

## References

- [1] B. Hettige, A. S. Karunananda (2008), Web-based English to Sinhala Selected Texts Translation system
- [2] Boretz, Adam,(2009), AppTek Launches Hybrid Machine Translation Software.
- [3] Brown, P. F., Della-Pietra, S. A., Della-Pietra, V.J. and Mercer, R. L. (1993), The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics, 19(2), pp 263-311.
- [4] Franz Josef Och, Hermann Ney (2004), The Alignment Template Approach to Statistical Machine Translation, 30(4)
- [5] <http://en.wikipedia.org/wiki/SYSTRAN>
- [6] Kevin Knight (1999), A Statistical MT Tutorial Workbook,
- [7] Weerasinghe A.R. (2005), A Statistical Machine Translation Approach to Sinhala-Tamil Language Translation,

## Parsers

These are some important code segments for the parsers written in Prolog.

### A.1 Code Segments for English Parser

#### A.1.1 Identifying nouns

##### Determinate

```
nEng(nEng(BaseEng, BaseSin, noun, Ani_Inani, Gender, Accu_Sub, Plu_Sin, determinate, none, none, none)) -->
[[t, h, e]], [N], {findEng(BaseEng, BaseSin, N, noun, Ani_Inani, Gender, Accu_Sub, Plu_Sin, none, none, none)}.
```

##### Indeterminate

```
nEng(nEng(BaseEng, BaseSin, noun, Ani_Inani, Gender, Accu_Sub, singular, indeterminate, none, none, none)) -->
[[a]], [N], {findEng(BaseEng, BaseSin, N, noun, Ani_Inani, Gender, Accu_Sub, singular, none, none, none)}.
```



University of Moratuwa, Sri Lanka.  
Electronic Theses & Dissertations  
lib.mrt.ac.lk

#### A.1.2 Identifying adjectives

```
adjEng(adjEng(BaseEng, BaseSin, adjective, Most, none)) -->
[ADJ], {findEng(BaseEng, BaseSin, ADJ, adjective, Most, none, none, none, none, none, none, none)}.
```

#### A.1.3 Identifying compound adjectives

```
adjComEng(adjComEng(ADJ)) --> adjEng(ADJ).
adjComEng(adjComEng(ADJ1, ADJ2)) --> adjEng(ADJ1), adjEng(ADJ2).
adjComEng(adjComEng(ADJ1, ADJ2, ADJ3)) -->
adjEng(ADJ1), adjEng(ADJ2), adjEng(ADJ3).
```

#### A.1.4 Identifying prepositions

```
prepEng(prepareEng(BaseEng, BaseSin, preposition, none)) -->
[PREP], {findEng(BaseEng, BaseSin, PREP, preposition, none, none, none, none, none, none, none)}.
```

#### A.1.5 Identifying noun phrases

```
npEng(npEng(ADJ, nEng(BaseEng, BaseSin, noun, Ani_Inani, Gender, Accu_Sub, plural, indeterminate, Prepos, Pronoun, Question)), charact(noun, Ani_Inani, Gender, Accu_Sub, plural, Prepos)) -->
```

adjComEng (ADJ) , nEng (nEng (BaseEng, BaseSin, noun, Ani\_Inani, Gender, Accu\_Sub, plural, none, Prepos, Pronoun, Question)) .

### A.1.6 Identifying verbs

#### %\*\*\*\*\*PRESENT TENSE\*\*\*\*\*

```
vEng (vEng (BaseEng, BaseSin, verb, ObjectType, Ani_Inani, Gender, Plu_Sin, simplepresent, Noun_Type, Question)) -->
[V], {findEng (BaseEng, BaseSin, V, verb, ObjectType, Ani_Inani, Gender, Plu_Sin, simplepresent, Noun_Type, Question, none)} .
```

```
vEng (vEng (BaseEng, BaseSin, verb, ObjectType, animate, none, singular, presentcont , pronounfirst, Question)) --> [[a,m]],
[V], {findEng (BaseEng, BaseSin, V, verb, ObjectType, animate, none, singular, ing_form, pronounfirst, Question, none)} . % i am
```

```
vEng (vEng (BaseEng, BaseSin, verb, ObjectType, Ani_Inani, Gender, singular, presentcont, pronounsecond, Question)) --> [[a,r,e]],
[V], {findEng (BaseEng, BaseSin, V, verb, ObjectType, Ani_Inani, Gender, singular, ing_form, pronounsecond, Question, none)} . %you are
```

```
vEng (vEng (BaseEng, BaseSin, verb, ObjectType, Ani_Inani, Gender, plural, presentcont , Noun_Type, Question)) --> [[a,r,e]],
[V], {findEng (BaseEng, BaseSin, V, verb, ObjectType, Ani_Inani, Gender, plural, ing_form, Noun_Type, Question, none)} .
```

```
vEng (vEng (BaseEng, BaseSin, verb, ObjectType, Ani_Inani, Gender, singular, presentcont , Noun_Type, Question)) --> [[i,s]],
[V], {findEng (BaseEng, BaseSin, V, verb, ObjectType, Ani_Inani, Gender, singular, ing_form, Noun_Type, Question, none)} .
```

```
vEng (vEng (BaseEng, BaseSin, verb, ObjectType, Ani_Inani, Gender, singular, presentperfect , pronounfirst, Question)) --> [[h,a,v,e]],
[V], {findEng (BaseEng, BaseSin, V, verb, ObjectType, Ani_Inani, Gender, singular, pastparticiple, pronounfirst, Question, none)} . %I have
```

```
vEng (vEng (BaseEng, BaseSin, verb, ObjectType, Ani_Inani, Gender, singular, presentperfect , pronounsecond, Question)) --> [[h,a,v,e]],
[V], {findEng (BaseEng, BaseSin, V, verb, ObjectType, Ani_Inani, Gender, singular, pastparticiple, pronounsecond, Question, none)} . %you have
```

```
vEng (vEng (BaseEng, BaseSin, verb, ObjectType, Ani_Inani, Gender, plural, presentperfect , Noun_Type, Question)) --> [[h,a,v,e]],
[V], {findEng (BaseEng, BaseSin, V, verb, ObjectType, Ani_Inani, Gender, plural, pastparticiple, Noun_Type, Question, none)} .
```

```
vEng (vEng (BaseEng, BaseSin, verb, ObjectType, Ani_Inani, Gender, singular, presentperfect , Noun_Type, Question)) --> [[h,a,s]],
[V], {findEng (BaseEng, BaseSin, V, verb, ObjectType, Ani_Inani, Gender, singular, pastparticiple, Noun_Type, Question, none)} .
```

%~~~~past tense as past partisiple. (the man has cut,I have licked)

```
vEng (vEng (BaseEng, BaseSin, verb, ObjectType, Ani_Inani, Gender, singular, presentperfect , pronounfirst, Question)) --> [[h,a,v,e]],
[V], {findEng (BaseEng, BaseSin, V, verb, ObjectType, Ani_Inani, Gender, singular, simplepast, pronounfirst, Question, none)} . %I have
```

```
vEng (vEng (BaseEng, BaseSin, verb, ObjectType, Ani_Inani, Gender, singular, presentperfect , pronounsecond, Question)) --> [[h,a,v,e]],
[V], {findEng (BaseEng, BaseSin, V, verb, ObjectType, Ani_Inani, Gender, singular, simplepast, pronounsecond, Question, none)} . %you have
```

```
vEng (vEng (BaseEng, BaseSin, verb, ObjectType, Ani_Inani, Gender, plural, presentperfect , Noun_Type, Question)) --> [[h,a,v,e]],
[V], {findEng (BaseEng, BaseSin, V, verb, ObjectType, Ani_Inani, Gender, plural, simplepast, Noun_Type, Question, none)} .
```

vEng (vEng (BaseEng, BaseSin, verb, ObjectType, Ani\_Inani, Gender, singular, presentperfect, Noun\_Type, Question)) --> [[h, a, s]],  
[V], {findEng (BaseEng, BaseSin, V, verb, ObjectType, Ani\_Inani, Gender, singular, simplepast, Noun\_Type, Question, none)}.

vEng (vEng (BaseEng, BaseSin, verb, ObjectType, Ani\_Inani, Gender, singular, presentperfectcont, pronounfirst, Question)) --> [[h, a, v, e]], [[b, e, e, n]],  
[V], {findEng (BaseEng, BaseSin, V, verb, ObjectType, Ani\_Inani, Gender, singular, ing\_form, pronounfirst, Question, none)}. %I have  
vEng (vEng (BaseEng, BaseSin, verb, ObjectType, Ani\_Inani, Gender, singular, presentperfectcont, pronounsecond, Question)) -->  
[[h, a, v, e]], [[b, e, e, n]],  
[V], {findEng (BaseEng, BaseSin, V, verb, ObjectType, Ani\_Inani, Gender, singular, ing\_form, pronounsecond, Question, none)}. %you have  
vEng (vEng (BaseEng, BaseSin, verb, ObjectType, Ani\_Inani, Gender, plural, presentperfectcont, Noun\_Type, Question)) --> [[h, a, v, e]], [[b, e, e, n]],  
[V], {findEng (BaseEng, BaseSin, V, verb, ObjectType, Ani\_Inani, Gender, plural, ing\_form, Noun\_Type, Question, none)}.  
vEng (vEng (BaseEng, BaseSin, verb, ObjectType, Ani\_Inani, Gender, singular, presentperfectcont, Noun\_Type, Question)) --> [[h, a, s]], [[b, e, e, n]],  
[V], {findEng (BaseEng, BaseSin, V, verb, ObjectType, Ani\_Inani, Gender, singular, ing\_form, Noun\_Type, Question, none)}.

**%\*\*\*\*\*PAST TENSE\*\*\*\*\***

vEng (vEng (BaseEng, BaseSin, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, simplepast, Noun\_Type, Question)) -->  
[V], {findEng (BaseEng, BaseSin, V, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, simplepast, Noun\_Type, Question, none)}.

vEng (vEng (BaseEng, BaseSin, verb, ObjectType, Ani\_Inani, Gender, plural, pastcont, Noun\_Type, Question)) --> [[w, e, r, e]],  
[V], {findEng (BaseEng, BaseSin, V, verb, ObjectType, Ani\_Inani, Gender, plural, ing\_form, Noun\_Type, Question, none)}.  
vEng (vEng (BaseEng, BaseSin, verb, ObjectType, Ani\_Inani, Gender, singular, pastcont, Noun\_Type, Question)) --> [[w, a, s]],  
[V], {findEng (BaseEng, BaseSin, V, verb, ObjectType, Ani\_Inani, Gender, singular, ing\_form, Noun\_Type, Question, none)}.

vEng (vEng (BaseEng, BaseSin, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, pastperfect, Noun\_Type, Question)) --> [[h, a, d]],  
[V], {findEng (BaseEng, BaseSin, V, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, pastparticiple, Noun\_Type, Question, none)}.  
vEng (vEng (BaseEng, BaseSin, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, pastperfect, Noun\_Type, Question)) --> [[h, a, d]],  
[V], {findEng (BaseEng, BaseSin, V, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, simplepast, Noun\_Type, Question, none)}.

vEng (vEng (BaseEng, BaseSin, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, pastperfectcont, Noun\_Type, Question)) --> [[h, a, d]], [[b, e, e, n]],  
[V], {findEng (BaseEng, BaseSin, V, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, ing\_form, Noun\_Type, Question, none)}.

**%\*\*\*\*\*FUTURE TENSE\*\*\*\*\***

vEng (vEng (BaseEng, BaseSin, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, simplefuture, Noun\_Type, none)) -->  
[[w, i, l, l]], [V], {findEng (BaseEng, BaseSin, V, verb, ObjectType, none, none, none, none, none, none, none)}.

```
vEng(vEng(BaseEng, BaseSin, verb, ObjectType, Ani_Inani, Gender, Plu_Sin, fu
turecont , Noun_Type, none)) -->
[[w, i, l, l]], [[b, e]], [V], {findEng(BaseEng, BaseSin, V, verb, ObjectType, An
i_Inani, Gender, Plu_Sin, ing_form, Noun_Type, Question, none)}.
```

```
vEng(vEng(BaseEng, BaseSin, verb, ObjectType, Ani_Inani, Gender, Plu_Sin, fu
tureperfect , Noun_Type, none)) -->
[[w, i, l, l]], [[h, a, v, e]], [V], {findEng(BaseEng, BaseSin, V, verb, ObjectTyp
e, none, none, none, none, none, none, none, none)}.
```

```
vEng(vEng(BaseEng, BaseSin, verb, ObjectType, Ani_Inani, Gender, Plu_Sin, fu
tureperfectcont , Noun_Type, none)) -->
[[w, i, l, l]], [[h, a, v, e]], [[b, e, e, n]], [V], {findEng(BaseEng, BaseSin, V, ve
rb, ObjectType, Ani_Inani, Gender, Plu_Sin, ing_form, Noun_Type, Question, no
ne)}.
```

%\*\*\*\*\*

## A.2 Code Segments for Sinhala Parser

### A.2.1 Identifying nouns

```
nSin(nSin(BaseSin, BaseEng, Noun, Ani_Inani, Gender, Accu_Sub, Plu_Sin, Dete
rminer, none, none, none)) -->
[N], {findSin(BaseSin, BaseEng, N, Noun, Ani_Inani, Gender, Accu_Sub, Plu_Sin
, Determiner, none, none, none), allnouns(Noun)}.
```

### A.2.2 Identifying adjectives

```
adjSin(adjSin(BaseSin, BaseEng, adjective, Most, none)) -->
[ADJ], {findSin(BaseSin, BaseEng, ADJ, adjective, Most, none, none, none, none
, none, none, none)}.
```

### A.2.3 Identifying compound adjectives

```
adjComSin(adjComSin(ADJ)) --> adjSin(ADJ) .
adjComSin(adjComSin(ADJ1, ADJ2)) --> adjSin(ADJ1), adjSin(ADJ2) .
adjComSin(adjComSin(ADJ1, ADJ2, ADJ3)) --
> adjSin(ADJ1), adjSin(ADJ2), adjSin(ADJ3) .
```

### A.2.4 Identifying verbs

%\*\*\*\*\*PRESENT TENSE\*\*\*\*\*

```
vSin(vSin(BaseSin, BaseEng, verb, ObjectType, Ani_Inani, Gender, Plu_Sin, si
mplepresent, NounType, none)) -->
[V], {findSin(BaseSin, BaseEng, V, verb, ObjectType, Ani_Inani, Gender, Plu_S
in, simplepresent, NounType, none, none)}.
```

```
vSin(vSin(BaseSin, BaseEng, verb, ObjectType, Ani_Inani, Gender, Plu_Sin, pr
esentcont , NounType, none)) -->
```

[V], [SITINAWA], {findSin(BaseSin, BaseEng, V, verb, ObjectType, none, none, none, while, none, none, none)},

findSin([s, '.i', t, '.i', n, w, '.A'], [s, t, a, y], SITINAWA, verb, \_, Ani\_Inani, Gender, Plu\_Sin, simplepresent, NounType, none, none) }.

vSin(vSin(BaseSin, BaseEng, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, presentperfect, NounType, none))-->

[V], [[æ, q]], {findSin(BaseSin, BaseEng, V, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, pastparticiple, NounType, none, none) }.

vSin(vSin(BaseSin, BaseEng, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, presentperfect, NounType, none))-->

[V], [[q, '.i', b, '.E']], {findSin(BaseSin, BaseEng, V, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, pastparticiple, NounType, none, none) }.

vSin(vSin(BaseSin, BaseEng, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, presentperfectcont, NounType, none))-->

[V], [[s, '.i', t]], [[æ, q]], {findSin(BaseSin, BaseEng, V, verb, ObjectType, none, none, none, while, none, none, none) }.

vSin(vSin(BaseSin, BaseEng, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, presentperfectcont, NounType, none))-->

[V], [[s, '.i', t]], [[q, '.i', b, '.E']], {findSin(BaseSin, BaseEng, V, verb, ObjectType, none, none, none, while, none, none, none) }.

#### **%\*\*\*\*\*PAST TENSE\*\*\*\*\***

vSin(vSin(BaseSin, BaseEng, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, simplepast, NounType, none))-->

[V], {findSin(BaseSin, BaseEng, V, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, simplepast, NounType, none, none) }.

vSin(vSin(BaseSin, BaseEng, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, pastcont, NounType, none))-->

[V], [SITINAWA], {findSin(BaseSin, BaseEng, V, verb, ObjectType, none, none, none, while, none, none, none)},

findSin([s, '.i', t, '.i', n, w, '.A'], [s, t, a, y], SITINAWA, verb, \_, Ani\_Inani, Gender, Plu\_Sin, simplepast, NounType, none, none) }.

vSin(vSin(BaseSin, BaseEng, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, pastperfect, NounType, none))-->

[V], [[q, '.i', b, '.u', 'N', '.i']], {findSin(BaseSin, BaseEng, V, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, pastparticiple, NounType, none, none) }.

vSin(vSin(BaseSin, BaseEng, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, pastperfectcont, NounType, none))-->

[V], [[s, '.i', t]], [[æ, q]], {findSin(BaseSin, BaseEng, V, verb, ObjectType, none, none, none, while, none, none, none) }.

vSin(vSin(BaseSin, BaseEng, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, pastperfectcont, NounType, none))-->

[V], [[s, '.i', t]], [[q, '.i', b, '.E']], {findSin(BaseSin, BaseEng, V, verb, ObjectType, none, none, none, while, none, none, none) }.

#### **%\*\*\*\*\*FUTURE TENSE\*\*\*\*\***

vSin(vSin(BaseSin, BaseEng, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, simplefuture, NounType, none))-->

[V], {findSin(BaseSin, BaseEng, V, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, future, NounType, none, none)}.

vSin(vSin(BaseSin, BaseEng, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, futurecont, NounType, none))-->

[V], [[s, '.i', t, '.i', y, '.i']], {findSin(BaseSin, BaseEng, V, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, pastparticiple, NounType, none, none)}.

vSin(vSin(BaseSin, BaseEng, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, futureperfect, NounType, none))-->

[V], [[s, '.i', t, '.i', y, '.i']], {findSin(BaseSin, BaseEng, V, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, pastparticiple, NounType, none, none)}.

vSin(vSin(BaseSin, BaseEng, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, futureperfectcont, NounType, none))-->

[V], [[s, '.i', t, '.i', y, '.i']], {findSin(BaseSin, BaseEng, V, verb, ObjectType, Ani\_Inani, Gender, Plu\_Sin, pastparticiple, NounType, none, none)}.

\*\*\*\*\*

### A.3 Tenses

The following table lists out 12 different tenses with examples in both Sinhala and English.

Tense	Type	Example: English	Example: Sinhala
Present Tense	Simple present	The man cuts the tree.	මිනිසා ගස කපන්නේය.
	Present continuous	The man is cutting the tree.	මිනිසා ගස කපමින් සිටින්නේය.
	Present perfect	The man has cut the tree.	මිනිසා ගස කපා ඇත.
	Present perfect continuous	The man has been cutting the tree.	මිනිසා ගස කපමින් සිට ඇත.
Past Tense	Simple past	The man cut the tree.	මිනිසා ගස කැපුවේය.
	Past continuous	The man was cutting the tree.	මිනිසා ගස කපමින් සිටියේය.
	Past perfect	The man had cut the tree.	මිනිසා ගස කපා තිබුණේය.
	Past perfect continuous	The man had been cutting the tree.	මිනිසා ගස කපමින් සිට ඇත.
Future Tense	Simple future	The man will cut the tree.	මිනිසා ගස කපයි.
	Future continuous	The man will be cutting the tree.	මිනිසා ගස කපමින් සිටියි.
	Future perfect	The man will have cut the tree.	මිනිසා ගස කපා සිටියි.
	Future perfect continuous	The man will have been cutting the tree.	මිනිසා ගස කපා සිටියි.

Table A-1: The tenses in language.

## Morphology

These are some important code segments for the morphology analyzers written in Prolog.

### B.1 Code Segments for English Morphology Analyzer

These are the code segments for the morphology analyzers.

#### B.1.1 Changing noun to accusative form

```
morphEng(word(W ,Noun ,Animate ,Gender ,subjective
,Singular ,determinate, none ,none ,none ),
word(W1 ,Noun ,Animate ,Gender ,accusative
,Singular ,determinate, none ,none ,none
)):-allnouns(Noun),engAccusative(W,W1).
```

```
engAccusative([i] ,[m,e]):-!.
engAccusative([w,e] ,[u,s]):-!.
engAccusative([h,e] ,[h,i,m]):-!.
engAccusative([s,h,e] ,[h,e,r]):-!.
engAccusative([t,h,e,y] ,[t,h,e,m]):-!.
```

```
morphEng(word(W ,Noun ,Ani_Inani ,Gender
,subjective ,singular ,Determiner , none ,none
,none ),
word(W ,Noun ,Ani_Inani ,Gender
,accusative ,singular ,Determiner , none ,none
,none )):-allnouns(Noun). % all nouns are subjective and
accusative in english (except the above).
```

#### B.1.2 Changing a verb to simple present tense

%\*\*\*\*\*verbs\*\*\*\*\*

```
morphEng( word( W ,verb ,ObjectType,none ,none ,none ,none
,none ,none ,none) ,
word( W1 ,verb ,ObjectType,Ani_Inani ,Gender ,singular
,simplepresent ,Noun ,none ,none)) :-
concat(W,[s],W1) ,
gender(Gender) ,
ani_inani(Ani_Inani),
allnouns(Noun).%a man cuts the tree
```



### B.1.3 Changing a verb to simple past tense

```
morphEng( word( W ,verb ,ObjectType,none ,none ,none ,none
,none ,none ,none) ,
word( W1 ,verb ,ObjectType,Ani_Inani ,Gender ,Plu_Sin
,simplepast ,Noun ,none ,none)) :- concat(W, [e,d],W1),

gender(Gender) ,

ani_inani(Ani_Inani),

plu_sin(Plu_Sin), allnouns(Noun).

morphEng( word( W ,verb ,ObjectType,none ,none ,none ,none
,none ,none ,none) ,
word( W1 ,verb ,ObjectType,Ani_Inani ,Gender ,Plu_Sin
,simplepast ,Noun ,none ,none)) :-
doubleLastAndConcat(W, [e,d],W1),

gender(Gender) ,

ani_inani(Ani_Inani),

plu_sin(Plu_Sin), allnouns(Noun).
```

### B.1.4 Changing a verb to past participle tense

```
morphEng( word( W ,verb ,ObjectType,none ,none ,none ,none
,none ,none ,none) ,
word( W1 ,verb ,ObjectType,Ani_Inani ,Gender ,Plu_Sin
,pastparticiple ,Noun ,none ,none)) :-
concat(W, [e,n],W1),

gender(Gender) ,

ani_inani(Ani_Inani),

plu_sin(Plu_Sin), allnouns(Noun).
```

## B.2 Code Segments for Sinhala Morphology Analyzer

### B.2.1 Changing noun to accusative form

Note that in Sinhala the accusative object form has a “Ta” or “Wa” suffix.

```
%##### Accusative #####

morphSin(word(W ,Noun ,Animate ,Gender ,subjective
,Singular ,determinate, none ,none ,none ),

word(W1 ,Noun ,Animate ,Gender ,Accu_Sub
,Singular ,determinate, none ,none ,none
)):-allnouns(Noun),sinAccusative(W,W1,Accu_Sub),!.
```

```

sinAccusative([m,m] , [m,'.A'] , accusative).
sinAccusative([a,p,'.i'] , [a,p] , accusative).
sinAccusative([o,w,'.u',n,.] , [o,w,'.u',n,.] , accusative).

sinAccusative([m,m] , [m,t] , accusativeTa).

morphSin(word(W , Noun , Ani_Inani , Gender
, subjective , singular , determinate, none
, none , none ),
word(W , Noun , Ani_Inani , Gender
, accusative , singular , determinate, none
, none , none )):-allnouns(Noun).% all singular determinate
nouns are subjective and accusative in sinhala (except the above).

morphSin(word(W , Noun , inanimate , none
, accusative , plural , Determinate, none
, none , none ),
word(W1 , Noun , inanimate , none
, accusativeTa , plural , Determinate, none
, none , none )):-
allnouns(Noun),concat(W,[w,l,t],W1).%gaswalata

morphSin(word(W , Noun , inanimate , none
, accusative , singular , determinate, none
, none , none ),
word(W1 , Noun , inanimate , none
, accusativeTa , singular , indeterminate, none
, none , none )):-
allnouns(Noun),concat(W,[k,t],W1).%gasakata

morphSin(word(W , Noun , animate , Gender
, accusative , Sin_Plu , Determinate, none
, none , none ),
word(W1 , Noun , animate , Gender
, accusativeTa , Sin_Plu , Determinate, none
, none , none )):-
allnouns(Noun),concat(W,[t],W1).%minisata,minisunta

morphSin(word(W , Noun , inanimate , none
, accusative , singular , determinate, none
, none , none ),
word(W1 , Noun , inanimate , none
, accusativeTa , singular , determinate, none
, none , none )):-allnouns(Noun),concat(W,[t],W1).%gasata

morphSin(word(W , Noun , inanimate , none
, accusative , Sin_Plu , Determinate, none
, none , none ),
word(W , Noun , inanimate , none
, accusativeWa , Sin_Plu , Determinate, none
, none , none )):-allnouns(Noun).%gasa

morphSin(word(W , Noun , animate , Gender
, accusative , Sin_Plu , Determinate, none
, none , none ),
word(W1 , Noun , animate , Gender
, accusativeWa , Sin_Plu , Determinate, none
, none , none )):-
allnouns(Noun),concat(W,[w],W1).%minisunwa

```

```
% ##### END OF ACCUSATIVE #####
```

## B.2.2 Changing noun to indeterminate form

```
morphSin( word( W ,noun ,inanimate ,none ,Accu_Sub
            ,singular ,determinate ,none ,none ,none )
            ,
            word( W1 ,noun ,inanimate ,none ,Accu_Sub
            ,singular ,indeterminate ,none ,none ,none )) :-
accu_sub(Accu_Sub),concat(W,[k,],W1).
```

```
morphSin( word( W ,noun ,animate ,Gender ,Accu_Sub
            ,singular ,determinate ,none ,none ,none )
            ,
            word( W1 ,noun ,animate ,Gender ,Accu_Sub
            ,singular ,indeterminate ,none ,none ,none )) :-
accu_sub(Accu_Sub),replacelastA(W,['.e',k,],W1).
```

## B.2.3 Changing noun to “from” form



University of Moratuwa, Sri Lanka.  
Electronic Theses & Dissertations  
www.lib.mrt.ac.lk

```
%from
```

```
morphSin( word( W ,noun ,animate ,Gender ,accusative
            ,Plu_Sin ,Determiner ,none ,none ,none )
            ,
            word( W1 ,noun ,animate ,Gender ,accusative
            ,Plu_Sin ,Determiner ,from ,none ,none )) :-
concat(W,[g, '.e',n,],W1) .
```

```
morphSin( word( W ,pronoun ,animate ,Gender
            ,accusative ,Plu_Sin ,Determiner ,none ,none ,none )
            ,
            word( W1 ,pronoun ,animate ,Gender
            ,accusative ,Plu_Sin ,Determiner ,from ,none ,none )) :-
concat(W,[g, '.e',n,],W1) .
```

```
morphSin( word( W ,noun ,inanimate ,none ,accusative ,singular
            ,determinate ,none ,none ,none )
            ,
            word( W1 ,noun ,inanimate ,none ,accusative ,singular
            ,indeterminate ,from ,none ,none )) :-
concat(W,[k, '.i',n,],W1) .
```

```
morphSin( word( W ,noun ,inanimate ,none ,accusative ,plural
            ,determinate ,none ,none ,none )
            ,
            word( W1 ,noun ,inanimate ,none ,accusative ,plural
            ,determinate ,from ,none ,none )) :-
concat(W,[w,l, '.i',n,],W1) .
```

```

morphSin( word( W ,noun ,inanimate ,none ,accusative ,singular
,determinate ,none ,none ,none ) ,
word( W1 ,noun ,inanimate ,none ,accusative ,singular
,determinate ,from ,none ,none )) :-
concat(W, ['.e',n, .],W1) .

```

### B.2.4 Changing adjective to “most” form


```

morphSin( word( W ,adjective ,none ,none ,none ,none ,none
,none ,none,none ) ,
word( W1 ,adjective ,most ,none ,none ,none ,none
,none ,none,none )) :- concat(W, [m],W1) .

```

## B.3 Examples of Morphology in Sinhala

### B.3.1 Nouns (noun, pronoun – first, second, third person)

VAR1	VAR2	VAR3	VAR4	VAR5	VAR6	VAR7	VAR8
Animate/ Inanimate	 Gender	Accusative/ Subjective	Singular/ Plural	Determiner	Preposition	Pronoun	Question
Animate මිනිසා	male මැස්සා	Subjective මිනිසා	singular ළමයා මම	definite මල	To මිනිසාට	itself භාවම පොතම ඔහුම	question මෙද?
inanimate ගස	female මැස්සී	Accusative මිනිසාට මිනිසාව	plural ළමයි අපි	indefinite මලක්	From මිනිසාගෙන්	Himself ඔහුම	also භාවන් පොතයි ඔහුද
	none ගෙඩිය				On ගසේ	Themselves ඔවුන්ම	is පොතය

	මම						ඔහුය
	ආපි						
					in	Herself	
					අල්මාරියේ	ඇයම	
					of	Yourself	
					අල්මාරියේ	ඔබම	

Table B-1: Morphology of nouns in Sinhala

### B.3.2 Verbs

VAR1	VAR2	VAR3	VAR4	VAR5	VAR6
Animate/ Inanimate	Gender	Singular/ Plural	Tense	Noun Type	Question
Inanimate කැවේය	Male බැලුවේය	Singular බලන්නේය	simplepresent කරන්නේය බලන්නේය	noun බලන්නේය	question කන්නෙහිද
Animate කැවේය	Female බැලුවාය	Plural බලන්නෝය	presentcont බලමින් සිටින්නේය	pronounfirst බලන්නෙමී බැලුවෙමු	
			Simplepast බැලුවේය	pronounsecond බැලුවේය	
			pastcontinuous බලමින් සිටියේය	Pronounthird බැලුවේය	

Table B-2: Morphology of verbs in Sinhala

### B.3.3 Adjective

VAR1
Most/More
Most
උසම
More
වඩා උස

Table B-3: Morphology of adjectives in Sinhala

### B.3.4 Adverb

VAR1
Most/More
Most
වේගවත්ම
More
වඩා වේගවත්

Table B-4: Morphology of adverbs in Sinhala

## B.4 Examples of Morphology in English

### B.4.1 Nouns (noun, pronoun – first, second, third person)

VAR1	VAR2	VAR3	VAR4	VAR5	VAR6	VAR7	VAR8
Animate/	Gender	Accusative/	Singula	Determin	Preposition	Pronoun	Question

Inanimate		Subjective	r/ Plural	er			
Animate Fly	male man	Subjective The Man We She	singular child	definite the flower	To To the man	itself the book itself	question is it me
inanimate tree	female girl	Accusative The Man Us Her	plural children we	indefinite a flower	From From the man	Himself He himself	
	none fruit I We				On On the tree	Themsel ves They themselv es	
					in in the cupboard	Herself She herself	
					of of the tree	Yourself You yourself	

Table B-5: Morphology of nouns in English

### B.4.2 Verbs

VAR1	VAR2	VAR3	VAR4	VAR5	VAR6
Animate/ Inanimate	Gender	Singular/ Plural	Tense	Noun Type	Question
Inanimate Fell	Male looked	Singular went	simplepresent Runs	noun looks	
Animate Fell	Female looked	Plural went	presentcont Is running	pronounfirst look	
			simplepast Ran	pronounsecond look	
			pastcontinuous Was running	pronounthird looks	

Table B-6: Morphology of verbs in English

### B.4.3 Adjective

VAR1
Most/More
Most
Tallest
More
Taller

Table B-7: Morphology of adjectives in English



#### B.4.4 Adverb

VAR1
Most/More
Most Fastest
More Faster

Table B-8: Morphology of adverbs in English

### B.5 Common Morphology Rules

#### B.5.1 Nouns

##### B.5.1.1 Accusative Forms in English

All words except the pronouns are the same in the accusative form.

Me	animate	none	accusative	singular	determinate	none	none	none
Us	animate	none	accusative	plural	determinate	none	none	none
Him	animate	male	accusative	singular	determinate	none	none	none
Her	animate	female	accusative	singular	determinate	none	none	none
You	animate	none	accusative	singular	determinate	none	none	none
You	animate	none	accusative	plural	determinate	none	none	none
Them	animate	none	accusative	plural	determinate	none	none	none
It	inanimate	none	accusative	singular	determinate	none	none	none
All Words	Ani-Inani	Gender	accusative	Sin_Plu	Determiner	none	none	none

Table B-9: Accusative forms in English

### B.5.1.2 Accusative Forms in Sinhala

ගස ගසට	inanimate	none	accusative	singular	determinate	none	none	none
ගස් ගස්වලට	inanimate	none	accusative	plural	Determiner	none	none	none
ගසක් ගසකට	inanimate	none	accusative	singular	indeterminate	none	none	none
මිනිසාට මිනිසාව	animate	Gender	accusative	singular	determinate	none	none	none
මිනිසෙකුට මිනිසෙකුව	animate	Gender	accusative	singular	indeterminate	none	none	none
මිනිසුන්ට මිනිසුන්ව	animate	Gender	accusative	plural	Determiner	none	none	none

Table B-10: Accusative forms in Sinhala.

### B.5.1.3 Plural Form in English

Trees	Inanimate	none	Sub_Accu	plural	Determiner	none	none	none
Dogs	animate	Gender	Sub_Accu	plural	Determiner	none	none	none

Table B-11: Plural form in English

#### B.5.1.4 Plural Form in Sinhala

ගස් සුදු බෙර	Inanimate	none	Sub_Accu	plural	Determiner	none	none	none
බල්ලෝ මිනිස්සු හරක් නරි	animate	Gender	Sub_Accu	plural	Determiner	none	none	none

Table B-12: Plural form in Sinhala

#### B.5.1.5 Indeterminate form in Sinhala

ගසක්	Inanimate	none	Sub_Accu	singular	indeterminate	none	none	none
බල්ලෙක්	animate	Gender	Sub_Accu	singular	indeterminate	none	none	none

Table B-13: Indeterminate form in Sinhala

#### B.5.1.6 Preposition “to” in Sinhala

ගසට	inanimate	none	accusative	singular	determinate	to	none	none
ගසකට	inanimate	none	accusative	singular	indeterminate	to	none	none
ගස්වලට	inanimate	none	accusative	plural	Determiner	to	none	none
බල්ලාට	animate	Gender	accusative	singular	determinate	to	none	none
බල්ලෙකුට	animate	Gender	accusative	singular	indeterminate	to	none	none
බල්ලන්ට	animate	Gender	accusative	plural	Determiner	to	none	none

Table B-14: To preposition in Sinhala

### B.5.1.7 Preposition “from” in Sinhala

ගසෙන්	inanimate	none	accusative	singular	determinate	from	none	none
ගසකින්	inanimate	none	accusative	singular	indeterminate	from	none	none
ගස්වලින්	inanimate	none	accusative	plural	Determiner	from	none	none
බල්ලාගෙන්	animate	Gender	accusative	singular	determinate	from	none	none
බල්ලන්ගෙන්	animate	Gender	accusative	plural	Determiner	from	none	none

Table B-15: From preposition in Sinhala

### B.5.1.8 Preposition “of” in Sinhala

ගසේ	inanimate	none	accusative	singular	determinate	of	none	none
ගසක	inanimate	none	accusative	singular	indeterminate	of	none	none
ගස්වල	inanimate	none	accusative	plural	Determiner	of	none	none
බල්ලාගේ	animate	Gender	accusative	singular	determinate	of	none	none
බල්ලන්ගේ	animate	Gender	accusative	plural	Determiner	of	none	none

Table B-16: Of preposition in Sinhala

## B.5.2 Verbs

### B.5.2.1 Simple Present Tense

#### English

Look	Ani_Inani	none	singular	simplepresent	pronounfirst	none
Look	Ani_Inani	none	Sin_Plu	simplepresent	pronounsecond	none
Looks	Ani_Inani	Gender	singular	simplepresent	All_Nouns	none
Look	Ani_Inani	Gender	Plural	simplepresent	All_Nouns	none

Table B-17: Simple present tense in English

#### Sinhala

බලන්නේය	inanimate	none	Sin_Plu	simplepresent	noun	none
බලන්නේය	animate	male	singular	simplepresent	All_Nouns	none
බලන්නීය	animate	female	singular	simplepresent	All_Nouns	none
බලන්නේය	animate	Gender	plural	simplepresent	All_Nouns	none
බලන්නෙමි/ බලමි	animate	none	singular	simplepresent	pronounfirst	none
බලන්නෙමු	animate	none	plural	simplepresent	pronounfirst	none

Table B-18: Simple present tense in Sinhala

### B.5.2.2 Present Continuous Tense

#### English

am looking	animate	none	singular	presentcont	pronounfirst	none
are looking	animate	none	Sin_Plu	presentcont	pronounsecon	none
is looking	Ani_Inani	Gender	singular	presentcont	All_Nouns	none
are looking	Ani_Inani	Gender	plural	presentcont	All_Nouns	none

Table B-19: Present continuous tense in English

## Sinhala

බලන්නේය බලමින් සිටින්නේය	inanimate	none	Sin_Plu	presentcont	noun	none
බලන්නේය බලමින් සිටින්නේය	animate	male	singular	presentcont	All_Nouns	none
බලන්නීය බලමින් සිටින්නීය	animate	female	singular	presentcont	All_Nouns	none
බලන්නෝය/බලනී බලමින් සිටින්නෝය/සිටිනී	animate	Gender	plural	presentcont	All_Nouns	none

Table B-20: Present continuous tense in Sinhala

### B.5.2.3 Simple past tense

#### English

Looked	Ani_Inani	Gender	Sin_Plu	simplepast	All_Nouns	none
--------	-----------	--------	---------	------------	-----------	------

Table B-21: Simple past tense in English

#### Sinhala

බලුවේය	inanimate	none	Sin_Plu	simplepast	noun	none
බලුවේය	animate	male	singular	simplepast	All_Nouns	none
බලුවාය	animate	female	singular	simplepast	All_Nouns	none
බලුවෝය	animate	Gender	plural	simplepast	All_Nouns	none

Table B-22: Simple past tense in Sinhala

### B.5.2.4 Past continuous tense

#### English

Was looking	animate	none	singular	presentcont	pronounfirst	none
Were looking	animate	none	Sin_Plu	presentcont	pronounsecon	none

Was looking	Ani_Inani	Gender	singular	presentcont	All_Nouns	None
Were looking	Ani_Inani	Gender	plural	presentcont	All_Nouns	None

Table B-23: Past continuous tense in English



University of Moratuwa, Sri Lanka.  
 Electronic Theses & Dissertations  
[www.lib.mrt.ac.lk](http://www.lib.mrt.ac.lk)

# Testing and Quality Assurance

## C.1 Testing and Quality Assurance

### C.1.1 Unit testing

Each component of the software will go through a unit test during all development stages. The following process will be carried out during the unit test.

- Review modular code
- Test component modules to specifications
- Identify anomalies to specifications
- Modify code
- Re-test modified code

### C.1.2 Integration testing

After each development stage the software will be tested as a whole before it is handed over. The following process will be carried out during each integration test stage.

- Test module integration
- Identify anomalies to specifications
- Modify code
- Re-test modified code

### C.1.3 Stress test

The robustness, availability and exception handling will be tested under a heavy load. In particular, the goals of such tests are to ensure the software doesn't crash in conditions of insufficient computational resources (memory or disc space), and unusually high concurrency.

The software will be stress tested using standard testing tools to observe the performance of the service during peak loads.



## Appendix D

### Schedule

#### D.1 Deliverables

These are the deliverables of the project and the dates on which they were delivered.

S/N	Due Date	Deliverable
1	19-Sep-09	Project proposal
2	25-Dec-09	Rule Based Translator
3	22-Jan-10	Working rules based transliteration software with transliteration module
4	13-Apr-10	Complete translation with statistical approach
5	01-Mar-10	Interim report
6	30-Jun-10	Final thesis
7	01-Aug-10	Research paper

Table D-1: The deliverables of the project

#### D.2 Plan of Action

The following table lists out the schedule which the work was carried out.

ID	Name	Duration	Start	Finish
1	<b>Proposal</b>	<b>10.d</b>	<b>7-Sep-09</b>	<b>18-Sep-09</b>
2	Preliminary study	7.d	7-Sep-09	15-Sep-09
3	Writing proposal	5.d	14-Sep-09	18-Sep-09
4	<b>Rule Based Module</b>	<b>70.d</b>	<b>21-Sep-09</b>	<b>25-Dec-09</b>
5	Waiting for approval and modify if necessary	8.d	21-Sep-09	30-Sep-09
6	Critical study on Problem and	10.d	1-Oct-09	14-Oct-09

	technologies			
7	Morphology Analyzer for Sinhala	15.d	15-Oct-09	4-Nov-09
8	Morphology Analyzer for English	10.d	5-Nov-09	18-Nov-09
9	Sinhala Parser	6.d	19-Nov-09	26-Nov-09
10	English Parser	6.d	27-Nov-09	4-Dec-09
13	Sinhala Rule Engine	6.d	10-Dec-09	17-Dec-09
14	English Rule Engine	6.d	18-Dec-09	25-Dec-09
15	<b>Transliteration</b>	<b>20.d</b>	<b>28-Dec-09</b>	<b>22-Jan-10</b>
16	Finding rules for common name syllables.	7.d	28-Dec-09	5-Jan-10
17	Creating rule base with prolog	5.d	6-Jan-10	12-Jan-10
18	Collecting exceptional names and creating name corpus	5.d	13-Jan-10	19-Jan-10
19	creating complete transliteration with rules and name entity dictionary	3.d	20-Jan-10	22-Jan-10
20	<b>Statistics Based Module</b>	<b>36.d</b>	<b>25-Jan-10</b>	<b>15-Mar-10</b>
21	Crawler for Sinhala and English web sites	6.d	25-Jan-10	1-Feb-10
22	<b>Creating Sinhala English Corpus</b>	<b>30.d</b>	<b>2-Feb-10</b>	<b>15-Mar-10</b>
23	Designing database	5.d	2-Feb-10	8-Feb-10
24	Extracting sentences from	4.d	9-Feb-10	12-Feb-10

	bulk text			
<b>25</b>	Extracting words from sentences	6.d	15-Feb-10	22-Feb-10
<b>26</b>	Finding statistics	15.d	23-Feb-10	15-Mar-10
<b>27</b>	<b>Complete Translator</b>	<b>21.d</b>	<b>16-Mar-10</b>	<b>13-Apr-10</b>
<b>28</b>	Aligning rule based solution with statistics	6.d	16-Mar-10	23-Mar-10
<b>29</b>	Fine Tuning	15.d	24-Mar-10	13-Apr-10
<b>30</b>	<b>Documents</b>	<b>15.d</b>	<b>9-Feb-10</b>	<b>1-Mar-10</b>
<b>31</b>	Interim report	15.d	9-Feb-10	1-Mar-10
<b>32</b>	<b>Thesis</b>	<b>53.d</b>	<b>19-Apr-10</b>	<b>30-Jun-10</b>
<b>33</b>	Draft	30.d	19-Apr-10	28-May-10
<b>34</b>	Final	20.d	3-Jun-10	30-Jun-10

Table D-2: Plan of action.