play. As a reminder, lines 250 through 254 recapitulate the accent-removing rules detailed in the previous chapter.

```
250 #sandhi $accented $abc* ā́ => $1 $2 ā .
251 #sandhi $accented $abc* á => $1 $2 a .
252 #sandhi $accented $abc* í => $1 $2 i .
253 #sandhi $accented $abc* ó => $1 $2 o .
254 #sandhi $accented $abc* ú => $1 $2 u .
```

These sandhi rules are responsible for removing any of the unwanted accent marks found after the leftmost accent, implementing the BAP from Chapter 1 [Kip10]. As a reminder of how the syntax works, the network looks for the sequence defined on the LHS and outputs the RHS for that sequence, either feeding it into the next sandhi rule or output the final form to the display. The rules in lines 250 through 254 are only separate in KATR as there is no simple way to remove the accent from an accented letter without listing each accented letter that might lost its accent.

After these rules come the various KATR clean-up rules responsible for replacing or removing any of the surviving relics and ensuring the output matches the correct orthographic conventions. These rules are again identical to the ones for the r-stems, so they will not be given explicitly in the guided derivation for the a-stems. The following table is the final output using all the rules.

Table 4.3: This table is the final output for the a-stems using PN. Just as with the previous tables, each cell contains up to three entries. The first line is the masculine, the second is the neuter, and the third is the feminine. When there are only two entries, the first is the masculine-neuter with the second being the feminine.

	Singular	**Dual**	**Plural**
Nom.	sukháḥ sukhám sukhā́	sukháu sukhé sukhé	sukhā́ḥ sukhā́ni sukhā́ḥ
Voc.	súkha súkham súkhe	súkhau súkhe súkhe	súkhāḥ súkhāni súkhāḥ
Acc.	sukhá m sukhā́m	sukháu sukhé sukhé	sukhā́n sukhā́ni sukhā́ḥ
Instr.	sukhéna sukháyā	sukhā́bhyām	sukháiḥ sukhā́bhiḥ
Dat.	sukhā́ya sukhā́yai	sukhā́bhyām	sukhébhyaḥ sukhā́bhyaḥ
Abl.	sukhā́t sukhā́yāḥ	sukhā́bhyām	sukhébhyaḥ sukhā́bhyaḥ
Gen.	sukhásya sukhā́yāḥ	sukháyoḥ	sukhā́nām
Loc.	sukhé sukhā́yām	sukháyoḥ	sukhásu sukhā́su

4.2 Compositional Derivation

Since the biggest difference between the two derivations is the information stored in the lexicon and the accent-assigning sandhi rules, the guided derivation will only explicitly state the differences.

```
255 HAPPY_M: %sukháḥ
256     <stem1> == s u kh á
257     <> = A-Masculine-Nouns
258 HAPPY_N: %sukhám
259     <stem1> == s u kh á
260     <> = A-Neuter-Nouns
261 HAPPY_F: %sukhā́
262     <stem1> == s u kh á
263     <> = A-Feminine-Nouns
```

Just as with the r-stems, the a-stems have rather the same lexical entries with only single stems. A-stems, being nominal with a static accent, are assumed to have their accent present in the lexicon as opposed to the dynamic r-stems. The nominal formation rules are roughly the same between the two networks except that this network lacks a specific formation process for the vocative and always states the stem as “<stem1> instead of “<stem1>, “<stem2>, and “<nympha>. The lack of a specific vocative rule is because there is no separate entry in the lexicon that would necessitate a separate process. The vocative can just use the Base-Noun rule where the ending is attached directly to the stem. Lets look at what the paradigm looks like after the compositional lexical entries go through the nominal formation nodes as seen in Table 4.4.

As we can see, the cells are beginning to look similar to the final output (ignoring the KATR relics), except with the vocative not having accents on the initial syllables and double accents in the weak dual and plural forms. Using the same sandhi rules for vowel combination from before, consecutive vowels are combined to form long vowels or new vowels.

Since the a-stems have a static accent except for the vocative, only one new rule is necessary: the vocative rule. This shape of the rule is identical to the rule described in the previous chapter when dealing with the r-stems. This was designed intentionally since there is a universal process where the accent will always move to the first syllable if not there already. To remind us, here is the shape of the vocative sandhi rule in KATR:

```
264 #sandhi wordStart $cons $cons* $vowel $abc* ∅ => wordStart $1 $2 $3 ’ $4 ∅ .
```

This rule does not remove any existing accents, so while the vocative forms in the r-stems were receiving their first accent, the a-stems are receiving a second accent. This is not a problem, however, as we have rules specifically for removing the unnecessary accents to the right of the desired accent (lines 250 through 254 above).

The final step would be to apply the KATR clean-up rules. Since these have been the same for both networks, I will skip explaining them. Let’s look at the final output form of the paradigm.

With this table, the guided derivations for both networks for both the r-stems and the a-stems are complete, representing the an athematic mobile accent paradigm and a static accent paradigm. The next chapter will begin analyzing the results. First counting the number of steps for both networks, then discussing what this

Table 4.4: This table is before morphophonological and clean-up sandhi rules have been applied. Just as with the previous table, each cell contains up to three entries. The first line is the masculine, the second is the neuter, and the third is the feminine. When there are only two entries, the first is the masculine-neuter with the second being the feminine.

	Singular	**Dual**	**Plural**
Nom.	s u kh á ḥ s u kh á m s u kh ā́	s u kh á ḁ s u kh á i s u kh á i	s u kh á a ḥ s u kh á a n i s u kh á a ḥ
Voc.	s u kh á ∅ s u kh á m s u kh á i ∅	s u kh á ḁ s u kh á i s u kh á i	s u kh á a ḥ s u kh á a n i s u kh á a ḥ
Acc.	s u kh á m s u kh á a m	s u kh á ḁ s u kh á i s u kh á i	s u kh á Á n s u kh á a n i s u kh á a ḥ
Instr.	s u kh á i n a s u kh á y ā́	s u kh á a bh y ā́ m	s u kh á ạ ḥ s u kh á a bh í ḥ
Dat.	s u kh á a y a s u kh á a y ạ	s u kh á a bh y ā́ m	s u kh á i bh y á ḥ s u kh á a bh y á ḥ
Abl.	s u kh á a t s u kh á a y ā H	s u kh á a bh y ā́ m	s u kh á i bh y á ḥ s u kh á a bh y á ḥ
Gen.	s u kh á s y a s u kh á a y ā H	s u kh á y ó ḥ	s u kh á n ā́ m
Loc.	s u kh á i s u kh á a y ā m	s u kh á y ó ḥ	s u kh á s ú s u kh á a s ú

means for what KATR can tell us about the elegance of the Paradigmatic Theory and Compositional Theory.

Table 4.5: This table is the final output for the a-stems using CN. Just as with the previous tables, each cell contains up to three entries. The first line is the masculine, the second is the neuter, and the third is the feminine. When there are only two entries, the first is the masculine-neuter with the second being the feminine.

	Singular	**Dual**	**Plural**
Nom.	sukháḥ sukhám sukhā́	sukháu sukhé sukhé	sukhā́ḥ sukhā́ni sukhā́ḥ
Voc.	súkha súkham súkhe	súkhau súkhe súkhe	súkhāḥ súkhāni súkhāḥ
Acc.	sukhám sukhā́m	sukháu sukhé sukhé	sukhā́n sukhā́ni sukhā́ḥ
Instr.	sukhéna sukháyā	sukhā́bhyām	sukháiḥ sukhā́bhiḥ
Dat.	sukhā́ya sukhā́yai	sukhā́bhyām	sukhébhyaḥ sukhā́bhyaḥ
Abl.	sukhā́t sukhā́yāḥ	sukhā́bhyām	sukhébhyaḥ sukhā́bhyaḥ
Gen.	sukhásya sukhā́yāḥ	sukháyoḥ	sukhā́nām
Loc.	sukhé sukhā́yām	sukháyoḥ	sukhásu sukhā́su

Chapter 5 Results and Conclusions

This chapter explains the results of the networks described in the previous chapters. Only one example for each is necessary to discuss general simplicity and elegance of a theory due to having an example of a nominal with a mobile accent and a nominal with a static accent. As a reminder, the r-stems represent the mobile accent type and the a-stems represent the static type.

When discussing the simplicity or elegance of a computational model, code coverage is a method often used. This method calculates how much of the code in a computational model or program is devoted to a particular function and how much of the code is being used for the function in question. Coverage is expressed as a ratio and will be referred as Code Coverage Ratio (CCR) in this thesis [Eri17].

For one network to be simpler or more elegant than another network, the total number of lines in the network is related to the number of lines necessary for the derivation(s) in question. I calculate the CCR by dividing all the lines used for each derivation(s) by all the lines of code in the network in question, multiplied by 100[35][Eri17]. Simplicity is not seen as a negative and in fact is a goal in the description of any computational model, as long as this simplicity is not achieved at the expense of accuracy [Fin16]. As a result, I consider simpler networks to be more elegant.

5.1 Results for Paradigmatic Network

Looking at the total number of lines for the Paradigmatic Network, we can see that there are 183 lines used by KATR for generating the r-stems and a-stems in the lexicon. The r-stems and the feminine a-stem each have 6 lines of code while the masculine and neuter a-stems have 5 lines each.

R-Stems

I have defined the number of lines used by KATR for a derivation as the number of unique lines in the KATR network necessary for the derivation. A line is considered unique if it has a unique line number. This means that lines that look the same but occupy different lines in the network are treated as unique lines while repetitions of a line through the use of paths do not constitute a unique line for the calculation. This distinction is illustrated below:

```
N-Gen2:
    gen pl == “<stem2>” n Endings
    .
R-Nouns:
    $strong == “<stem1>” Endings
    voc == “<nympha>” Endings
```

[35] I wanted to reflect the CCR as a percentage and not as a decimal or fraction.

Table 5.1: This table is a list of the number of lines devoted to the r-stem derivation for PN.

Sections of the Derivation	**Number of Lines**
Lexical Entry	6
Nominal Formation Rules	17
Endings	24
Sandhi Rules	22
Definitions	12
Total	81
CCR	81/183 = 44.3%

271 acc pl == “<stem2>” A-Masculine
272 gen pl = N-Gen2
273 loc sg == “<stem1>” Endings
274 <> = Base-Nouns2
275 .

Each of the lines above are unique lines as they each have their own separate line in the network. For most of the lines, they are visually unique; however, lines 267 and 275 are visually identical. For consistency and ease of counting, I have decided that they are considered unique because they have a function. They indicate the end of their respective nodes. This inclusion has been done for all the calculations so neither network is being biased for or against in this situation. Line 272 is a path that connects two nodes together. The first time a node is accessed, all the relevant lines are counted for the calculation and any subsequent accessing of a node is not counted, done for consistency and ease. Neither network is biased for or against in this situation.

The total number of lines of code used to generate the r-stem in Chapter 3 is 81 lines, shown in table 5.1 and broken down as follows. Six lines are from the lexical entry for *pitr̥-*. The nominal formation lines are the eight lines for the R-Nouns node, three lines from the A-Masculine node, and three lines for the N-Gen2 node, and three lines for the Base-Nouns2 node. For the A-Masculine node, only the node label, the relevant RHS (i.e. the accusative plural), and the closing period, included for consistency, are counted as these lines are the minimum number of lines for accessing the relevant information of this node. The 83 lines also includes 24 lines of the Endings node, the six clean-up sandhi rules, seven of the morphophonological sandhi rules, nine of the phonological rules, the eleven lines defining the various variables of the networks, and line defining the characters used for the morphophonemes as atoms instead of nodes, which would the default process.

When we divide this number by the total number of lines for the Paradigmatic Network, we get 44.3% of the Paradigmatic Network devoted to generating *pitr̥-*. This percentage is compared to the percentage for the r-stems using the Compositional Network below.

A-Stems

Continuing where the r-stems left off, table 5.2 shows the number of lines used for the CCR calculation.

Table 5.2: This table is a list of the number of lines devoted to the a-stem derivation for PN.

Sections of the Derivation	**Number of Lines**
Lexical Entry	16
Nominal Formation Rules	41
Endings	48
Sandhi Rules	22
Definitions	12
Total	139
CCR	76.0%

The a-stem section of the lexicon has three subsections: one for each gender. The masculine and neuter both have five lines and the feminine has six lines, totaling eleven lines that must be consulted by KATR to generate the full a-stem paradigm. The network also has to use a total of 41 lines for the nominal formation nodes: 21 lines for the A-Feminine-Nouns node, five for the A-Neuter-Nouns node, twelve for the A-Masculine-Nouns node, and three for the N-Gen1 node. The nodes for the endings contain a total of 48 lines: eight for the A-Feminine node, six for the A-Neuter node, ten for A-Masculine node, and 24 for the Endings node. The a-stem derivation contains a total of 16 lines for the various sandhi rules: six for the clean-up rules, eleven for the morphophonological rules, and five for the phonological rules. Finally the a-stem derivation requires the same number of lines for the variables and atom definitions: a total of 12 lines.

Adding up the number of lines necessary to generate the a-stem paradigm gives us a total of 139 lines for the Paradigmatic Network. When we divide this number by the total number of lines for this network, we get 76.0% of the Paradigmatic Network devoted to generating the a-stems.

Total for the Derivations

Table 5.3 shows how the final CCR for PN was calculated.

The r-stem and the a-stem have a total of 22 lines for the lexicon: six lines for the r-stem and the feminine a-stem and five lines for the masculine and neuter a-stems. There are a total of 55 lines for the nominal formation nodes: eight for the R-Nouns node, 21 lines for the A-Feminine-Nouns node, five for the A-Neuter-Nouns node, twelve for the A-Masculine-Nouns node, three for the N-Gen1 node, three for the N-Gen2 node, and three for the Base-Nouns2 node. The nodes for the endings contain a total of 48 lines: eight for the A-Feminine node, six for the A-Neuter node, ten for

Table 5.3: This table is a list of the number of lines devoted to the r- and a-stem derivations for PN.

Sections of the Derivation	**Number of Lines**
Lexical Entry	22
Nominal Formation Rules	55
Endings	48
Sandhi Rules	33
Definitions	12
Total	170
CCR	170/183 = 92.9%

A-Masculine node, and 24 for the Endings node. The r-stem and a-stem derivations contain a total of 33 lines for the various sandhi rules: six for the clean-up rules, 13 for the morphophonological rules, and 14 for the phonological rules. Finally the a-stem derivation requires the same number of lines for the variables and atom definitions: a total of 12 lines.

Adding up the number of lines necessary to generate the a-stem paradigm gives us a total of 170 lines for the Paradigmatic Network. When we divide this number by the total number of lines for this network, we get 92.9% of the Paradigmatic Network devoted to generating the r-stem and a-stems.

5.2 Results for Compositional Network

Now that the paradigmatic network has been described, we can look at the Compositional Network. Looking at the total number of lines for the Compositional Network, we can see that there are 178 lines used by KATR for generating the r-stems and a-stems. All nominals in this network have the same number of lines due to the lexical entries having the same structure. There are only four lines for each lexical entry.

R-Stems

Similar to the r-stems for PN, table 5.4 shows the number of lines used for the CCR calculation.

The total number of lines of code used for generated the r-stem in Chapter 3 is 84 lines, broken down as follows. Four of the lines are from the lexical entry for *pitr̥-*. The nominal formation rules have a total of twelve lines: five lines for the R-Nouns node, three lines for the N-Gen node, and four lines for the Base-Nouns node. The 84 lines also includes three lines from the A-Masculine node and 25 lines of the Endings node. The 32 lines for the various sandhi rules, the eleven lines defining the various variables of the networks, and line defining the characters used for the morphophonemes as atoms instead of nodes are also part of the 88 lines.

Table 5.4: This table is a list of the number of lines devoted to the r-stem derivation for CN.

Sections of the Derivation	**Number of Lines**
Lexical Entry	4
Nominal Formation Rules	12
Endings	28
Sandhi Rules	32
Definitions	12
Total	88
CCR	88/178 = 49.4%

When we divide this number by the total number of lines for the Compositional Network, we get 49.4% of the Compositional Network devoted to generating *pitṛ-*. This percentage is larger than the percentage for the Paradigmatic Network, despite containing more lines.

A-Stems

Continuing where the r-stems left off, table 5.5 shows the number of lines used for the CCR calculation.

Table 5.5: This table is a list of the number of lines devoted to the a-stem derivation for CN.

Sections of the Derivation	**Number of Lines**
Lexical Entry	12
Nominal Formation Rules	38
Endings	51
Sandhi Rules	21
Definitions	12
Total	134
CCR	134/178 = 75.3%

Since every lexical entry contains the same number of lines in the Compositional Network, there are a total of twelve lines for the lexical entries for the a-stems that must be consulted by KATR to generate the full a-stem paradigm. The network also has to use a total of 38 lines for the nominal formation nodes: 20 lines for the A-Feminine-Nouns node, four for the A-Neuter-Nouns node, eleven for the A-Masculine-Nouns node, and three for the N-Gen node. The nodes for the endings contain a total of 48 lines: seven for the A-Feminine node, nine for the A-Neuter node, ten for A-Masculine node, and 25 for the Endings node. The a-stem derivation contains a total of 21 lines for the various sandhi rules: seven for the clean-up rules and 14 for the morphophonological and the phonological rules. Finally the a-stem derivation requires the same number of lines for the variables and atom definitions: a total of 12 lines.

Adding up the number of lines necessary to generate the a-stem paradigm gives us a total of 134 lines for the Compositional Network. When we divide this number by the total number of lines for this network, we get 75.3% of the Compositional Network devoted to generating the a-stems. Again this percentage is larger than the percentage for the Paradigmatic Network, despite containing more lines.

Total for the Derivations

Table 5.6 shows how the final CCR for CN was calculated.

Table 5.6: This table is a list of the number of lines devoted to the r- and a-stem derivations for CN.

Sections of the Derivation	**Number of Lines**
Lexical Entry	16
Nominal Formation Rules	47
Endings	51
Sandhi Rules	40
Definitions	12
Total	166
CCR	166/178 = 93.3%

The r-stem and the a-stem have a total of 16 lines for the lexicon as each lexical entry has only 4 lines. There are a total of 47 lines for the nominal formation nodes: five for the R-Nouns node, 20 lines for the A-Feminine-Nouns node, four for the A-Neuter-Nouns node, eleven for the A-Masculine-Nouns node, three for the N-Gen node, and four for the Base-Nouns node. The nodes for the endings contain a total of 51 lines: seven for the A-Feminine node, nine for the A-Neuter node, ten for A-Masculine node, and 25 for the Endings node. The r-stem and a-stem derivations contain a total of 40 lines for the various sandhi rules: seven for the clean-up rules, 33 for the morphophonological and phonological rules. Finally the a-stem derivation requires the same number of lines for the variables and atom definitions: a total of 12 lines.

Adding up the number of lines necessary to generate the a-stem paradigm gives us a total of 166 lines for the Compositional Network. When we divide this number by the total number of lines for this network, we get 93.3% of the Compositional Network devoted to generating the r-stem and a-stems.

5.3 Conclusions About the Networks

Since the goal of this project was to look at the simplicity and elegance of computing an IE languages nominal formation process from the lexicon to the surface realization through the various phonological rules, the CCR of lines for a derivation to the total number of lines is crucial. The larger CCR, the more of the network is used for the derivation. With a final calculations for the paradigmatic network and the compositional network, the more elegant network appears to be the compositional network.

This comes as a surprise as KATR biases paradigmatic systems, requiring fewer phonological rules in theory to construct the surface forms. The difference came down to the total number of lines for networks. Since the Paradigmatic Network has more lines overall, the derivations would have to be larger to comprise a larger percentage of the overall network. Since the differences are so slight, I do not feel confident ruling that either network is truly more elegant computationally. I chose to count the number of lines in a particular way (e.g. considering only the unique lines and unique meaning that they had their own line in the network). If I were to have chosen a different way of counting the number of lines or defining elegance in a different way, it might be possible for the most elegant network to change. There is a fundamental level of complexity in the Paradigmatic Theory where the lexicon must include more information than the lexicon for the Compositional Theory. Any redefining of elegance would require handling the expanded lexicon as well.

5.4 Future Direction

In the future, I would like to refine the Compositional Network, reducing the phonological and morphophonological rules to the most abstract level possible. I would want to expand the network to include verbs and build in derivation processes. This expansion would allow the network to generate nominal forms not found in the lexicon, which would fully test both the Compositional Theory but also the network in general as the output forms would be wholly reliant on the way the network has been encoded.

I also want to expand this project to Proto-Indo-European directly, having a single network that can access different components based on the worldview of anyone accessing it. It would ask for a particular root, ask for their worldview, and the output the full paradigm including the complexity value of their worldview. This complexity value would be the number of steps the network needed to take to generate their paradigm, including the derivation steps if a verbal root were given but a nominal output were desired.

Appendix A: Paradigmatic KATR Network

This appendix contains the KATR network for the r-stems and the a-stems for the Paradigmatic Network.

%Language: Sanskrit
%Theory: Paradigmatic
%Author: Ryan McDonald

%KATR Variables
#vars $number: sg du pl .
#vars $case: nom voc acc ins dat abl gen loc .
#vars $strong: nom voc acc .

#hideInternal

#atom Á H .

%Phonological Variables
#vars $abc: a á e é i í o ó u ú ā ā́ ī ī́ ū ū́ ạ ḁ r̥ ŕ̥ r̥̄ k kh g gh ṅ c ch j jh ñ ṭ ṭh ḍ ḍh ṇ t th d dh n p ph b bh m y r l v ś ṣ s ḥ .
#vars $accented: á ā́ á¯ é í ī́ ó ú ū́ ạ́ ḁ́ ŕ̥ r̥̄́ .
#vars $vowel: a e i o u ā ī ū ạ ḁ .
#vars $short: a á e é i í o ó u ú .
#vars $cons: k kh g gh ṅ c ch j jh ñ ṭ ṭh ḍ ḍh ṇ t th d dh n p ph b bh m y r l v ś ṣ s ḥ .
#vars $ruki: i í ī ī́ u ú ū ū́ r r̥ ŕ̥ k .
#vars $nocor: a e i o u ā ī ū ai au c ch j jh ñ ṭ ṭh ḍ ḍh t th d dh n ś s .
#vars $retro: r r̥ r̥̄ ŕ̥ r̥̄́ ṣ .

%%%%% Phonology1 %%%%%
%%%%% Sanskrit Sandhi %%%%%
#sandhi a a => ā .
#sandhi a i => e .
#sandhi a => ạ .
#sandhi a => ḁ .
#sandhi $ruki s => $1 ṣ .
#sandhi $short n ā́ m => $1 ¯ n ā́ m .
#sandhi $cons ŕ̥ n ā́ m => $1 r̥̄ n ā́ m .
#sandhi á r ḥ => ā́ .
#sandhi ā́ r ḥ => ā́ .
#sandhi ŕ̥ H => ú ḥ .
#sandhi ŕ̥ Á => r̥̄́ .
#sandhi ŕ̥ $vowel => r $1 .

```
#sandhi r ∅ => ḥ .
#sandhi $retro $nocor* n $vowel => $1 $2 ṇ $3 .

%%%%% Phonology2 %%%%%
%%%%% Accent Sandhi %%%%%
#sandhi á ḁ => áu .
#sandhi á a => ā́ .
#sandhi ā́ a => ā́ .
#sandhi á Á => ā́ .
#sandhi á i => é .
#sandhi á ạ => ái .
#sandhi ŕ̥ $abc* ā́ => ŕ̥ $1 ā .
#sandhi r̥̄́ $abc* ā́ => r̥̄ $1 ā́ .
#sandhi $accented $abc* ā́ => $1 $2 ā .
#sandhi $accented $abc* á => $1 $2 a .
#sandhi $accented $abc* í => $1 $2 i .
#sandhi $accented $abc* ó => $1 $2 o .
#sandhi $accented $abc* ú => $1 $2 u .

%%%%% Phonology2 %%%%%
%%%%% Clean-Up Sandhi %%%%%
#sandhi ạ => a i .
#sandhi ḁ => a u .
#sandhi á⁻ => ā́ .
#sandhi ∅ => .
#sandhi H => ḥ .
#sandhi wordEnd => .

%%%%% Morphology %%%%%
%%%%% Base Endings %%%%%
Endings:
    {nom sg} == ḥ wordEnd
    {$strong du} == ḁ wordEnd
    {nom pl} == a ḥ wordEnd
    {voc sg} == ∅ wordEnd
    {voc pl} = <nom pl>
    {acc sg} == a m wordEnd
    {acc pl} == Á ḥ wordEnd
    {ins sg} == ā́ wordEnd
    {ins du} == bh y ā́ m wordEnd
    {ins pl} == bh í ḥ wordEnd
    {dat sg} == é wordEnd
    {dat du} = <ins du>
    {dat pl} == bh y á ḥ wordEnd
    {abl sg} == H wordEnd
```

```
    {abl du} = <ins du>
    {abl pl} = <dat pl>
    {gen sg} = <abl sg>
    {gen du} == ó ḥ wordEnd
    {gen pl} == ā́ m wordEnd
    {loc sg} == i wordEnd
    {loc du} = <gen du>
    {loc pl} == s ú wordEnd
    .

%%%%% Declension Specific Endings %%%%%
A-Masculine:
    {acc sg} == m wordEnd
    {acc pl} == Á n wordEnd
    {ins sg} == i n a wordEnd
    {ins pl} == ạ ḥ wordEnd
    {dat sg} == a y a wordEnd
    {abl sg} == a t wordEnd
    {gen sg} == s y a wordEnd
    <> = Endings
    .

A-Neuter:
    {$strong sg} = A-Masculine:<acc sg>
    {$strong du} == i wordEnd
    {$strong pl} == a n i wordEnd
    <> = A-Masculine
    .

A-Feminine:
    {nom sg} == a wordEnd
    {voc sg} == i ∅ wordEnd
    {voc du} = A-Neuter
    {dat sg} == ạ wordEnd
    {loc sg} == ā m wordEnd
    <> = Endings
    .

%%%%% Nominal Formations %%%%%
Base-Nouns2:
    <> == “<stem2>” Endings
    .

N-Gen1:
    {gen pl} == “<stem1>” n Endings
    .
```

```
N-Gen2:
    gen pl == “<stem2>” n Endings
    .

A-Masculine-Nouns:
    {voc} == “<nympha>” Endings
    {ins du} == “<stem1>” a Endings
    {dat du} = <ins du>
    {dat pl} == “<stem1>” i Endings
    {abl du} = <dat du>
    {abl pl} = <dat pl>
    {gen du} == “<stem1>” y Endings
    {gen pl} = N-Gen1
    {loc du} = <gen du>
    <> == “<stem1>” A-Masculine
    .

A-Neuter-Nouns:
    {voc} == “<nympha>” A-Neuter
    {$strong} == “<stem1>” A-Neuter
    <> = A-Masculine-Nouns
    .

A-Feminine-Nouns:
    {nom sg} == “<stem2>” A-Feminine
    {$strong du} = A-Neuter-Nouns
    {voc} == “<nympha>” A-Feminine
    {ins sg} == “<stem1>” y Endings
    {ins du} = A-Masculine-Nouns
    {ins pl} == “<stem2>” Endings
    {dat sg} == “<stem2>” y A-Feminine
    {dat du} = <ins du>
    {dat pl} == “<stem2>” Endings
    {abl sg} == “<stem2>” y Endings
    {abl du} = <dat du>
    {abl pl} = <dat pl>
    {gen sg} = <abl sg>
    {gen du} = A-Masculine-Nouns
    {gen pl} = A-Masculine-Nouns
    {loc sg} == “<stem2>” y A-Feminine
    {loc du} = <gen du>
    {loc pl} = “<stem2>” Endings
    <> == “<stem1>” Endings
    .

R-Nouns:
    {$strong} == “<stem1>” Endings
```

```
    {voc} == "<nympha>" Endings
    {acc pl} == "<stem2>" A-Masculine
    {gen pl} = N-Gen2
    {loc sg} == "<stem1>" Endings
    <> = Base-Nouns2
    .

%%%%% Lexicon %%%%%
HAPPY_M: %sukháḥ
    <stem1> == s u kh á
    <nympha> == s ú kh a
    <> == A-Masculine-Nouns
    .

HAPPY_N: %sukhám
    <stem1> == s u kh á
    <nympha> == s ú kh a
    <> == A-Neuter-Nouns
    .

HAPPY_F: %sukhā́
    <stem1> == s u kh á
    <stem2> == s u kh ā́
    <nympha> == s ú kh a
    <> == A-Feminine-Nouns
    .

LEADER: %netā
    <stem1> == n e t ā́ r
    <stem2> == n e t ŕ̥
    <nympha> == n é t ā r
    <> == R-Nouns
    .

FATHER: %pitā
    <stem1> == p i t á r
    <stem2> == p i t ŕ̥
    <nympha> == p ít a r
    <> == R-Nouns
    .

MOTHER: %mātā
    <stem1> == m ā t á r
    <stem2> == m ā t ŕ̥
    <nympha> == m ā́ t a r
    <> == R-Nouns
    .

#show <$case :: $number>
```

Appendix B: Compositional KATR Network

This appendix contains the KATR network for the r-stems and the a-stems for the Compositional Network.

%Language: Sanskrit
%Theory: Compositional
%Author: Ryan McDonald

%KATR Variables
#vars $number: sg du pl .
#vars $case: nom voc acc ins dat abl gen loc .
#vars $strong: nom voc acc .

#hideInternal

#atom Á H .

%Phonological Variables
#vars $abc: a á e é i í o ó u ú ā ā́ ī ī́ ū ū́ ạ ạ́ ḁ ḁ́ r̥ ŕ̥ r̥̄́ k kh g gh ṅ c ch j jh ñ ṭ ṭh ḍ ḍh ṇ t th d dh n p ph b bh m y r l v ś ṣ s ḥ .
#vars $accented: Á á ā́ á¯ é í ī́ ó ú ū́ ạ́ ḁ́ ŕ̥ r̥̄́ H .
#vars $vowel: a e i o u ā ī ū ạ ḁ Á á ā́ é í ī́ ó ú ū́ ạ́ ḁ́ ŕ̥ r̥̄ .
#vars $short: a á e é i í o ó u ú .
#vars $cons: k kh g gh ṅ c ch j jh ñ ṭ ṭh ḍ ḍh ṇ t th d dh n p ph b bh m y r l v ś ṣ s ḥ H .
#vars $ruki: i ī u ū r r̥ ŕ̥ k .
#vars $cor: a e i o u ā ī ū ai au c ch j jh ñ ṭ ṭh ḍ ḍh t th d dh n ś s .
#vars $retro: r r̥ r̥̄ ŕ̥ r̥̄́ ṭ ṭh ḍ ḍh ṇ ṣ .

%%%%% Phonology %%%%%
#sandhi t a r => t á r .
#sandhi t ā r => t ā́ r .
#sandhi á r $cons* $accented => ŕ̥ $1 $2 .
#sandhi ā́ r $cons* $accented => ŕ̥ $1 $2 .
#sandhi wordStart $cons $cons* $vowel $abc* ∅ => wordStart $1 $2 $3 ´ $4 ∅ .
#sandhi a ´ => á .
#sandhi ā ´ => ā́ .
#sandhi e ´ => é .
#sandhi i ´ => í .
#sandhi u ´ => ú .

#sandhi $ruki s => $1 ṣ .
#sandhi $short n ā́ m => $1 ¯ n ā́ m .
#sandhi $cons ŕ̥ n ā́ m => $1 r̥̄ n ā́ m .

```
#sandhi $retro $cor* n $vowel => $1 $2 ṇ $3 .
#sandhi $cons r $cons => $1 r̥ $2 .
#sandhi $cons r´ $cons => $1 ŕ̥ $2 .
#sandhi á r ḥ => ā́ .
#sandhi ā́ r ḥ => ā́ .
#sandhi ŕ̥ $vowel => r $1 .
#sandhi r Á => r̥̄́ .
#sandhi r ∅ wordEnd => ḥ ∅ .

#sandhi á ḁ => áu .
#sandhi á a => ā́ .
#sandhi á Á => ā́ .
#sandhi á i => é .
#sandhi á ạ => ái .
#sandhi ŕ̥ H => ú H .
#sandhi í $abc* á => í $1 a .
#sandhi ŕ̥ $abc* ā́ => ŕ̥ $1 ā .
#sandhi r̥̄́ $abc* ā́ => r̥̄ $1 ā́ .
#sandhi $accented $abc* ā́ => $1 $2 ā .
#sandhi $accented $abc* á => $1 $2 a .
#sandhi $accented $abc* é => $1 $2 e .
#sandhi $accented $abc* í => $1 $2 i .
#sandhi $accented $abc* ó => $1 $2 o .
#sandhi $accented $abc* ú => $1 $2 u .

#sandhi ạ => a i .
#sandhi ḁ => a u .
#sandhi H =>  .
#sandhi á⁻ => ā́ .
#sandhi ∅ => .
#sandhi wordEnd => .
#sandhi wordStart => .

%%%%% Morphology %%%%%
%%%%% Base Endings %%%%%
Endings:
    {nom sg} == ḥ wordEnd
    {$strong du} == ḁ wordEnd
    {nom pl} == a ḥ wordEnd
    {voc sg} == ∅ wordEnd
    {voc du} == ḁ ∅ wordEnd
    {voc pl} == a ḥ ∅ wordEnd
    {acc sg} == a m wordEnd
    {acc pl} == Á ḥ wordEnd
    {ins sg} == ā́ wordEnd
    {ins du} == bh y ā́ m wordEnd
```

```
    {ins pl} == bh í ḥ wordEnd
    {dat sg} == é wordEnd
    {dat du} = <ins du>
    {dat pl} == bh y á ḥ wordEnd
    {abl sg} == H wordEnd
    {abl du} = <ins du>
    {abl pl} = <dat pl>
    {gen sg} = <abl sg>
    {gen du} == ó ḥ wordEnd
    {gen pl} == ā́ m wordEnd
    {loc sg} == i wordEnd
    {loc du} = <gen du>
    {loc pl} == s ú wordEnd
    .

%%%%% Declension Specific Endings %%%%%
A-Masculine:
    {acc sg} == m wordEnd
    {acc pl} == Á n wordEnd
    {ins sg} == i n a wordEnd
    {ins pl} == ạ ḥ wordEnd
    {dat sg} == a y a wordEnd
    {abl sg} == a t wordEnd
    {gen sg} == s y a wordEnd
    <> == Endings
    .

A-Neuter:
    {voc sg} == m ∅ wordEnd
    {$strong sg} = A-Masculine:<acc sg>
    {voc du} == i ∅ wordEnd
    {$strong du} == i wordEnd
    {voc pl} == a n i ∅ wordEnd
    {$strong pl} == a n i wordEnd
    <> = A-Masculine
    .

A-Feminine:
    {nom sg} == a wordEnd
    {voc sg} == i ∅ wordEnd
    {dat sg} == ạ wordEnd
    {loc sg} ==  m wordEnd
    <> = Endings
    .

%%%%% Nominal Formations %%%%%
Base-Nouns:
```

```
    {gen pl} = N-Gen
    <> == "<stem1>" Endings
    .

N-Gen:
    {gen pl} == "<stem1>" n Endings
    .

A-Masculine-Nouns:
    {ins du} == "<stem1>" a Endings
    {dat du} = <ins du>
    {dat pl} == "<stem1>" i Endings
    {abl du} = <dat du>
    {abl pl} = <dat pl>
    {gen du} == "<stem1>" y Endings
    {gen pl} = N-Gen
    {loc du} = <gen du>
    <> == "<stem1>" A-Masculine
    .

A-Neuter-Nouns:
    {$strong} == "<stem1>" A-Neuter
    <> = A-Masculine-Nouns
    .

A-Feminine-Nouns:
    {nom sg} == "<stem1>" A-Feminine
    {$strong du} = A-Neuter-Nouns
    {ins sg} == "<stem1>" y Endings
    {ins du} = A-Masculine-Nouns
    {ins pl} == "<stem1>" a Endings
    {dat sg} == "<stem1>" ay A-Feminine
    {dat du} = <ins du>
    {dat pl} == "<stem1>" a Endings
    {abl sg} == "<stem1>" ay Endings
    {abl du} = <dat du>
    {abl pl} = <dat pl>
    {gen sg} = <abl sg>
    {gen du} = A-Masculine-Nouns
    {gen pl} = A-Masculine-Nouns
    {loc sg} == "<stem1>" ay A-Feminine
    {loc du} = <gen du>
    {loc pl} == "<stem1>" a Endings
    <> == "<stem1>" A-Feminine
    .

R-Nouns:
    {nom sg} == "<stem1>" Endings
```

```
    {acc pl} == “<stem1>” A-Masculine
    <> = Base-Nouns
    .

%%%%% Lexicon %%%%%
HAPPY_M: %sukháḥ
    <stem1> == wordStart s u kh á
    <> == A-Masculine-Nouns
    .

HAPPY_N: %sukhám
    <stem1> == wordStart s u kh á
    <> == A-Neuter-Nouns
    .

HAPPY_F: %sukhā́
    <stem1> == wordStart s u kh á
    <> == A-Feminine-Nouns
    .

LEADER: %netā
    <stem1> == wordStart n e t ā r
    <> == R-Nouns
    .

FATHER: %pitā
    <stem1> == wordStart p i t a r
    <> == R-Nouns
    .

MOTHER: %mātā
    <stem1> == wordStart m ā t a r
    <> == R-Nouns
    .

#show <$case :: $number>
```

Bibliography

[Sch72] Joachem Schindler. “Das Wurzelnomen im Arischen und Greichischen”. PhD thesis. Würzberg, 1972.

[Kip82] Paul Kiparsky. *Explanation in Phonology.* Publications in Language Sciences Vol. 4. De Gruyter, 1982.

[Kip84] Paul Kiparsky. “Lexical Phonology of Sanskrit Word Accent”. In: *Amrtādhāra: R.N. Dandekar Felicitation Volume.* Ed. by S. D. Joshi. Delhi: Ajanta Publications, 1984, pp. 201–210.

[May86] Manfred Mayrhofer. *Etymologisches Wörterbuch des Altindoarischen.* C. Winter, 1986.

[EG96] Roger Evans and Gerald Gazdar. “DATR: A language for lexical knowledge representation”. In: *Computational Linguistics* 22 (1996), pp. 167–216.

[Fin+02] Raphael Finkel et al. *KATR: A set-based extension of DATR.* Tech. rep. University of Kentucky Department of Computer Science, 2002.

[Whi02] William Dwight Whitney. *Sanskrit Grammar: Including both the Classical Language and the Older Dialects.* 19th ed. Cambridge, Massachusetts: Harvard University Press, 2002.

[MFM03] Michael Meier-Brugger, Matthias Fritz, and Manfred Mayrhofer. *Indo-European Linguistics.* De Gruyter, 2003.

[MA06] J. P. Mallory and D. Q. Adams. *The Oxford Introduction to Proto-Indo-European and the Proto-Indo-European World.* New York: Oxford University Press, 2006.

[Cla07] James Clackson. *Indo-European Linguistics: An Introduction.* Cambridge textbooks in linguistics. Cambridge University Press, 2007.

[Hay09] Bruce Hayes. *Introductory Phonology.* Blackwell textbooks in linguistics: 23. Wiley-Blackwell, 2009.

[For10] Benjamin W. Fortson. *Indo-European Language and Culture: An Introduction.* Wiley-Blackwel, 2010.

[Kip10] Paul Kiparsky. “Compositional vs. Paradigmatic Approaches to Accent and Ablaut”. In: *Proceedings of the 21st Annual UCLA Indo-European Conference.* Ed. by H. Craig Melchert Stephanie W. Jamison and Brent Vine. Bremen: Hempen, 2010, pp. 137–81.

[BH12] Dunstan Brown and Andrew Hippisley. *Network Morphology: A defaults-based theory of word structure.* Vol. 133. Cambridge, England: Cambridge University Press, 2012.

[SL12] Andrew Spencer and Ana R. Luis. *Clitics: An Introduction.* Cambridge University Press, 2012.

[Cam13] Lyle Campbell. *Historical Linguistics: an Introduction.* The MIT Press, 2013.

[Fin16] Raphael Finkel. "Computer-Based Tools for Word and Paradigm Computational Morphology". In: *Oxford Research Encyclopedia of Linguistics* (2016).

[Beg17] Gašper Beguš. "The RUKI-Rule in the Rigveda". PhD thesis. Universa v Ljubjuani, 2017.

[Eri17] Ulf Eriksson. *Measuring Code Quality with Test Coverage Metrics.* May 23, 2017. URL: `https://reqtest.com/testing-blog/test-coverage-metrics/`.

[Rup17] Antonia Ruppel. *The Cambridge introduction to Sanskrit.* Cambridge University Press, 2017.

[Byr18] Andrew Byrd. "The Phonology of Proto-Indo-European". In: *Handbook of Comparative and Historical Indo-European Linguistics*. Ed. by Jared Klein et al. Volume 3. De Gruyter Mouton, 2018, pp. 2056–79.

[LY18] Jesse Lundquist and Anthony D. Yates. "The Morphology of Proto-Indo-European". In: *Handbook of Comparative and Historical Indo-European Linguistics*. Ed. by Jared Klein et al. Volume 3. De Gruyter Mouton, 2018, pp. 2079–195.

[Uni19] Lehigh University. *Thousands of medieval manuscripts now online in full color through digitization project.* Nov. 26, 2019. URL: `https://phys.org/news/2019-11-thousands-medieval-manuscripts-online-full.html`.

Vita

Associate of Arts; General Arts and Sciences, Pellissippi State Community College
Bachelor of Arts; Linguistics and Classics, University of Tennessee, Knoxville
Master of Arts (expected); Linguistics, University of Kentucky

Ryan Mark McDonald