Numeral System in Tamil: Generation

S. Agesthialingom

Numerals in any natural language form a separate group of nouns which behave differently in certain respects both in its formation and in its grammatical functions. All the other nouns denote a person, place, thing, time, etc., whereas numerals denote the number of the things, persons etc., in addition to its behaviour as noun.

The present paper is concerned with the generation of this category of nouns and tries to show that with a small number of generating rules it is possible to generate all the numbers in Tamil, of course with a certain number of morphophonemic rules. Here the integer numbers are alone considered and the fractions and complex numbers like *oṉṟē kāl* (1 ¼) *oṉṟarai* (1 ½) are not dealt with. It is to be mentioned here that they can be easily taken care of by certain rules like

$$NUM_{comp} \longrightarrow NUM_F + FRA$$

Where NUM_{comp} stands for complex numbers and NUM_F for integer numbers like *oṉṟu*, *iraṇṭu* etc., FRA for fractions like *kāl* (1/4), *arai* (1/2), *mukkāl*; (3/4) etc.,.

As is well known, integer numbers are classified into: one-digit numbers, two-digit numbers, three-digit numbers, etc.. In Tamil there are special names for various digits though it is not uniform in all the digits. For example there is no special name for the one-digit numbers, whereas two-digit numbers are denoted by *pattu* "ten", three-digit numbers by *nūṟu* "hundred", four- and five-digits are called *āyiram* "thousand", six- and seven-digits are called *ilaṭcam* "lakh" and eight-digits and above are called *kōṭi* "crore". This can be shown as follows.

Integer Numbers							
One digit	Two digits	Three digits	Four digits	Five digits	Six digits	Seven digits	Eight digits
1 – 9	10 - 99	100 - 999	1000 -	99999	100000 -	9999999	10000000 and above
	Tens	Hundreds	Thousands		Lakhs		Crores
	pattu	*nūṟu*	āyiram		*ilaṭcam*		*kōṭi*
Nu_1	Nu_2	Nu_3	Nu_4		Nu_5		Nu_6

Since *āyiram* and its multiples and additions can take care of 1000 to 99999, i.e. both four-digit numbers and five-digit numbers they are clubbed together and are represented by NU_4. This is also the case with *ilaṭcam*. NU_5 represents both six-digits and seven-digits.

South Indian Horizons (F. Gros Felicitation Volume) pp. 323–332

All these digits or all the numbers can be generated by a rule like the following:

1.1

$$NU_F \longrightarrow \left\{ \begin{array}{l} NU_1 \\ NU_2 \\ NU_3 \\ NU_4 \\ NU_5 \\ NU_6 \end{array} \right\}$$

It means that the numbers may be one-digit numbers, or two-digits, or three-digits, ..., or eight-digits, or more digit numbers.

- NU_1 stands for *oṉṟu* "one" to *oṉpatu* "9"
- NU_2 stands for *pattu* "ten" to *toṇṇūṟṟu oṉpatu* (99)
- NU_3 stands for nūṟu "hundred" to *toḷḷāyirattu toṇṇūṟṟu oṉpatu* (999)
- NU_4 stands for āyiram "thousand" to *toṇṇūṟṟu oṉpatu āyirattu toḷḷāyirattu toṇṇūṟṟu oṉpatu* (99999)
- NU_5 stands for *ilaṭcam* "lakh" *oru ilaṭcam*" (100000 to 9999999)
- NU_6 stands for *kōṭi* "crore" "*oru kōṭi* to infinite" (10000000 to infinite).

Three kinds of rules

This grammar consists of three kinds of rules namely

- (1) generation rules (I)
- (2) allomorphic rules (II)
- (3) sandhi rules (III).

The first set of rules is responsible for generating all numbers. The second set takes care of allomorphic variations and the third takes care of sandhi changes.

All these rules put together can generate all the natural numbers in Tamil and they can be used as input data and by various processes it may be possible for us to get the output.

Generation rules

1.2

$$NU_1 \longrightarrow \left\{ \begin{array}{l} \textit{oṉṟu} \\ \textit{iraṇṭu} \\ \textit{mūṉṟu} \\ \textit{nāṉku} \\ \textit{aintu} \\ \textit{āṟu} \\ \textit{ēḻu} \\ \textit{eṭṭu} \\ \textit{oṉpatu} \end{array} \right\}$$

NU_1 stands for numbers *oṉṟu* (1) to *oṉpatu* (9) i.e. one to nine. NU_1 is *oṉṟu*, *iraṇṭu* or *mūṉṟu*, etc. All these are represented by NU_1

It is known that all these numbers have allomorphs and they will be taken care of by appropriate morphophonemic rules (II).

Similarly *pattu, nūṟu, āyiram, ilaṭcam,* etc. may also have to undergo morphophonemic rules and they will be also treated by morphophonemic rules.

1.3 $\qquad NU_2 \longrightarrow NU_1 \times \textit{pattu} + (NU_1)$

Here simple bracket ( ) means optional. i.e. NU_1 may be there or may not be there. The second digit number is represented by this rule.

This rule shows that two digit numbers are the multiples of *pattu* and also with or without NU_1. This can be shown in something like the following.

$$\left\{\begin{array}{l}\underline{\text{onru}}\\ \underline{\text{irantu}}\\ \underline{\text{mūnru}}\\ \underline{\text{nānku}}\\ \underline{\text{aintu}}\\ \underline{\text{āru}}\\ \underline{\text{ēlu}}\\ \underline{\text{ettu}}\\ \underline{\text{onpatu}}\end{array}\right\} \text{X pattu} + \left(\left\{\begin{array}{l}\underline{\text{onru}}\\ \underline{\text{irantu}}\\ \underline{\text{mūnru}}\\ \underline{\text{nānku}}\\ \underline{\text{aintu}}\\ \underline{\text{āru}}\\ \underline{\text{ēlu}}\\ \underline{\text{ettu}}\\ \underline{\text{onpatu}}\end{array}\right\} \right)$$

So this rule can generate

oṉṟu x *pattu* > *pattu* (*oṉṟu* will be deleted by a rule later 2.13)
iraṇṭu x *pattu* > *irupatu* (2.3; 2.14) etc.,

Similarly if we add NU_1 after *pattu* we will get

oṉṟu x *pattu* + *oṉṟu* > *patiṉoṉṟu* (2.11;13)
oṉṟu x *pattu* + *iraṇṭu* > *paṉṉiraṇṭu* (2.12;13) etc.,

Multiplication & Addition

Here we find two kinds of processes in the generation of numbers in Tamil, namely multiplication and addition. When one to nine are added before *pattu* they are in the process of multiplication.

oṉṟu x *pattu* > *pattu* "ten" (2.13)
iraṇṭu x *pattu* > *irupatu* "Twenty" (2.3,14)

But in the case of *pattu+oṉṟu*, it is in the process of addition.

pattu + *oṉṟu* > *patiṉoṉṟu* (11) (2.11,13)

It is to be noted that it is the same in the case of *nūṟu, āyiram, ilaṭcam,* etc.,

When the smaller number comes before a bigger number it is in a multiplication process whereas when the smaller numbers come after the bigger numbers it is in a process of addition. That is why the signs x and + are used in the rule.

In the morphophonemic rules also this is reflected

iraṇṭu x *pattu* > *irupatu* (20) (2.3 ; 14)
irupatu āyiram (multiplication)
irupattu iraṇṭu (addition) (2.3)

Deletion of *oṉṟu*

When *oṉṟu* comes before *pattu* (ten) *oṉṟu* is deleted (2.13) and the result is *pattu* only. This may also be the case before *nūṟu*, *āyiram*, *ilaṭcam* and *kōṭi* where it is optional: it can be said *ōrāyiram* or *āyiram* itself (2.1). This will be taken care of by morphophonemic rules.

$$1.4 \quad NU_3 \longrightarrow NU_1 \text{ x } \textit{nūṟu} + \left(\begin{Bmatrix} NU_1 \\ NU_2 \end{Bmatrix} \right)$$

By this rule all the three digit numbers from 100 to 999 can be generated.

NU_1 x *nūru*

nūṟu
iraṇṭu X *nūṟu* > *irunūṟu* (2.3)
mūṉṟu X *nūṟu* > *muṉṉūṟu* (2.3, 3.1) etc.,

NU_1 x *nūru* + NU_1

nūṟṟu oṉṟu (101) (2.15)
irunūṟṟu iraṇṭu (202) (2.3, 15)

NU_1 x *nūru* + NU_2

nūṟṟup pattu (110)
nūṟṟup paṉṉiraṇṭu (112)

NU_2, as mentioned earlier, represent all the numbers of two digits, that is from 10 to 99. Therefore, when this is added to NU_1+*nūṟu* this would generate all the numbers, in the first case. 101-199, and then 201-299 etc.,

$$\textbf{1.5} \quad NU_4 \longrightarrow \begin{Bmatrix} NU_1 \\ NU_2 \end{Bmatrix} + \textit{āyiram} + \left(\begin{Bmatrix} NU_1 \\ NU_2 \\ NU_3 \end{Bmatrix} \right)$$

This rule takes care of all the numbers from 1000 to 99999. This rule can be split into two rules, having initial NU_1 and NU_2 separately. However it is clubbed together.

NU_1
NU_1 x āyiram

āyiram/ōrāyiram	(1000) (2.2)
oṉpatu āyiram	(9000)

NU_1 x *āyiram* + NU_1

āyirattu oṉṟu	(1001) (2.2; 2.16)
oṉpatu āyirattu oṉpatu	(9009) (2.16)

NU_1 x *āyiram* + NU_2

āyirattup pattu	(1010) (2.16)
oṉpatu āyirattu toṇṇūṟṟu oṉpatu	(9099) (2.15,16)

NU_1 x *āyiram* + NU_3

āyirattu nūṟu	
oṉpatu āyirattut toḷḷayirattu toṇṇūṟṟu oṉpatu	(9999) (2.9) (3.7) (2.15 ;2.16)

NU_2

NU_2 x *āyiram*

pattu x *āyiram* (10000)

toṇṇūṟṟu oṉpatu āyiram (99000) (2.9) (3.7)

NU_2 x *āyiram* + NU_1

pattu āyirattu oṉṟu (10001)

toṇṇūṟṟu oṉpatu āyirattu oṉpatu (99009) (2.9) (2.16)

NU_2 x *āyiram* + NU_2

pattu āyirattu oṉṟu (10001)

toṇṇūṟṟu oṉpatu āyirattu oṉpatu (99009) (2.9) (3.7)

NU_2 x *āyiram* + NU_3

pattu āyirattu nūṟu (10100)

toṇṇūṟṟu oṉpatu āyirattut toḷḷāyirattut toṇṇūṟṟu oṉpatu (99999) (2.9) (3.7)

1.6 $$NU_5 \longrightarrow \left\{ \begin{matrix} NU_1 \\ NU_2 \end{matrix} \right\} \text{ X } \textit{ilaṭcam} + \left(\left\{ \begin{matrix} NU_1 \\ NU_2 \\ NU_3 \\ NU_4 \end{matrix} \right\} \right)$$

This rule takes care of all the six- and seven-digit numbers, namely *(oru) ilaṭcam*, 100000 to 9999999.

NU_1

NU_1 x *ilaṭcam*

oru ilaṭcam (100000)

oṉpatu ilaṭcam (900000)

NU_1 x *ilaṭcam* + NU_1

oru ilaṭcattu oṉṟu (100001) (2.17)

oṉpatu ilaṭcattu oṉpatu (900009)

NU_1 x *ilaṭcam* +NU_2

oru ilaṭcattup pattu (100010)

oṉpatu ilaṭcattu toṇṇūṟṟu oṉpatu (900099)

NU_1 x *ilaṭcam* + NU_3

oru ilaṭcattu nūṟu (100100)

oṉpatu ilaṭcattut tonnūṟu oṉpatu āyirattut toḷḷāyirattut toṇṇūṟṟu oṉpatu (999999)

NU_1 x *ilaṭcam* + NU_4

oru ilaṭcattu āyiram (101000)

oṉpatu ilaṭcattut toll āyirattut toṇṇūṟṟu oṉpatu (900999)

NU_2

NU_2 x *ilaṭcam*

pattu ilaṭcam (1000000)

toṇṇūṟṟu oṉpatu ilaṭcam (9900000)

NU_2 x *ilatcam* + NU_1

pattu ilaṭcattu oṉṟu (1000001) (2.9)
toṇṇūṟṟu oṉpatu ilaṭcattu oṉpatu (9900009)

NU_2 x *ilatcam* + NU_2

pattu ilaṭcattup pattu (1000010)
toṇṇūṟṟu oṉpatu ilatcattat toṇṇūṟṟu oṉpatu (9900099)

NU_2 x *ilatcam* +NU_3

pattu ilaṭcattu nūṟu (1000100)
toṇṇūṟṟu oṉpatu ilaṭcattu tonnūṟṟu oṉpatu (9900999)

NU_2 x *ilatcattu* + NU_4

pattu ilaṭcattu āyiram (1001000)
toṇṇūṟṟu oṉpatu ilaṭcattu toṇṇūṟṟu oṉpatu āyirattu toḷḷāyirattu toṇṇūṟṟu oṉpatu (9999999)

$$NU_6 \longrightarrow \left\{ \begin{array}{l} NU_1 \\ NU_2 \\ NU_3 \\ NU_4 \\ NU_5 \\ (NU \end{array} \right\} X\textit{kōṭi}+ \quad \left(\left\{ \begin{array}{l} Nu_1 \\ Nu_2 \\ Nu_3 \\ Nu_4 \\ Nu_5 \end{array} \right\} \right)$$

This rule takes care of generation of not only 8 digit numbers and also all the possible numbers. *orukōṭi*, *pattuk kōṭi*, *nūṟu kōṭi*, *ilaṭcam kōṭi*, etc.

Note that this is a recursive rule which can go on generation *kōṭiyēkōṭi pattu kōṭiyē kōṭi* etc., by making use of the item NU_6.

This rule is something like S → (S) ab, which is considered as a recursive rule by Chomsky and other generative grammarians. This rule generates *kōṭi* to infinite.

NU1 x *kōṭi*	*orukōṭi* to *oṉpatu kōṭi*
NU2 x *kōṭi*	10000000 to 90000000
NU3 x *kōṭi*	100, 0000000 to 999, 0000000

Thus we can go on generating higher numbers and we add with these numbers NU1, NU2, NU3, NU4, etc., we can have numbers like 10000001 etc.

2. Allomorphic rules

It has been said that numbers like *oṉṟu*, *iraṇṭu* etc have various allomorphs and these variants can be taken care of by the following rules.

2.1

$$\textit{oṉṟu} \left(\begin{array}{l} \textit{nūṟu} \\ \textit{ilaṭcam} \\ \textit{kōṭi} \end{array} \right) \text{x} \quad => (\textit{oru}) \left(\begin{array}{l} \textit{nūṟu} \\ \textit{ilaṭcam} \\ \textit{kōṭi} \end{array} \right)$$

orunūṟu, nūṟu; ilaṭcam etc.,

2.2 *oṉṟu* x *āyiram* => (*ōr-*) *āyiram*

ōr-āyiram / āyiram

2.3

{ *iraṇṭu*, *mūṉṟu*, *nāṉku*, *aintu*, *āṟu*, *ēḻu*, *eṭṭu* } x { *pattu*, *nūṟu* } => { *iru*, *mu*, *nāl*, *aim*, *aru*, *eḻu*, *eṇ* } { *pattu*, *nūṟu* }

This rule accounts for

irupattu, muppattu, nāṟpattu, aimpattu, aṟupattu, eḻupattu, eṇpattu.
irunūṟu, muṉṉūṟu, nāṉūṟu, ainūṟu, aṟunūṟu, eḻunūṟu, eṇṇūṟu.

2.4 opt { *iraṇṭu*, *eṭṭu* } x *āyiram* => { *īr*, *eṇ* } *āyiram*

iraṇṭāyiram / īrāyiram
eṭṭāyiram / eṇṇāyiram

2.5 opt

mūṉṟu x *āyiram* => *mū-āyiram*
mūṉṟāyiram / mūvāyiram

2.6 opt

nāṉku>nālu

It is to be noted that often *nāṉku* and *nālu* are used without any distinction, one for the other.

nāṉku kōṭi / nālu kōṭi

2.7 opt

aintu x *āyiram* => *ai-āyiram*
aintāyiram / aiyāyiram

2.8 *ēḻu* x *āyiram* => *ēḻ-āyiram*

2.9

oṉpatu x { *pattu*, *nūṟu* } => *toḷ-* { *nūṟu*, *āyiram* }

toṇṇūṟu
toḷḷāyiram

2.10 opt.

pattu + *āyiram* > *patiṉ* - *āyiram*
pattāyiram, patiṉāyiram

2.11

$$\textit{oṉṟu} \times \textit{pattu} + \left[\begin{array}{l} \textit{oṉṟu} \\ \textit{mūṉṟu} \\ \textit{nāṉku} \\ \textit{aintu} \\ \textit{āru} \\ \textit{ēḻu} \\ \textit{eṭṭu} \end{array}\right] \Rightarrow \textit{oṉṟu} \times \textit{patiṉ} \left[\begin{array}{l} \textit{oṉṟu} \\ \textit{mūṉṟu} \\ \textit{nāṉku} \\ \textit{aintu} \\ \textit{āru} \\ \textit{ēḻu} \\ \textit{eṭṭu} \end{array}\right]$$

aintu

It is noted here that when the combinations of *oṉṟu* x *pattu* + *oṉṟu* etc., are generated, we have to get *patiṉoṉṟu* etc.; *pattu* has an allomorph *patiṉ* in this context. That is being taken care of by this rule.

2.12 *oṉṟu* x *pattu* + *iraṇṭu* => *oṉṟu+paṉ+iraṇṭu*

2.13

$$\textit{oṉṟu} \times \left[\begin{array}{l} \textit{pattu} \\ \textit{patiṉ} \\ \textit{paṉ} \end{array}\right] \Rightarrow \left[\begin{array}{l} \textit{pattu} \\ \textit{patiṉ} \\ \textit{paṉ} \end{array}\right]$$

pattu, patiṉoṉṟu, paṉṉiraṇṭu.

2.14

$$\left[\begin{array}{l} \textit{iru} \\ \textit{mu} \\ \textit{nāl} \\ \textit{aim} \\ \textit{aṟu} \\ \textit{eḻu} \\ \textit{eṇ} \end{array}\right] \times \textit{pattu} \left[\begin{array}{c} \# \\ \times \end{array}\right] \Rightarrow \left[\begin{array}{l} \textit{iru} \\ \textit{mu} \\ \textit{nāl} \\ \textit{aim} \\ \textit{aṟu} \\ \textit{eḻu} \\ \textit{en} \end{array}\right] \textit{-patu} \left[\begin{array}{c} \# \\ \times \end{array}\right]$$

Note that when *irupattu* etc are having changes, *pattu* > *patu* when they are followed by no other integer numbers or when they are in multiplication relation.

iru x *pattu* x *āyiram* > *irupatu āyiram*

irupatu twenty

2.15 *nūṟu* +=> *nūṟṟu*
nūṟu + *irupatu* > *nūṟṟu irupatu*

2.16 *āyiram*+=>*āyirattu*
āyirattu nūṟu

2.17 *ilaṭcam*+=>*ilaṭcattu*
ilaṭcattu nāṉku

2.18 *kōṭi* +=> *kōṭiyē*

orukōṭiyē nāṉku

III Sandhi rules

3.1 *mu* - C > *mu* CC
muppatu, muppattu, munnūṟu

3.2 *ai*-V > *aiy*V
aiyāyiram

3.3 *ṇ-n* > *ṇṇ*
eṇ-nūṟu > *eṇṇūṟu*
eṇ-nūṟṟu > *eṇṇūṟṟu*

3.4 *m-n* > *nn*
aim-nūṟu > *ainnūṟu*

3.5 *l*-Stop > *ṟ*-Stop
nāl-patu > *nāṟpatu*

3.6 *l-n* > *ṉ*
nāl-nūṟu > *nāṉūṟu*

3.7 *ḷ-n* > *ṇṇ*
toḷ-nūṟu > *toṇṇūṟu*

3.8 *ṉ*+*n* > *ṉ*
patiṉ-nāṉku > *patiṉāṉku*

3.9 #(C) VC-V > #(C) VCCV
toḷ-āyiram > *toḷḷāyiram*
paṉ-iraṇṭu > *paṉṉiraṇṭu*

$$\begin{bmatrix} \text{PP} \\ \text{NP} \\ \text{VP} \end{bmatrix} u + \text{V} > \begin{bmatrix} \text{PP} \\ \text{NP} \\ \text{VP} \end{bmatrix} \text{-V}$$

aintu + *āyiram* > *aintāyiram*
mūṉṟu + *āyiram* > *mūnrāyiram*
eṭṭu + *āyiram* > *eṭṭāyiram*
āṟu + *āyiram* > *āṟāyiram*

Grammar

It is defined by generative grammarians that "a grammar is a finite set of rules which generates an infinite number of grammatical sentences and no ungrammatical ones".

With the foregoing finite number of rules it is possible to generate thousands of numbers.

An Electronic grammar is to be thought of in this line. It should be possible to produce large numbers of grammatical sentences and the joint work of linguists and software engineers should make it possible to produce such a grammar.

Bibliography

S. Agesthialingom, 1967. A Generative Grammar of Tamil, Annamalai University, A. Nagar.
Chomsky, Noam A. 1957. Syntactic Structures, The Hague.
Chomsky, 1965. Aspects of Theory of Syntax, MIT Press, Cambridge.
Davis Garden, B. 1969. Computer Processing McGraw Hill, Inc. Sydney, Rep. 1986.
Grishman, R. 1986. Computational Linguistics, An Introduction, Cambridge.
Pon. Kothandaraman. A Grammar of Contemporary Literary Tamil. IITS, Chennai.