What is a word? Evidence from a computational approach to Navajo verbal morphology*

Sonya Bird
University of Arizona

The goal of this paper is to take a look at what the field of computational linguistics can tell us about the nature of words. I start with two assumptions: 1) all morphology can be captured using finite-state processing machines (Sproat, 1992), and 2) words can be defined in morphological terms. I show that a finite-state machine cannot handle the facts of Navajo verbal morphology in a satisfactory manner, and propose an alternative way of dealings with these facts, using a more powerful machine of the type used to model syntax. Based on a discussion of how best to handle the Navajo facts computationally, I conclude that Navajo verbs constitute a class of words that cannot be defined in strictly morphological terms, but rather involve syntax as well as morphology.

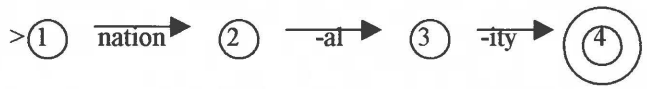
1 Outline

It is generally assumed that morphology can be dealt with using the simplest kind of processing machine: the finite state automaton. In this paper, I argue that dealing with Navajo verbs requires a more powerful machine than this, and therefore these verbs involve more than simply morphology. I start by introducing the basic characteristics of finite state automata (section 2), and of Navajo verbal morphology (section 3). In section 4 I focus on long-distance dependencies between elements of the Navajo verb, dependencies which cause problems for finite state machines. Section 5 offers a finite state account of these dependencies, and outlines its flaws. Section 6 offers an much more elegant account, using a more powerful machine. Finally, section 7 concludes by discussing the implications of the findings presented throughout the paper, in terms of the way in which Navajo words – and words in general – should be defined.

2 Finite state automata

Finite state automata (FSAs) are the simplest kinds of processing machines. They involve nodes, and transitions between these nodes (Allen, 1995; Hopcroft and Ullman, 1979). In morphological processing machines, morphemes are added during the transitions from node to node (Karttunen, 1983). Example (1) illustrates the mechanism of FSAs; the word *nationality* is generated by adding on its component morphemes one at a time, during the transitions between states 1-4. The nodes are represented by circles, and the transitions by arcs. The symbol > indicates the initial state; the double circles indicate the final state, i.e. the state in which the form produced is a well-formed word[1].

(1) FSA creating the word *nationality*



FSAs can either generate or recognize words given their morphological structure. In this paper, I refer only to their *generative* capacity.

One characteristic of FSAs that will be crucial to this paper is that they have no memory other than the path that they go down. The only way of keeping track of which elements have been generated is by

* This work was supported in part by the Social Sciences and Humanities Research Council of Canada, grant # 752-98-0274. Thanks to Michael Hammond, Terence Langendoen, and everyone at WSCLA V for useful discussion.

[1] It is possible to have more than one final state, as is the case in the FSAs discussed in the following sections.

looking back at the sequence of transitions that occurred during the generative process. This limitation will be discussed further in the following sections.

3 Navajo verbal morpholgy

The standard view of Navajo verbal morphology is that it is templatic (Hale, 1972; Kari, 1976; Sapir and Hoijer, 1967; Young and Morgan, 1987; Young, 2000). Verbs consist of templates, whose slots are filled by the verb preceded by various prefixal elements, both derivational and inflectional. A "generic" version of this template, based on work by aforementioned researchers, is provided in (2). The examples in (3) illustrate how this template accounts for the order of morphemes in the verbal complex.

(2) Navajo verbal template

0	1	2	3	4	5	6	7	8	9	10
objects of PostPs	a. PostPs b. Lexical Prefixes	iterative mode	distr. plural	direct object	a. areal b. deictic subject	Adverbial (lexical) Prefixes	Aspect	subject	class	Vstem

(3) Examples of the templatic approach to Navajo's verbal morphology[2]

	Prefix breakdown	*Verb form*	*English gloss*
a.	di-sh-ní 6-8-10	dishní	I say it
b.	da-bi-di-sh-ní 3 -4-6-8-10	dabidishní	I say it to them individually
c.	bí-náá-da-ho-ji-yi-lh-'aah 0- 1 - 3 -4 -5-7-9-10	bínáádahojiilh'aah	they (3+) are learning it again

To generate a form like (3)b, a FSA simply adds one morpheme at a time using a series of transitions:

(4) Example: FSA creating the word *dabidishní*

Although this mechanism may seem fairly straight-forward, a closer look at the data will show that FSAs quickly run into difficulties when dealing with Navajo. In the following section, I focus on one source of these difficulties: long-distance dependencies between various verbal morphemes.

4 Long-distance dependencies between various morphemes

The complexity of Navajo verbal morphology is due in part to the long-distance dependencies between various verbal morphemes, i.e. dependencies between non-contiguous morphemes. Example (5) illustrates such a dependency; the verb stem *nish* requires the atelic marker *na-*, and these two morphemes are separated by another one, the classifier *l-*[3]. The dependency is shown by the line linking the two relevant morphemes.

---

[2] Throughout this paper, the sequence *lh* is used for the voiceless lateral fricative. Nasalization on vowels is indicated by tilda (for example, nasalized *a* is written ã). The numbers refer to the position the morpheme holds in the verbal template.

[3] The following abbreviations are used in the examples throughout the text:

sg, dl:	singular, dual	I:	imperfective	semil:	semiliterative
1, 2:	first, second person	P:	perfective	rev:	reversionary
		class:	classifier	SRO:	solid round object (thematic prefix)

(5) Example of a long-distance dependency in Navajo

naa- l- nish
atelic-class -work
2- 9 10

naalnish
I am working

It is often the case that one verb form contains multiple dependencies. *Nested* dependencies are ones in which one dependency is embedded in another - see (6)a. *Crossed* dependencies are ones in which two dependencies overlap - see (6)b. In some verbs, there are both nested and crossed dependencies, as illustrated in (6)c.

(6) Multiple long-distance dependencies in Navajo

a. Nested dependencies

na- sooh- l- nish
atelic-P+2dl[4]-class-work(P)
2- 7+8- 9- 10

nishoolhnish
you (2) work (perfective)

b. Crossed dependencies

ch'í-ni- nish- t'aah
out- SRO-P+1sg-protrude
1- 6- 7+8- 10

ch'íninisht'aah
I slowly stuck my head out

c. Nested and crossed dependencies

ch'í-ná- náá- nish- d- zid
out- rev-semil-P+ 1sg- class- wake
1- 1- 1- 7+8- 9- 10

ch'ínánáánishdzid
I woke up again

These dependencies are ubiquitous in Navajo, as in other Athapaskan languages, and therefore must be dealt with by any computational model of verbal morphology. We turn next to how FSAs account for these dependencies.

5 FSAs and Navajo long-distance dependencies

As already mentioned, FSAs have no external memory; memory consists of the path that is followed when generating a particular output, i.e. which transitions and which nodes were used. What does this mean for long-distance dependencies? In order to capture them, it is necessary to have a separate branch for each string that involves a dependency. The result is that, for each dependency, all of the material that is between the two co-dependent morphemes must be duplicated. A FSA handling all the long-distance dependencies in Navajo will necessarily involve massive duplication, and this will be its downfall.

[4] I follow McDonough (1990) in considering the mode and subject prefixes merged into one portemanteau morpheme, containing information on both the aspect and the subject of the verb. I call these mode+subject complexes *conjugation patterns*.

As a concrete example, take the adjectival stative verbs in Navajo. These form 3 classes, the *di-* class, the *lhi-* class, and the *ni-* class:

(7) Example: adjectival stative verbs (3 classes)

	di-		**lhi-**		**ni-**	
a.	di-ts'id	tough	lhi-kizh	spotted	ni-tsaa	big
b.	di-jool	round	lhi-tso	yellow	ni-teel	wide
c.	di-jéé'	sticky	lhi-bá	gray	ni-daaz	heavy

These adjectival verbs can be conjugated by adding a person/number prefix between the adjectival prefix and the adjectival stem. The conjugation prefixes for the 3 classes are illustrated in (8). Notice that the prefixes for *di-* and *lhi-* classes are the same. I group these together and call them *Conj1*, to distinguish them from the *ni-* class conjugation prefixes, which I call *Conj2*. Some examples of conjugated adjectival verbs are found in (9).

(8) Conjugation prefixes

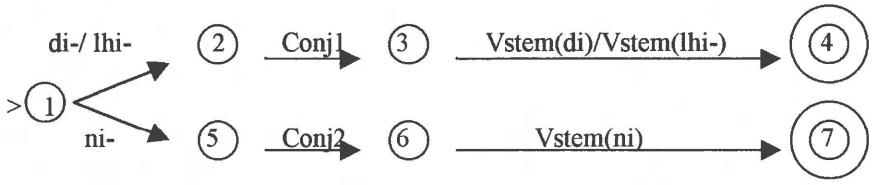
		Conj1		*Conj2*
	di-	=	**lhi-**	**ni-**
1st sg	nish-		nish-	sh-
2nd sg	ni-		ni-	í-
3rd sg/dl	∅		∅	∅
1st dl	nii-		nii-	ii-
2nd dl	noh-		noh-	oh-

(9) Some examples of conjugated adjectival verbs

a.	di-nish-ts'id	I am tough
b.	lhi-nish-kizh	I am spotted
c.	ni-sh-tsaa	I am heavy

Let us take a look at how these adjectival verbs can be generated using a FSA. The first machine to consider, presented in (10), is one that collapses the conjugation prefixes of the *di-* and *lhi-* classes. This machine is simple and elegant, but as we shall see, overgenerates.

(10) FSA for adjectival verbs without duplication (elegant but overgenerates)



Conj1 = {nish-, ni-, ∅, nii-, noh-}
Conj2 = {sh-, í-, ∅, ii-, oh-}
Vstem(di) = {ts'id, jool, jéé', ...}
Vstem(lhi) = {kizh, tso, bá, ...}
Vstem(ni) = {tsaa, teel, daaz, ...}

The problem with the FSA in (10) is that by collapsing the conjugation prefixes of the *di-* and *lhi-* classes, there is no way of ensuring that the adjectival stems will be generated with the appropriate adjectival prefixes. This is illustrated by the examples in (11).

(11) Possible outputs (overgeneration):

a. *grammatical forms generated*

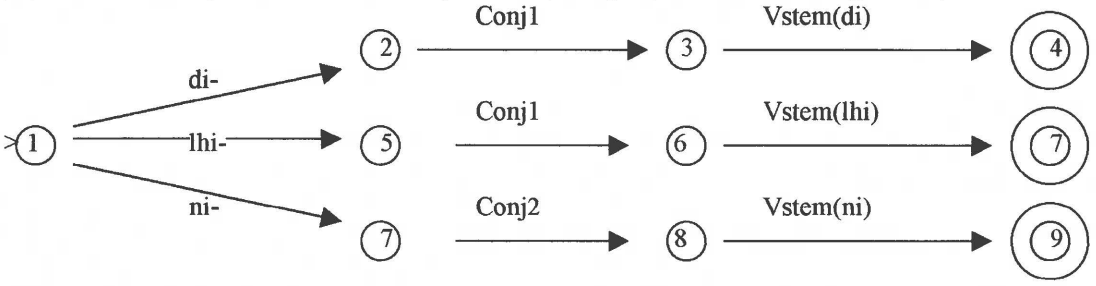
di-nish-ts'id	I am tough
lhi-ni-kizh	you are spotted

b. *ungrammatical forms generated*

*di-nish-kizh	*di-class* prefix with *lhi-class* verb
*lhi-ni-ts'id	*lhi-class* prefix with *di-class* verb

In order to avoid generating adjectival verbs like the ungrammatical ones in (11)b, a FSA must have entirely separate branches for *di-* and *lhi-* classes. The FSA in (12) includes these separate branches, and as a result involves a certain amount of duplication. Indeed, in (12), *Conj1* is found twice, once in the *di-* class branch, and a second time in the *lhi-* class branch.

(12) FSA for adjectival verbs with duplication (inelegant, but does not overgenerate)



Although the FSA in (12) is not as elegant as that in (10), it does not overgenerate. It may seem like a certain amount of duplication is a small price to pay for a machine that generates only well-formed words. However, this duplication quickly becomes a problem, as the following example illustrates.

A more complex example involves the dependencies which exist between the conjugation patterns (mode+subject prefixes) and the verb stem in active verbs. In Navajo, there are 18 possible conjugation patterns, depending on 1) the temporal information that is to be conveyed (tense, mood, aspect), 2) the particular verb stem, and 3) the particular prefixes[5]. Every time there is a dependency between a verb stem and a prefix that precedes the conjugation pattern, the conjugation pattern (which is between the two morphemes) must be duplicated. The result is that not only must there be separate branches for each of the 18 conjugation patterns, but also additional branches (duplications) for each dependency between a prefix preceding the conjugation pattern and a verb stem. In (13), two examples are presented in which the conjugation falls between two co-dependent morphemes. In (13)a, it falls between *ni-* and *-t'ąą́*; in (13)b, it falls between *na-* and *–nish* (as does the classifier *l-*).

(13) Examples (the relevant co-dependent morphemes are underlined)

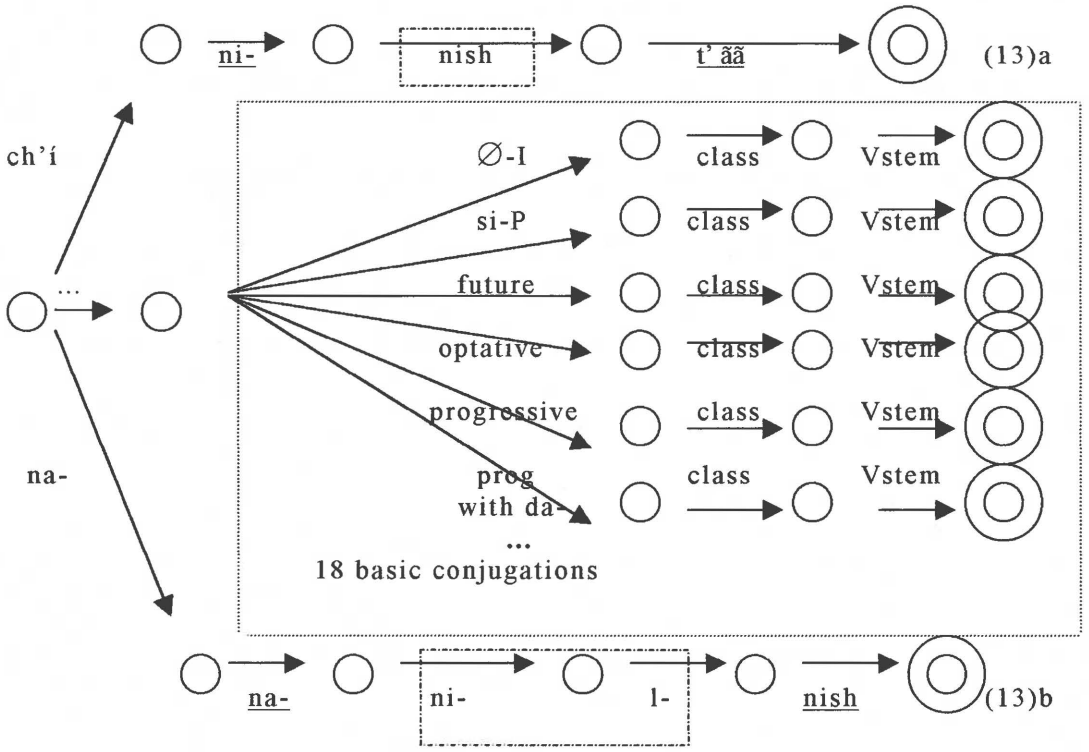
a. ch'í- <u>ni-</u> nish- <u>t'ąą́</u> — I slowly stuck my head out
out- SRO-P+1sg- protrude
1- 6- 7+8- 10

b. <u>na-</u> ni- l- <u>nish</u> — You are working
atelic-I+2sg-class-work (I)
2- 7+8- 9- 10

The FSA in (14) illustrates the number of branches required to account for the two forms in (13), as well as for the basic set of 18 conjugation patterns.

[5] For a complete list of the conjugation patterns, please see the appendix.

(14) FSA dealing with the dependencies in (13):



[ ] = duplicated material necessary to keep track of long-distance dependencies.

The FSA in (14) offers some idea of the complexity (in terms of amount of duplication and number of branches) that is involved in a finite-state approach to Navajo verbal morphology. The set of possible verb forms in Navajo is bound, therefore it is possible to create a FSA to deal with all the data. However, dealing with all of the long-distance dependencies would require the FSA to be extremely big and clunky.

6 Alternative: an augmented finite state automaton (AFSA)

Once again, the reason that a FSA has difficulty dealing with Navajo verbal morphology is because it lacks any external memory, which could keep track of long-distance dependencies in the verbal complex. A more appropriate machine for generating Navajo verbs is an augmented finite state automaton (AFSA). AFSAs are equivalent in generative power to FSAs, but they contain a memory that allows them to keep track of long-distance dependencies (Allen, 1995; Hopcroft and Ullman, 1979).

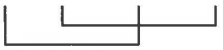
There are several variants in terms of how the memory is structured; the one presented here is a *random-access* (data-driven) memory[6]. Memory consists of an unordered set; elements (morphemes) stored in memory can be accessed as needed. This allows one to deal with both crossed and nested dependencies without referring to stacks and cues. The AFSA in (15)b generates the form in (15)a; the memory works as follows: in state 1, the memory set is empty. When *ch'i* is added, it is inserted into the memory, since it is involved in a dependency (it requires a particular conjugation in the perfective mode: the *ni-* perfective). Thus, in state 2, the memory set contains the element *ch'i*. In the transition to state 3, *ni-* is added, which is also entered into memory, since it is involved in a dependency with the verb stem. Once *nish-* is added in the transition to state 4, *ch'i* can be taken out of memory, since the dependency it is

[6] Terence Langendoen, personal communication.

involved in has been satisfied. Similarly, once the verb stem is added, *ni-* can be taken out of memory. In generating a well-formed verb, the memory set will always be null in the initial and the final states.

(15) Using random-access memory

a. form: ch'í- ni- nish- t'ãã *I slowly stuck my head out*



b. AFSA

Memory

state	*memory set*
1	∅
2	{ch'í}
3	{ch'í, ni}
4	{ni}
5	∅

There are two main advantages to using an auxiliary memory: 1) the massive duplication involved with FSAs is eliminated, and 2) the memory captures well the nature of the dependencies. Indeed, the AFSA makes it clear which elements are linguistically more closely related to one another, since they must be checked against one another in the memory set, in order to generate a well-formed word.

Another way to conceptualize the random-access memory is through the use of features. In (15)b above, the elements contained in the memory were the actual morphemes. An alternative is to store in the memory only certain relevant features, rather than entire morphemes. For example, for the form in (15)a, the following features could be used:

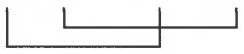
(16) Use of features

	morpheme	*features*
a.	lexical prefix ch'í (1)	[ni-perfective]
b.	ni-perfective (7)	[ni-perfective]
c.	Vstem t'ãã(10)	[SRO]
d.	adverbial prefix ni- (6)	[SRO]
e.	Vstem (10)	[...] particular conjugations (7)

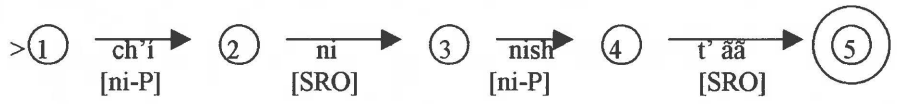
The AFSA in (17)b is identical to the one in (15)b, except that the memory consists of a set of features, rather than a set of morphemes.

(17) Memory and features

a. form: ch'í- ni- nish- t'ãã *I slowly stuck my head out*



b. AFSA



Memory

state	*memory set*
1	∅
2	{[ni-P]}
3	{[SRO], [ni-P]}
4	{[SRO]}
5	∅

The model outlined in (17) is reminiscent of the feature checking process involved in the Minimalist model of syntax, where well-formed sentences result from successful feature checking (Chomsky, 1997). The details of a morphological processing machine that uses a random-access memory set of the type outlined above still need to be worked out. One issue that I have glossed over how to tell whether a feature should be placed in memory for reference to it later on[7]. Perhaps the best solution would be to enter all features into memory, and mark those which must be checked before the end of the generative process.

The final example presented shows that AFSAs will not generate forms in which dependencies are not satisfied. In (18)a, the SRO prefix, required by the verb stem *t'ãã* has been omitted. When the AFSA arrives at the final state, there is nothing in the memory set against which to check the [SRO] feature on the verb stem. For this reason, the machine crashes, and fails to produce the faulty form.

(18) An ungrammatical example

a. form: *ch'í- -nish-t'ãã *The ni- prefix is missing*

b. AFSA

Memory

state	*memory set*
1	∅
2	{[ni-P]}
3	{[ni-P]}
4	∅
5	∅ → CRASH!

7 Conclusion - Implications for the structure of words in languages such as Navajo

In the preceding sections, we have seen that it is possible to create a FSA that will generate all possible verb forms (since the system is bound), but it will necessarily involve massive duplication. A much more efficient machine is an AFSA, which uses an auxiliary memory to keep track of long-distance dependencies. The random-access memory proposed here is an unordered set, which has as members features that are specified in the lexical entry of each morpheme.

The AFSA outlined above resembles machines used to capture syntactic facts of natural languages, which are more computationally complex than morphological facts. As mentioned above, its mechanism also resembles the one used in the Minimalist Program to produce well-formed sentences. Given that the machine used for creating Navajo verbs must be more powerful than one that can deal with morphology, it seems that Navajo verbs involve more than morphology – rather, they involve syntax as

[7] This would be necessary in the case of a right-to-left dependency, for example if the verb stem requires a particular prefix.
[8] This transition does not involve addition of any morphemes.

well. Thus, words – or at least Navajo verbs – should be defined syntactically as well as morphologically. Note that there is another possibility here: Navajo verbs in fact involve only morphology, and the claim that all morphology can be handled by finite-state machines is wrong. Certainly, there are other morphological phenomena which are problematic for finite-state accounts - for example, reduplication (Sproat, 1992). I have chosen here to assume that finite-state machines *can* handle all morphological processes, and that Navajo verbs involve more than morphology. I have done this because the alternate approach leads to serious implications for the field of computational linguistics, implications that I am not yet certain are well-founded. In any case, this issue certainly requires further attention.

Finally, the information used by the AFSA to generate verbs (features) tells us about the what information is likely to be included in lexical entries associated with words. Further research in this area will contribute valuable insight into the question of how words such as Navajo verbs are stored in the mind.

References

Allen, J. 1995. *Natural language understanding*, 2nd edition. Redwood City, CA: The Benjamin/Cummings Publishing Company, Inc.

Chomsky, N. 1997. *The Minimalist Program*. Cambridge, Massachusetts: The MIT Press.

Hale, K. 1972. *Letter to Jim Kari*, April 26 1972.

Hopcroft, J. and Ullman, J. 1979. *Introduction to Automata Theory, Languages, and Computation*. Reading, Massachusetts: Addison-Wesley Publishing Company.

Kari, J. M. 1976. *Navajo Verb Prefix Phonology*. New York, NY: Garland Publishing Inc.

Karttunen, L. 1983. "KIMMO: a general morphological processor". *Texas Linguistics Forum* 22, 165-186.

McDonough, J. 1990. *Topics in the Phonology and Morphology of Navajo Verbs*. Doctoral dissertation. University of Massachusetts.

Sapir, E. and Hoijer, H. 1967. "The phonology and morphology of the Navajo verb. *UCPL 50*. BerkleyCA.

Sproat, R. 1992. *Morphology and Computation*. MIT Press, Cambridge Massachusetts.

Young, R. W. 2000. *The Navajo Verb System – an Overview*. Albuquerque, NM: University of New Mexico Press.

Young, R.W. and Morgan, W. Sr. 1987. *The Navajo Language: A Grammar and Colloquial Dictionary*, 2nd edition. Albuquerque, NM: University of New Mexico Press.

Appendix - Eighteen possible conjugation patterns in Navajo verbs

a.	Imperfective		Number (total: 18)
	i)	∅-imperfective (yi-imperfective)	1
	ii)	ni-imperfective	2
	iii)	long-vowel imperfective	3
	iv)	si-imperfective	4
b.	Perfective		
	i)	yi-perfective	5, 6
	ii)	ni-perfective	7, 8
	iii)	long vowel perfective	9, 10
	iv)	si-perfective	11,12

Each perfective form is further subdivided into two forms that go with

1) ∅ and *lh* classifiers, and
2) d and l classifiers, for a total of 8 forms.

c.	Future		
	i)	regular future	13
	ii)	long vowel future	14
d.	Optative		
	i)	regular optative	15
	ii)	long vowe optative	16
e.	Progressive		
	i)	in combination with distributive plural da-	17
	ii)	not in combination with distributive plural da-	18