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 $o\underline{n}\underline{r}\bar{e}$ $k\bar{a}l$ (1 ½) $o\underline{n}\underline{r}arai$ (1 ½) are not dealt with. It is to be mentioned here that they can be easily taken care of by certain rules like

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

Where NUM_{comp} stands for complex numbers and NUM_{F} for integer numbers like $o\underline{n}\underline{r}u$, $ira\underline{n}\underline{t}u$ etc., FRA for fractions like $k\bar{a}l$ (1/4), arai (1/2), $mukk\bar{a}l$; (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 *nūṛu* "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	ทนิ <u>r</u> น	<u>āyiram</u>		ilaṭcam		kōṭi
Nu_1	Nu ₂	Nu ₃	Nu_4		Nu_5		Nu ₆

Since $\bar{a}yiram$ 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 NU₄. This is also the case with *ilaṭcam*. NU₅ represents both six-digits and seven-digits.

All these digits or all the numbers can be generated by a rule like the following: 1.1

$$\begin{array}{c|c}
NU_1 \\
NU_2 \\
NU_3 \\
NU_4 \\
NU_5 \\
NU_6
\end{array}$$

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

- NU₁ stands for onru "one" to onpatu "9"
- NU2 stands for pattu "ten" to toṇṇūrṛu oṇpatu (99)
- NU₃ stands for nūru "hundred" to toḷḷāyirattu toṇṇūṛru oṇpatu (999)
- NU₄ stands for āyiram "thousand" to tonnūrru onpatu āyirattu toļļāyirattu tonnūrru onpatu (99999)
- NU₅ stands for *ilaṭcam* "lakh" oru ilaṭcam" (100000 to 9999999)
- NU₆ stands for *kōṭi* "crore" "oru kōṭi to infinite" (10000000 to infinite).

Three kinds of rules

This grammar consists of three kinds of rules namely

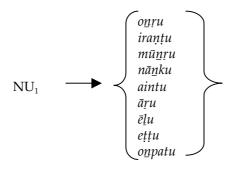
- (1) generation rules (I)
- (2) allomorphic rules (II)
- (3) sandhi rules (III).

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 stands for numbers $o\underline{n}\underline{r}u$ (1) to $o\underline{n}\underline{p}atu$ (9) i.e. one to nine. NU_1 is $o\underline{n}\underline{r}u$, $ira\underline{n}\underline{t}u$ or $m\underline{u}\underline{n}\underline{r}u$, etc. All these are represented by NU_1

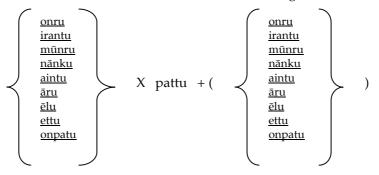
It is known that all these numbers have allomorphs and they will be taken care of by appropriate morphophonemic rules (II).

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

1.3
$$NU_2 \rightarrow NU_1 \times pattu + (NU_1)$$

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 NU₁. This can be shown in something like the following.



So this rule can generate

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

Similarly if we add NU₁ after pattu we will get

```
onru x pattu + onru > patinonru (2.11;13)
onru x pattu + irantu > pannirantu (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.

```
o<u>n</u>ru × pattu > pattu "ten" (2.13)
iranṭu × pattu > irupatu "Twenty" (2.3,14)
```

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

```
pattu + o\underline{n}\underline{r}u > pati\underline{n}o\underline{n}\underline{r}u (11) (2.11,13)
```

It is to be noted that it is the same in the case of nūṛu, 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

```
iranṭu x pattu > irupatu (20) (2.3; 14) irupatu āyiram (multiplication) irupattu iranṭu (addition) (2.3)
```

Deletion of onru

When $o\underline{n}\underline{r}u$ comes before pattu (ten) $o\underline{n}\underline{r}u$ is deleted (2.13) and the result is pattu only. This may also be the case before $n\underline{u}\underline{r}u$, $\underline{u}\underline{v}\underline{r}am$, $ilat\underline{c}am$ and $k\underline{o}ti$ where it is optional: it can be said $\underline{o}\underline{r}\underline{a}\underline{v}\underline{r}am$ or $\underline{u}\underline{v}\underline{r}am$ itself (2.1). This will be taken care of by morphophonemic rules.

norphophonemic rules.

$$1.4 \text{ NU}_3$$
 \longrightarrow $NU_1 \times n\bar{u}\underline{r}u + \left\{\begin{array}{c} NU_1 \\ NU_2 \end{array}\right\}$

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

$NU_1 \times n\bar{u}ru$

пūŗи

iranṭu X nūṛu > irunūṛu (2.3) mūṇṛu X nūṛu > muṇṇūṛu (2.3, 3.1) etc.,

$NU_1 \times n\bar{u}ru + NU_1$

nūṛṛu oṇṛu (101) (2.15) irunūṛṛu iraṇṭu (202) (2.3, 15)

$NU_1 \times n\bar{u}ru + NU_2$

nūṛṛup pattu (110) nūṛṛup paṇṇiraṇṭu (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 $NU_1 + n\bar{u}ru$ this would generate all the numbers, in the first case. 101-199, and then 201-299 etc.,

1.5
$$NU_4 \longrightarrow \begin{cases} NU_1 \\ NU_2 \end{cases} + \bar{a}yiram + \left\{ \begin{cases} NU_1 \\ NU_2 \\ NU_3 \end{cases} \right\}$$

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.

NU₁
<u>NU₁ x āyiram</u> *āyiram/ōrāyiram onpatu āyiram*

āyiram/ōrāyiram (1000) (2.2) o<u>n</u>patu āyiram (9000)

 $\underline{NU_1} \times \underline{ayiram} + \underline{NU_1}$

 āyirattu oṇru
 (1001) (2.2; 2.16)

 oṇpatu āyirattu oṇpatu
 (9009) (2.16)

 $NU_1 \times \bar{a}yiram + NU_2$

 āyirattup pattu
 (1010) (2.16)

 oṇpatu āyirattu toṇṇūṛṛu oṇpatu
 (9099) (2.15,16)

$NU_1 \times \bar{a}yiram + NU_3$

āyirattu nū<u>r</u>u

onpatu āyirattut toļļayirattu toņņūrru onpatu (9999) (2.9) (3.7) (2.15;2.16)

 NU_2

NU₂ x āyiram

 $pattu \times \bar{a}yiram$ (10000)

toṇṇūṛru oṇpatu āyiram (99000) (2.9) (3.7)

 $NU_2 \times \bar{a}yiram + NU_1$

pattu āyirattu o<u>n</u>ru (10001)

toṇṇūṛru oṇpatu āyirattu oṇpatu (99009) (2.9) (2.16)

 $NU_2 \times \bar{a}yiram + NU_2$

pattu āyirattu o<u>n</u>ru (10001)

toṇṇūṛṛu oṇpatu āyirattu oṇpatu (99009) (2.9) (3.7)

 $NU_2 \times \bar{a}yiram + NU_3$

pattu āyirattu nūṛu (10100) toṇṇūṛṛu oṇpatu āyirattut toḷḷāyirattut toṇṇūṛṛu oṇpatu (99999) (2.9) (3.7)

1.6 $NU_5 \longrightarrow \begin{cases} NU_1 \\ NU_2 \end{cases}$ $X ilatcam + \left(\begin{cases} NU_1 \\ NU_2 \\ NU_3 \\ NU_4 \end{cases} \right)$

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

 \underline{NU}_1

 $NU_1 \times ilatcam$

oru ilaṭcam (100000) oṇpatu ilaṭcam (900000)

 $NU_1 \times ilatcam + NU_1$

oru ilaṭcattu oṇru (100001) (2.17) oṇpatu ilaṭcattu oṇpatu (900009)

 $NU_1 \times ilatcam + NU_2$

oru ilaṭcattup pattu (100010) oṇpatu ilaṭcattu toṇṇūṛṛu oṇpatu (900099)

 $NU_1 \times ilatcam + NU_3$

oru ilaṭcattu nūṛu (100100)

onpatu ilaţcattut tonnūru onpatu āyirattut toļļāyirattut toņnūrru onpatu (999999)

 $NU_1 \times ilatcam + NU_4$

oru ilaṭcattu āyiram (101000) oṇpatu ilaṭcattut toll āyirattut toṇṇūrru oṇpatu (900999)

NU₂

NU₂ x ilatcam

pattu ilaṭcam (1000000) toṇṇūṛru oṇpatu ilaṭcam (9900000)

 $\underline{NU_2} \underline{x} i la \underline{t} cam + \underline{NU_1}$

pattu ilaṭcattu oṇru (1000001) (2.9) toṇṇūṛru oṇpatu ilaṭcattu oṇpatu (9900009)

 $NU_2 \times ilatcam + NU_2$

pattu ilaṭcattup pattu (1000010) toṇṇūṛru oṇpatu ilatcattat toṇṇūṛru oṇpatu (9900099)

 $NU_2 \times ilatcam + NU_3$

pattu ilaṭcattu nūru (1000100) toṇṇūrṛu oṇpatu ilaṭcattu tonnūrṛu oṇpatu (9900999)

NU₂ x ilatcattu + NU₄

pattu ilaṭcattu āyiram (1001000)

tonnūrru onpatu ilatcattu tonnūrru onpatu āyirattu toļļāyirattu tonnūrru onpatu (9999999)

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

Note that this is a recursive rule which can go on generation $k\bar{o}tiy\bar{e}k\bar{o}ti$ pattu $k\bar{o}tiy\bar{e}k\bar{o}ti$ etc., by making use of the item NU_6 .

This rule is something like S \longrightarrow (S) ab, which is considered as a recursive rule by Chomsky and other generative grammarians. This rule generates $k\bar{o}ti$ to infinite.

 NU1 x kōṭi
 orukōṭi to oṇpatu kōṭi

 NU2 x kōṭi
 10000000 to 90000000

 NU3 x kōṭi
 100, 0000000 to 999, 0000000

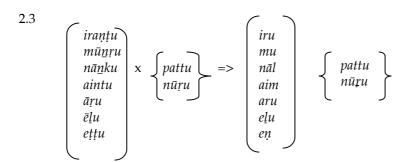
Thus we can go on generating higher numbers and we add with these numbers NU1, NU2, NU3, NU4, etc., we can have numbers like 10000001 etc.

2. Allomorphic rules

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

$$\begin{array}{ccc}
o\underline{n}\underline{r}u & \begin{pmatrix} n\underline{u}\underline{r}u \\ ila\underline{t}cam \\ k\underline{o}\underline{t}i \end{pmatrix} & x & => (oru) & \begin{pmatrix} n\underline{u}\underline{r}u \\ ila\underline{t}cam \\ k\underline{o}\underline{t}i \end{pmatrix}$$

2.2 o<u>n</u>ru x āyiram => (ōr-) āyiram ōr-āyiram/āyiram



This rule accounts for

irupattu, muppattu, nāṛpattu, aimpattu, arupattu, elupattu, enpattu. irunūṛu, muṇṇūru, nāṇūru, ainūru, arunūru, elunūru, eṇṇūru.

2.4 opt
$$\begin{pmatrix} iranțu \\ etțu \end{pmatrix}$$
 \times $\bar{a}yiram => \begin{pmatrix} \bar{t}r \\ en \end{pmatrix}$ $\bar{a}yiram$

iraṇṭāyiram / īrāyiram eṭṭāyiram / eṇṇāyiram

2.5 opt

2.6 opt

nā<u>n</u>ku>nālu

It is to be noted that often $n\bar{a}\underline{n}ku$ and $n\bar{a}lu$ are used without any distinction, one for the other.

nānku kōţi / nālu kōţi

2.7 opt

aintu x āyiram => ai-āyiram aintāyiram / aiyāyiram

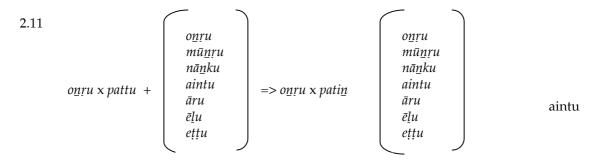
2.8 $\bar{e}\underline{l}u \times \bar{a}yiram => \bar{e}\underline{l}-\bar{a}yiram$

2.9
$$o\underline{n}patu \times \left(\begin{array}{c} pattu \\ n\bar{u}ru \end{array} \right) => \qquad to!- \left(\begin{array}{c} n\bar{u}ru \\ \bar{a}yiram \end{array} \right)$$

$$to!, \bar{u}ru \\ to!, \bar{u}ru \\ to!,$$

2.10 opt.

pattu + āyiram > pati<u>n</u> - āyiram pattāyiram, pati<u>n</u>āyiram



It is noted here that when the combinations of $o\underline{n}\underline{r}u \times pattu + o\underline{n}\underline{r}u$ etc., are generated, we have to get $pati\underline{n}o\underline{n}\underline{r}u$ etc.; pattu has an allomorph $pati\underline{n}$ in this context. That is being taken care of by this rule.

2.12
$$o\underline{n}\underline{r}u \times pattu + ira\underline{n}\underline{t}u => o\underline{n}\underline{r}u + pa\underline{n} + ira\underline{n}\underline{t}u$$

$$\begin{array}{cccc}
2.13 & & & & & pattu \\
o\underline{n}\underline{r}u & \times & & patt\underline{n} \\
pa\underline{n} & & pat\underline{n} & pat\underline{n}
\end{array} => \qquad \begin{array}{cccc}
pattu \\
pati\underline{n} \\
pa\underline{n}
\end{array}$$

pattu, patinonru, panniranțu.

$$\begin{pmatrix}
iru \\
mu \\
n\bar{a}l \\
aim \\
aru \\
elu \\
en
\end{pmatrix}$$
 \times pattu
$$\begin{pmatrix}
\# \\
\chi
\end{pmatrix}$$
 $=>$

$$\begin{pmatrix}
iru \\
mu \\
n\bar{a}l \\
aim \\
aru \\
elu \\
en
\end{pmatrix}$$
 $-patu$

$$\begin{pmatrix}
\# \\
\chi
\end{pmatrix}$$

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.

$$iru \times pattu \times \bar{a}yiram > irupatu \bar{a}yiram$$
 $irupatu$ twenty

- 2.16 āyiram+=>āyirattu āyirattu nūṛu
- 2.17 ilaṭcam+=>ilaṭcattu ilaṭcattu nānku

2.18
$$k\bar{o}ti +=> k\bar{o}tiy\bar{e}$$

orukōṭiyē nāṇku

III Sandhi rules

- 3.1 mu C > mu CCmuppatu, muppattu, munnūru
- 3.2 ai-V > aiyV aiyāyiram

- 3.5 $l ext{-Stop} > r ext{-Stop}$ $n\bar{a}l ext{-}patu > n\bar{a}rpatu$

- 3.8 $\underline{n}+n > \underline{n}$ $pati\underline{n}-n\underline{a}\underline{n}ku > pati\underline{n}\underline{a}\underline{n}ku$
- 3.9 #(C) VC-V > #(C) VCCV tol-āyiram > tol-lāyiram pan-iranṭu > panniranṭu

$$\left(\begin{array}{c}
PP \\
NP \\
VP
\end{array}\right) u + V > \left(\begin{array}{c}
PP \\
NP \\
VP
\end{array}\right) - V$$

aintu + āyiram > aintāyiram mūṇṛu + āyiram > mūnrāyiram eṭṭu + āyiram > eṭṭāyiram āṛu + āyiram > āṛāyiram

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.