Multi-Verb Constructions - Parsing with a Deterministic Finite State Automaton

Kalyanamalini Sahoo
EFL University, Hyderabad
India
kalyana@efluniversity.ac.in

ABSTRACT: *This article discusses the parsing of multi-verb constructions in the Indic language Odia, in a Deterministic Finite State Automaton (DFA). It deals with two verb (V-v) sequences, and in the case of passivization, 3 verb sequences. These V-v sequences are formed by combining a main verb and a fully or partially bleached 'light' verb; where the main verb carries the lexical semantic information; the Tense, Aspect, Mode marking occurs on the light verb; and the sequence as a whole determines the argument structure. Although both the verbs in the V-v sequences are form-identical with a main verb in the language, the second one is a light verb because of its semantic and grammatical bleaching. The verbs cannot change their position, and are always spelled as a single word. Such double positional slots for the verbs pose constraint for the processing of the string by DFA, as it would be different from a standardly accepted single verb string. Moreover, both the verbs vary in their grammatical and semantic functions, and hence, choose a particular type of verb to co-occur with. This paper proposes the parsing of such multi-verb constructions in a DFA. Such morphological parsing is new in Odia and can be used in various applications like morphological analyzer, spell-checker, machine translation, information retrieval, etc.*

Keywords: Morphological parsing, Deterministic Finite State Automaton, Multiword expressions, Morphotactics

Abbreviations: DFA: Deterministic Finite state Automaton; FSA: Finite State Automaton; 1SG: 1st person singular; 3SG: 3rd person singular; AGR: agreement; ASP: aspect; AUX: auxiliary; CAUS: causative; CL: classifier; CM: conjunctive morpheme; FUT: future; HYP: hypothetical; INSTR: instrumental; NEG: negation; PASS: passive; PERF: perfective; PROG: progressive.

Received: 18 October 2015, Revised 20 November 2015, Accepted 5 December 2015

1.Introduction

Morphological analysis of words is a basic requirement for automatic language processing, and indispensable when dealing with agglutinative languages like Odia, an Indo-Aryan language spoken in the eastern part of India. As we know, for natural language processing some applications such as lemmatization, tagging, machine translation, information retrieval, phrase recognition, etc. require a detailed morpho-syntactic parsing of the whole word. In this context, considering Odia multi-verb constructions, this work proposes a model for designing a morphological parser which can provide lexical, morphological and syntactic information for each lexical unit in the analyzed verbal form. It draws out a finite-state machine that accepts valid sequences of morphemes in a verbal form and rejects invalid ones. We can use the DFA to solve the problem of morphological recognition; determining whether an input string of morphemes makes up a legitimate Odia word or not. We

do this by taking the DFA and plugging in each multi-verb sequences into it.

Considering morphological analysis, such a parser focuses on three main aspects, as proposed by (Ritchie, Pulman, Black & Russel 1992):

(1) i) Morphographemics: occurrence of orthographic variations while linking morphemes.

ii) Morphotactics: the co-occurrence restrictions of morphemes for the formation of valid words.

iii) Feature-combination: the way these morphemes can be grouped so that their morpho-syntactic features can be combined.

As a consequence of rich morphology of Odia we control morphotactic phenomena, as much as possible, in the morphological segmentation phase. Alternatively, a model with minimal morphotactic treatment, as noted by (Ritchie, Pulman, Black & Russel 1992) would produce too many possible analyses after segmentation, which could be rejected in a second phase. They do not adopt finite state mechanism to control morphotactic phenomena, and their two-level implementation incorporates a straightforward morphotactics, reducing the number of sub-lexicons to the indispensable (prefixes, lemmas and suffixes). This approximation would be highly inefficient for agglutinative languages like Odia, as it would create many nonsensical interpretations that would be rejected by the system. Therefore, we separate sequential morphotactics (i.e., which sequences of morphemes can or cannot combine with each other to form valid words), which will be recognized by means of continuation classes, and non-sequential morphotactics like long-distance dependencies that will be controlled by the word-grammar.

The remainder of this paper is organized as follows. Section 2 gives a brief description of the Odia verbal forms including multi-verb constructions. Section 3 discusses the syntactic and semantic aspects of multi-verb constructions. Section 4 describes the architecture for morphological processing, specifies the phenomena covered by the analyzer, explains its design criteria, and presents the processing details. Section 5 shows an evaluation of the DFA and ends with some concluding remarks.

2. Odia Verbal Forms

2.1 Single-verb Verbal Forms

Odia is a syntactically head-final and morphologically agglutinative language. A number of morphemes carrying different grammatical functions get affixed to the verbal root to make a verbal form. The major inflectional subsystems that cluster around the verb are: tense, aspect, agreement markers, negation markers, auxiliary morpheme etc. Considering single-verb verbal forms, Odia typically contains a verbal root followed by a sequence of morphemes, as in (2).

(2) *kar-i-paar-u-na-th-il-aa*
do-CM–Modal-ASP_{PROG}–NEG-AUX-$Tense_{PAST}$-$AGR_{3rd\ SG}$
'(S/he) was not able to do.'

In (2), the verbal root 'do' is followed by the conjunctive morpheme (CM), the ability modal morpheme, the progressive aspectual morpheme, the negative morpheme, the auxiliary morpheme, the past tense morpheme and the 3rd person singular morpheme.

Agreement distinguishes finite verbal forms from non-finite verbal forms in Odia, although tense has extended functions in both finite as well as nonfinite constructions. In a finite verbal form, the realization of the verbal root, Agr. and Tense (except for the present tense) is obligatory, while the realization of Asp, Aux, Modal or CM is optional. The sequence of items in a finite verbal form can be shown as follows (Sahoo 2001):

(3) Root-(CM)-(Modal)-(ASP)-(NEG)-(AUX)-Tense-AGR

Nayak (1987) also has proposed the same sequence of items but for CM and modal. The morphemes that occur in a finite verbal form can be listed as follows: there are two Aspect morphemes: *u* (Progressive) and *i* (Perfective); three Auxiliary morphemes: *achh* (Present), *th* (Past and Future), and *thaa* (Hypothetical); one modal morpheme: *paar*, one conjunctive morpheme: *i*. The Tense morphemes are realized as *il* (Past), *ib* (Future) and *ant* (Hypothetical). The present tense morpheme

is not lexically realized (Sahoo 2001; Nayak 1987). Agreement morphemes are marked for person, number and honorificity. In a finite clause, the Auxiliary and Aspect are interdependent morphemes, in the sense that, the Auxiliary morpheme cannot occur in the absence of the Aspect morpheme, and the vice versa. However, the Aspect morpheme can occur in the absence of an Auxiliary morpheme in a nonfinite construction. Similarly, the Modal morpheme is dependent on the CM, but the reverse is not true.

Participials (PRTP), gerundives (GER), conditionals (COND), infinitivals, telic affirmative affixes (Tel Aff) and conjunctive morphemes (CM), which lack agreement features are realized in non-finite verbal forms in Odia. So, the classification of verbal forms in the language can be shown as follows (Sahoo 2001):

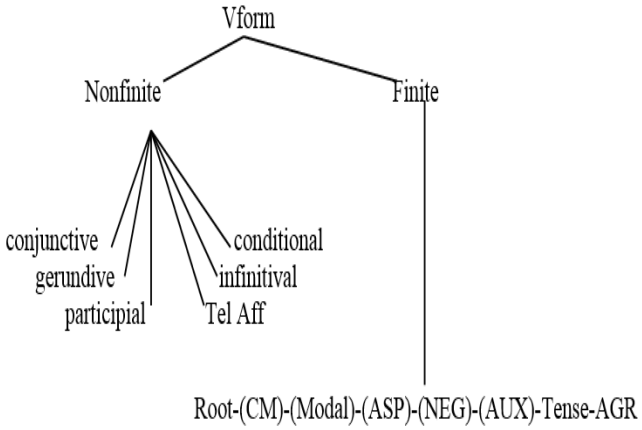

Figure 1. The classification of verbal forms in Odia

2.2 Two verb sequences

Two verb sequences (marked in bold in the examples (4)-(6)) are formed by combining a main verb and a fully or partially bleached light verb, where the main verb carries the lexical semantic information, and the Tense, Aspect and Mode (TAM) marking occurs on the light verb. The two verb sequence is considered as a single word. E.g.

(4) *kie glas-Taa* ***bhaang-i-de-i-ch-i***
somebody glass-the break-CM-give-PERF-AUX-3SG
'Somebody has broken the glass.'

(5) *se sandhyaa naheuNu* ***soi-(i)-paD-il-aa***
he evening NEG-happen sleep-(CM)-fall-PAST-3SG
'He slept off before evening.'

(6) *se raajaa-dwaaraa taara sabu kaama* ***kar-aa-i-ne-l-aa***
he Raja-by his all work do-CAUS-CM-take-PAST-3SG
'He got all his work done through Raja.'

Note that in the above examples, V_1 is usually the combination of a verbal root and a conjunctive suffix (usually *-i-*)[1], although in the presence of a CAUS morpheme (as in (6)), first the CAUS morpheme (*–aa–*) gets attached to the verbal root and then the CM morpheme gets attached to the V-CAUS sequence. Hence, in a way, the CM morpheme conjoins V_1 and V_2. The light verb (V_2) is combined with several morphemes carrying grammatical features like ASPect, AUXiliary, Tense, AGReement, etc. Interestingly, the CAUS morpheme can be added to the V_1 (not to V_2), as in (6), which adds an extra causative layer; in other words, the CAUS morpheme in (6) expresses the causation relationship between the subject (*se* 'he') and Raja (in an Instrument role) who did the actual work.

[1]*This affix 'i' can be compared with 'tvaa(ya)', or '(t)ya' / '(t)yaa' suffixes in Sanskrit, which were used to form gerundives. These suffixes were also sometimes referred to as conjunctive participles (Butt & Lahiri 1998).*

In a two verb sequence, both the verbs may carry contrastive lexical meaning, still can co-occur, in the case where the meaning of the light verb is completely bleached; but cannot co-occur in the case of partial bleaching of the light verb. E.g.

(7) *ghara bhitaraku aas-i-jaa*
house inside-to come-CONJ-go-IMP
'Come inside the house.'

(8) *bahiTaa eThaaru *ne-i-die / *de-i-nie / ne-i-jaa*
book-CL from here *take-CONJ-give-IMP / *give-CONJ-take-IMP / take-CONJ-go-IMP
'Take away the book from here.'

In (7), it is possible for the contrastive verbs 'come' and 'go' can co-occur, while this is not possible for 'take and 'give' as in example (8). This indicates that in example (7), the light verb is bleached completely, so that the meaning of 'go' is not involved in the sequence, as the overall meaning is 'come in', which is represented by the main verb. In example (8), the sequence 'take-go' denotes the meaning 'take away', so we can say that the meaning of 'go' is not completely bleached. Moreover, if we consider example (4), in the sequence 'break-give', as the light verb 'give' is bleached semantically, it does not carry any thematic role like 'Giver' , 'Receiver', or 'Gift'. This indicates that the light verbs are semantically bleached either completely or partially to certain extent. Such semantic bleaching of light verbs, not only differentiates them from regular main verbs in the language, but also indicates their grammaticalization status.

2.3 Types of Light Verbs

Types of light verbs that occur in Odia as second element in V-v constructions can be listed as follows (Sahoo 2012; Lemmens & Sahoo (to appear)):

(9) Motion verbs: *jaa* 'go', *aas* 'come', *chaal* 'walk', *paD* 'fall', *pakaa* 'drop', *uTh* 'rise'
Stative verbs: *bas* 'sit', *rah* 'stay'
Transfer verbs: *de* 'give', *ne* 'take'

Types of V-v sequences by using these light verbs are listed in the following tables.

List of motion verbs

ଯା *jaa* **'go'**	ଚାଲ *-chaal* **'walk'**	ପଡ-*paD* **'fall'**	ପକା-*pakaa* **'drop'**	ଉଠ-*uTh* **'rise'**	ଆସ-*aas* **'come'**
ଆସିଯା *aasi-jaa* /come-go/	ଦେଇଚାଲିଛି *dei-chaalichi* /give-walk/	ଶୋଇପଡିଲା *soi-paDilaa* /sleep-fall/	ହସିପକାଇଲା *hasi-pakaailaa* /laugh-drop/	ହସିଉଠିଲା *hasi-uThilaa* /laugh-rise/	ନେଇଆସିଲା *nei-aasilaa* /take-come/
ନେଇଯାଅ *nei-jaa* /Take-go/	ନେଇଚାଲିଛି *nei-chaalichi* /take-walk/	ଉଠି ପଡିଲା *uThi-paDilaa* /wake up-fall/	କୁଣ୍ଢାଇପକାଇଲା *kunDhaai-pakaailaa* /embrace-drop/	ଭାସିଉଠିଲା *Bhaasi -uThilaa* /float-rise/	ଦେଇଆସିଲା *dei-aasilaa* /give-come/
ଭାଙ୍ଗିଗଲା *bhaangi-galaa* /break-go/	ଖାଇଚାଲିଛି *khaai-chaalichi* /eat-walk/	ଭାଙ୍ଗିପଡିଲା *bhaangi-paDilaa* /break-fall/	ଧରିପକାଇଲା *Dhari-pakaailaa* /catch-drop/	ଜଳିଉଠିଲା *jaLi-uThilaa* /burn-rise/	ଭାସିଆସିଲା *bhaasi-aasilaa* /float-come/
ରହିଗଲା *rahi-galaa* /stay-go/	ପିଇଚାଲିଛି *pii-chaalichi* /drink-walk/	ବଳି ପଡିଲା *baLi-paDilaa* /leave-fall/	ଲଗେଇପକାଇଲା *lagei-pakaailaa* /apply-drop/	ଫୁଟିଉଠିଲା *phuTi-uThilaa* /blossom-rise/	ଲିଭିଆସିଲା *libhi-aasilaa* /extinguish-come/

Table 1. List of Motion Verbs

List of stative verbs

ବସ -*bas* **'sit'**	ରହ -*rah* **'stay'**
କହି-ବସିଲା *kahi-basilaa* /tell-sit/	ବସି-ରହିଛି *basi-rahichhi* /sit-stay/
କରି-ବସିଲା *kari-basilaa* /do-sit/	ଲାଗି-ରହିଛି *laagi-rahichi* /touch-stay/
ଖାଇ-ବସିଲା *khaai-basichi* /eat-sit/	ଶୋଇ-ରହିଛି *soi-rahichi* /sleep-stay/
ଧରି-ବସିଲା *dhari-basichi* /catch-sit/	ପଡି-ରହିଛି *paDi-rahichi* /fall-stay/

List of transfer verbs

ଦେ -*de* **'give'**	ନେ -*ne* **'take'**
ଦେଇଦେଲା *dei-delaa* /give-give/	କରିନେଲା *kari-nelaa* /do-take/
ଆଣିଦେଲା *aaNi-delaa* /bring-give/	ଖାଇନେଲା *khaai-nelaa* /eat-take/
କରିଦେଲା *kari-delaa* /do-give/	ଚୋରାଇନେଲା *choraai-nelaa* /steal-take/
ପକାଇଦେଲା *pakaai-delaa* /drop-give/	କରାଇନେଲା *karaai-nelaa* /do.CAUS-take/

Table 2. List of Stative & Transfer Verbs

Passivized two verb sequence resulting in 3 verb sequences

ଦେଇଦେଲା →	ଦେଇଦିଆଗଲା
dei-delaa /give-give/	*dei-diaa-galaa* /give-give-go/
କରିନେଲା →	କରିନିଆଗଲା
kari-nelaa /do-take/	*kari-niaa-galaa* /do-take-go/
ଭସାଇଦେଲା →	ଭସାଇ ଦିଆଗଲା
bhasaai-delaa /float.CAUS-give/	*bhasaai-diaa-galaa* /float.CAUS-give-go/
ଲିଭାଇଦେଲା →	ଲିଭାଇ ଦିଆଗଲା
libhaai-delaa /extinguish-give/	*libhaai-diaa-galaa* / extinguish -give-go/

Table 3. Passivized two verb sequences resulting in 3 verb sequences

3. Syntactic and Semantic Features

3.1 Transitivity

Considering the transitivity aspect of the two verb sequences, question arises as the light verbs are semantically bleached and no longer assign thematic roles, can we still consider them as transitive or intransitive as their form-identical lexical counterpart.

The answer is yes, because generally the choice of the light verb for co-occurrence with a main verb is determined by transitivity constraints. This is shown in example (10).

(10) a. Sequence of intransitive verbs (includes alternation verbs used intransitively):

$[V_{intrans} - v_{intrans}]$ *$[V_{trans} - v_{intrans}]$

-jaa '-go' : come-go, die-go *kill-go

-paD '-fall': slip-fall, sleep-fall

b. Sequence of transitive or ditransitive verbs:

$[V_{trans} - v_{trans}]$		$*[V_{intrans} - v_{trans}]$
-de '-give'	: break-give, kill-give	*die-give
-pakaa '-drop'	: embrace-drop, give-drop	

3.2 Causatives and Passives

Causative and passive morphemes play a major role in multi-verb constructions. Passivization, which is done by using the passive morpheme *–aa* and the passive auxiliary *jaa* 'go', transforms the two verb sequence into a three verb sequence. Causatives play a role from the point of transitivity. The causative morpheme can get attached to an alternating verb and hence, can co-occur with a transitive light verb.

A single verb can be causativized in the following way, where the causative morpheme occurs getting attached to the verbal root:

(11) *mũ lahuNi taraL-**aa**-il-i*
I butter melt.CAUS-1SG
'I melted the butter.'

The structure of causation of a V-v sequence can be shown as follows:

(12) $[[V + CAUS] - [v_{trans}]]$
$[[V_{alternating} + CAUS] - [v_{trans}]]$

The structure can be illustrated in the following example:

(13) *mũ lahuNi taraL-**aa**-i-de-l-i*
I butter melt-CAUS-CM-give-PAST-1SG
'I made the butter melt.'

Passivization of a verbal form is done by using the passive morpheme *–aa* (which is form-identical with the causative morpheme *-aa*) and the passive auxiliary *jaa* 'go'. Passivization of a single verb sequence can be done as the following:

(14) *(mo-dwaara) lahuNi taraL-**aa-ga-l-aa***
(I-INSTR) butter melt-PASS-go-PAST-3SG
'The butter was melted (by me).'

Please note that in the above example (14), the second verb ***ga-l-aa*** 'went' is not a light verb, it is the passive auxiliary 'go'. Passivization of a V-v sequence follows the following structure as in (15) and this can be illustrated in the example as in (16):

(15) $*[V_{intrans} - v_{intrans}]$
$[V_{trans} - v_{trans}]$
$[[V_{alternating} + CAUS] - [v_{trans}]]$

(16) *(mo-dwaara) lahuNi taraL-**aa**-i-**di**-**aa**-**ga**-**l**-**aa***
(I-INSTR) butter melt-CAUS-CM-give-PASS-PAST-3SG
'The butter was melted (by me).'

3.3 Completive Meaning

The light verb construction indicates the completion of event/action, whereas in the single verb construction the completion of the action is not in focus or presupposed.

(17) *tu* *ebe* *khaa-i-**di**-e,* *mũ* *pare* *khaa-ib-i*
you$_{[-Hon]}$ now eat-CM-give-3SG, I later eat-FUT-1SG
'You eat (complete eating) now, I will eat later.'

Note that also future tense marking is possible, preserving the completive meaning:

(18) *tu* *paLaa,* *mũ* *bahi-Taa library-re pheraa-i-de-**b**-i*
you leave-IMP, I book-CL library-PP return-CM-give-FUT-1SG
'You leave, I will return the book in the library.'

The light verb can co-occur with a perfective morpheme as well as with a progressive morpheme. This is shown in the examples (19) and (20), respectively.

(19) *mũ bahi-Taa* *aaN-i-kar-i* *tumaku* *de-i-de-i-**th**-il-i*
I book-CL bring.CM-do-PERF you.ACC give-CM-give-PERF-AUX-PAST-1SG
'(Me) having brought the book, I gave (it) to you.'

(20) *chakleT rakh-u* *rakh-u-ta* *tu* *sabu* *khaa-i-de-**u**-chh-u*
chocolate keep-PROG keep-PROG-EMP you all eat-CM-give-PROG-AUX-2SG
['Chocolate keeping keeping, you are eating up all (the chocolates).']
'While keeping the chocolates, you are eating up all of them.'

As illustrated in (21), the light verb constructions cannot have negation. (21a) shows the negation of a single verb construction, while (21b) shows the negation of a two verb sequence. E.g.

(21) a. *bahi-Taa mũ* *taaku* *de-l-i,* *kintu* *se* *ne-l-aa-**ni***
book-CL I him give-PAST-1SG but he take-PAST-3SG-NEG
'I gave him the book, but he didn't take it.'
[= 'I offered it to him, but he didn't take it.']

b. **bahi-Taa* *mũ taaku de-i-de-l-i,* *kintu* *se* *ne-l-aa-**ni***
book-CL I him give-CM-give PAST-1SG but he take-PAST-3SG-NEG
'I gave him the book, but he didn't take it.'
[means, the 'giving' act is completed, but as the receiver didn't take it...]

Note that there is a completive meaning associated with the light verb in (21b), which necessarily denotes the completion of the event and thus, does not allow the event to be shown 'undone', hence, not prone to negation. So, these light verbs give a completive meaning of the event/action.

Summarizing, we can say that the light verbs differ from the form-identical main verbs in grammatical function and semantic content. Unlike the regular main verbs, the semantic content of light verbs are bleached either completely or partially. Hence, they may not have their thematic roles. The multi-verb constructions are not prone to negation as they denote a completive meaning of the action/event.

In the following section, we will show the kinds of morphological knowledge that needs to be represented to produce a well-formed multi-verb construction in Odia. For this purpose, we choose a DFA for the computation of verbal forms.

4. A Deterministic Finite State Automaton

Since we cannot list every word in the language, computational lexicons are structured as a list of stems and affixes with a representation of the morphotactics. One way to model morphotactics is the finite-state automaton. We use a DFA to solve the problem of morphological recognition. It will determine whether an input string of morphemes makes up a legitimate

Odia verbal form or not. Such identification of sequences has a number of practical applications like spell checker, machine translation, information retrieval, etc.

4.1 The Machinery

As popular in the literature (Roche & Schabes 1997; Jurafsky & Martin 2000; Hanneforth 2011; Moore 2014) a deterministic Finite State Automaton is a device that receives a string of symbols as input, reads the string one symbol at a time from left to right, and after reading the last symbol halts and indicates either acceptance or rejection of the input. The automaton performs computation by reacting on a class of inputs (on strings or sequences of symbols). The concept of a *state* is the central notion of an automaton. A state of an automaton is analogous to the arrangement of bits in the memory banks and registers of an actual computer. Here, we consider a state as a characteristic of an automaton which changes during the course of a computation and which serves to determine the relationship between inputs and outputs. For our automaton, the *memory* consists simply of the states themselves. The computations of a DFA are directed by a 'program', which is a finite state of instructions for changing from state to state as the automaton reads input symbols. Given an input, the computation begins in a designated state, the *initial state*. After reading the input, the automaton either accepts or rejects it after some finite amount of computation. Given the current state and the symbol, it has only one choice of state to move to. As such a DFA has no memory to store information except for its current state, and it cannot return to earlier states in the string either.

In a more formal way, a deterministic finite state automaton can be defined as follows (Roche & Schabes 1997; Jurafsky & Martin 2000).

(22) A (deterministic) finite-state automaton is a quintuple $(Q, \Sigma, q_0, F, \delta)$ where

- Q is a finite set of N states $q_0, q_1, \ldots, q_n$

- Σ is a finite input alphabet of symbols

- $q_0 \in$ Q is the initial state

- $F \subseteq$ Q, the set of final states

- $\delta(q, i)$ is the transition function or transition matrix between states. Given a state $q \in Q$ and an input symbol $i \in \Sigma$, $\delta(q, i)$ returns a new state $q' \in$ Q. δ is thus a relation from $Q \times \Sigma$ to Q.

Thus, if the automaton is in a state $q \in Q$ and the symbol read from the input is a, then $d(q, a)$ uniquely determines the state to which the automaton passes. This property entails high run-time efficiency, since the time it takes to recognize a string is linearly proportional to its length. The FSA is called deterministic as it defines at most one transition for each state and each input symbol. Formally, it is called deterministic if for all symbol state pairs $q, a, | \delta(q, a) | \leq 1$ (Hanneforth 2011).

Succinctly, if the Machine configuration is like (23):

(23) $[q, \omega]$ wherre $q \in Q, \omega \in \Sigma^*$

Then it would yield relations like the following:

(24) $[q, a\omega] \mapsto^*_M [\delta(q, a), \omega]$

And the language could be represented as the following:

(25) $\{\omega \in \Sigma^* \mid \underbrace{[q_0, \omega] \mapsto^*_M [q, \lambda] \Lambda\ q \in F}\ \}$

4.2 The DFA for Odia

This section discusses how the DFA can be conceived as applying to Odia verbal forms. Figure 2 shows the processing of multi-verb verbal forms in Odia by using a DFA.

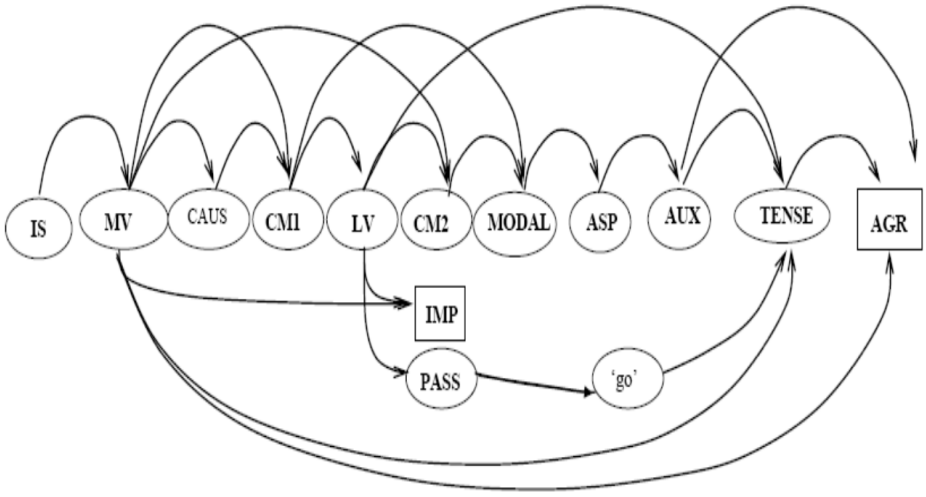

Figure 2. The DFA for multi-verb verbal forms in Odia

The automaton is represented as a directed graph: a finite set of vertices (nodes), together with a set of directed links between pairs of vertices called arcs. Each node corresponds to a state. States are represented as circles with name tags in them. Arcs are represented by arrows going from one state to another state. The final states are represented by rectangles. The machine starts at the initial state, runs through a sequence of states by computing a morpheme in each transition, and ends in the final state. The path moves from the initial point on the left to the final point on the right, proceeding in the direction of arrows. Once the arrow moves one step, there is no backward movement (Of course, recursion of an item can be shown by using closed loops). Each state through which the speaker passes represents the grammatical restrictions that limit the choice of the next morpheme. The resulting DFA is deterministic in the sense that given an input symbol and a current state, a unique next state is determined.

It starts at the initial state (Q_0), checks the next morpheme of the input. If it matches the symbol on an arc leaving the current state, then it crosses that arc, and moves to the next state, and thus, advances one symbol in the input. Such a process gets iterated until the machine reaches the final state, successfully recognizing all the morphemes in the input string. But if the machine gets some input that does not match an arc, then it gets stuck there and never gets to the final state. This is considered as the DFA/machine rejecting or failing to accept an input.

For multi-verb constructions (cf. Figure 2), the DFA starts at the initial state (Q_0). From the initial state it can choose the MV state directly. From the MV state, it has various options to move to the next state: it can move to the CAUS state, or directly to the CM_1, ASP, Tense, AGR or IMP state, out of which IMP and AGR are final states. From the CAUS state it moves to the CM_1 state. From CM_1 state it can move either to LV state or to the Modal state. Note that the CM_1 state is the only anchor for distinguishing main verb from the light verb, the CM_1 always immediately precedes the LV, while it follows the MV. From the LV state, it has 4 choices to move forward: CM_2, PASS, Tense or IMP, which is a final state. From the CM_2 state, it moves in a string to → Modal → ASP → AUX →Tense → AGR. From the AUX state also, it can move to AGR. From the PASS state, it goes to the 'go' verb state, which is constant for passivization, and from there, it can move to the Tense state and from there to AGR, which is a final state. Likewise, the DFA processes the verbal forms until it reaches the final state.

For implementation, we can test how a multi-verbal form in the language be processed by this machine. Take an example like (26)[2]:

(26) *khaa-i-de-l-aa*
eat-CM-give-PAST-3SG
'ate (it) up'

[2] *Please note that in the example (26), the morpheme -l- is the elided form of the PAST tense morpheme –il. So in the V-v sequence although the input morphemes are khaa-i-de-il-aa , the sequence is pronounced as khaa-i-de-l-aa.*

This verbal form can be processed in a deterministic finite state automaton (DFA) as in (27). The verbal form *khaa-i-de-l-aa* has 5 states: state 0 is the initial state and state 5 is the final state. It also has 5 transitions.

(27) $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$

$\Sigma = \{khaa, i, de,il, aa\}$

Q_0= the initial state (IS)

$F = \{q_5\}$

$\delta(q, i)$ can be defined by the transition table as follows:

	Input				
State	*khaa*	*i*	*de*	*il*	*aa*
0	1	Ø	Ø	Ø	Ø
1	Ø	2	Ø	Ø	Ø
2	Ø	Ø	3	Ø	Ø
3	Ø	Ø	Ø	4	Ø
4	Ø	Ø	Ø	Ø	5
5:	Ø	Ø	Ø	Ø	Ø

Table 4. The state transition table for the DFA for *khaa-i-de-il-aa*

In the transition table 4, state5 is marked with a colon to indicate that it is a final state. Ø indicates an illegal or missing transition. It can be read as follows: "*if we are in state 0 and we see the input khaa, we must go to the state 1. If we are in state 0 and we see the input i, de, il or aa we fail*."

Similarly, the DFA computes the verbal forms in a multi-verb construction. As figure 2 shows, the machine starts at the initial state and proceeds in the direction indicated by the arrows, computing the verbal forms successfully.

5. Evaluation

5.1 Corpus Data Analysis

To estimate the performance of the parser, we carry out a direct evaluation of it. The EMILLE/CIIL Corpus (ELRA-W0037) is consulted for the data. No annotation of this corpora is available as such. We used the morphological parser for the analysis of 11079 V-v sequences. We manually measured the accuracy of the morphological parser by counting the number of correctly analyzed sequences out of the total number of sequences. In the cases, where multiple analyses of any sequence were available, we accepted it only when all the correct analyses were present. For an appropriate evaluation of the parser, for the experiment, we added most of the roots used in the corpus to the lexicon, and implemented the rules accordingly.

Out of the 11079 V-v sequences, 10567 (95.378%) were found to be correctly analyzed. Of the remaining 512 words, 205 words could not be recognized by the parser and 307 words were assigned the incorrect or insufficient analyses. By taking a closer look at these unparsed 512 words, we could come up with the causes of recognition failure as listed in table 5, and the causes of insufficient analyses as listed in table 6.

Cause	Number of sequences
Inadequate rules	34 (16.58%)
Not available in the lexicon	46 (22.44%)
Passivized single V constructions resulting in 2 verb sequences	56 (27.31%)
Fusion (*sandhi*)[3]	37 (18.04%)
Irregular forms needing further investigation	32 (15.60%)

Table 5. Causes of recognition failure (205 sequences)

Cause	Number of sequences
Choice of pattern matching light verbs	34 (11.07%)
Alternating verbs	62 (20.19%)
Ambitransitive verbal forms	31 (10.09%)
Fusion (*sandhi*)	29 (9.44%)
Over-generation of identical V-v pairs	66 (21.49%)
Written vs spoken form	42 (13.68%)
Others (multiple analyses, 3 verb sequences, etc.)	43 (14.00%)

Table 6. Causes of insufficient analyses (307 sequences)

5.2 Conclusion

We specify the co-occurrence restrictions of both the verbs (Main verb and the light verb) in a verbal form and use the DFA to solve the problem of morphological recognition; determining whether an input string of morphemes makes up an acceptable Odia word or not. Such a morphological parser will help us to build a computational lexicon structured as a list of stems and affixes with a representation of the morphotactics and also can be used for designing a morpho-syntactic analysis for each word in unrestricted Odia texts. The design of the DFA we propose is new for Odia, as far as we know. We think that this design could be interesting for the parsing of V-v sequences in Odia as well as in other languages having similar multi-verb constructions.

Acknowledgement

Part of this work is carried out during a research stay (sabbatical) at the Center for Grammar, Cognition and Typology, University of Antwerp, which is hereby gratefully acknowledged. I am indebted to the 3S Infosolutions, Bhubaneswar for funding for this work. I would also like to express my sincere thanks to the participants of CLIN25 conference and the anonymous reviewers of this article for their valuable comments and suggestions.

References

[1] Butt, M., Lahiri, A. (1998). The status of light verbs in historical change. Ms. Universität Konstanz.

[3] The phonological processes that occur at morpheme or word boundaries.

[2] Hanneforth, T. (2011). 'A Practical Algorithm for Intersecting Weighted Context-free Grammars with Finite-State Automata'. *Proceedings of the 9th International Workshop on Finite State Methods and Natural Language Processing*, pages 57–64, Blois (France), July 12-15, 2011. Association for Computational Linguistics.

[3] Jurafsky, D., Martin, J.H. (2000). Speech and Language Processing: An Introduction to Natural Language Processing, *Computational Linguistics, and Speech Recognition*. New Jersey: Prentice Hall.

[4] Lemmens, M., Sahoo, K. (to appear). 'Something's gotta go, something's gotta give: Completion, mirativity and transitivity in Odia light verb constructions', *The Studia Linguistica: A Journal of General Linguistics*.

[5] Moore, C. (2014). Automata, languages, and grammars. *Lecture Notes*. July 23.

[6] Nayak, R. (1987). Non-finite clauses in Oriya. Doctoral dissertation, CIEFL, Hyderabad. India.

[7] Ritchie, G., Pulman, S.G., Black, A.W., Russel, G. J. (1992). Computational Morphology: Practical Mechanisms for the English Lexicon. ACL-MIT Series on Natural Language Processing, MIT Press.

[8] Roche, E., Schabes, Y. (eds.). (1997). *Finite State Language Processing*. The MIT Press.

[9] Sahoo, K. (2001). *Oriya Verb Morphology and Complex Verb Constructions*. Ph.D dissertation. Norwegian University of Science and Technology, Trondheim, Norway.

[10] Sahoo, K. (2012). Telicity vs. Perfectivity: A Case Study of Odia Complex Predicates, *The SKY journal of Linguistics* 25, 2012. The Linguistic Association of Finland.

Biography

Kalyanamalini Sahoo, currently working as a Researcher at the University of Antwerp, is Assistant Professor of General and English Linguistics at the English and Foreign Languages University, Hyderabad, India. She mostly works on Computational Morphology and Syntax.