Generating Polysyllabic Mandarin Onomatopoeia by Reduplication — a Combination of Bottom-up and Top-down Finite-state Tree Transducers

Jing Ji

1 Introduction

Most polysyllabic onomatopoeia in Mandarin derive from monosyllabic or disyllabic ones by reduplication, which is reflected in segmental similarity between syllables. For example, the word [pʰi li pʰa la] originates from [pʰa] by reduplicating twice, making the first and second foot with the same pair of onset and syllables in the same foot with the same rhyme. Previous studies have summarized several patterns of reduplicative onomatopoeia (Zhu Dexi, 1982; Wang Hongjun, 1996; Shi Yuzhi, 1995). However, their description of phonotactic constraints is insufficient to cover all the well-formed segmental changes during reduplication.

This paper makes analysis of different patterns of polysyllabic onomatopoeia stemming from reduplication, defines phonotactic constraints for segmental changes and applies deterministic finite-state tree transducers to generate well-formed polysyllabic onomatopoeia. It turns out that due to syllable structure and coarticulation, there exist two distinct kinds of segmental variation during reduplication, which can be processed by top-down and bottom-up tree transducers respectively. Therefore, the final generator combines deterministic bottom-up finite-state tree transducers (DBUTT) and top-down ones (DTDTT) together to generate all the well-formed polysyllabic patterns with monosyllabic or disyllabic words as input.

2 Reduplication Patterns of Mandarin Onomatopoeia

Shao Jingmin (1981) proposed two types of patterns of Mandarin onomatopoeia according to the way of formation. Examples are listed in Table1 and Table2.

Table 1. Monosyllable and Its Reduplication Pattern

Pattern	Example	IPA
A	哗 (sound of water flow)	[xua]
AA	汪汪 (barking)	[uaŋ uaŋ]
AAA	噔噔噔 (sound of going upstairs)	[təŋ təŋ təŋ]
AAAA	砰砰砰砰 (knock at door)	[pʰəŋ pʰəŋ pʰəŋ pʰəŋ]

Table 2. Disyllable and Its Reduplication Pattern

Pattern	Example	IPA
AB	哗啦; 滴答; 咕嘟; 噼啪	[xua la]; [ti ta]; [ku tu]; [pʰi pʰa]
AABB	滴滴答答 (sound of rain drops)	[ti ti ta ta]
ABAB	咕嘟咕嘟 (sound of drinking water)	[ku tu ku tu]
ABB	哗啦啦 (sound of water flow)	[xua la la]
AAB	滴滴答 (sound of rain drops)	[ti ta ta]
CDAB	稀里哗啦 (broken sound of glassware)	[ɕi li xua la]
ACBD	劈里啪啦 (sound of firecrackers)	[pʰi li pʰa la]

In Table 1, pattern A is regarded to be the base whereas in Table 2, AB serves as base pattern. However, considering the derivational relationship between two syllables in AB (always alliterative or rhyming), patterns in Table 2 actually derive from monosyllabic pattern A by reduplicating twice. AB can only be taken as the original base when it is a disyllabic single-morpheme word, meaning there is no derivational relationship between syllables. Figure 1 further illustrates the derivational relationship between patterns in the above table.

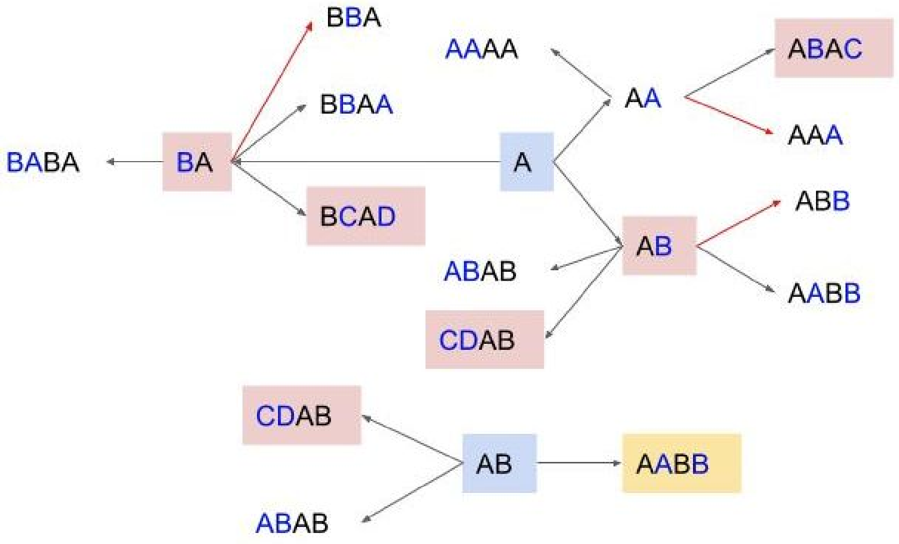


Figure 1. Derivational Relationship between Reduplication Patterns

As is shown in the graph above, A and AB in blue box are the original base forms and all others are derivational forms. For each pattern, blue color is used for reduplicants. Rightward arrows indicate postpositional reduplicants whereas leftward arrows represent prepositional ones, with black arrows for total reduplication and red for partial reduplication. The number of arrows away from the base pattern equals to the times of reduplication. Patterns in pink box are those going through segmental changes during reduplication.

It is worth noting that prepositional reduplication (leftward arrow) always takes the whole word as base, like CDAB, whereas postpositional reduplication (rightward arrow) transforms each syllable separately, such as BDAC. AAAA, ABAB and BABA patterns, however, can be regarded as either prepositional or postpositional. For the sake of convenience, they are considered as prepositional reduplication during process.

3 Constraints of Reduplication Patterns

3.1 Constraints of Segmental Changes by OT

Constraints of reduplication fall into three categories, markedness, correspondence and alignment constraints. Different patterns result from different ranking of constraints. Markedness constraints are defined below to explain well-formed segmental variation during reduplication.

Differ-BR-N: Decrease the sonority of nuclea to the utmost in the prepositional reduplicant. Assign one violation mark for every nucleus in the prepositional reduplicant that isn't lowered in sonority or isn't lowered to the utmost compared with the base.

Differ-BR-O: Increase the sonority of onset to the utmost in postpositional reduplicant. Assign one violation mark for every onset in the postpositional reduplicant that isn't increased in sonority or isn't increased to the utmost compared with the base.

According to the constraints, [ti ta] cannot go through CDAB pattern to become [ti ti ti ta] because the sonority of the first vowel is not lowered. Also, [ti li ta la] instead of [tu lu ta la] is the well-formed reduplication since the vowel in the reduplicant in [tu lu ta la] is not lowered to the utmost.

3.2 Hierarchy of Restriction on Base Form

As is shown in the previous analysis, either monosyllabic or disyllabic word can be the base of reduplication. There is no precondition for reduplication from monosyllable to disyllable. For trisyllabic and quadrisyllabic onomatopoeia, however, different patterns form a hierarchy with regard to the degree of restriction on the base underlined in the following table.

Table 3. Restriction Hierarchy of Trisyllabic and Quadrisyllabic Patterns

Patterns	Restriction on Base
ABAB/BABA/BBAA	none
AABB	representing continuous sound
BBA	same onset
ABB	same rhyme
CDAB	same rhyme; nucleus ≠ [i],[y],[ɿ]
BDAC	same onset; onset ≠ [l]; each syllable = morpheme
ABAC	same syllable; onset ≠ [l]; each syllable = morpheme

4 Process by Deterministic Finite-state Tree Transducers

In order to be processed by tree transducers, all the input words were transformed to trees of two depth, with three layers representing word(“W”), syllable(“S”) and phoneme(IPA) respectively. The input trees were named after Pinyin of the words. For instance, the word [pa ta] was transformed as follows.

```
b = Tree(‘p’); a = Tree(‘a’); d = Tree(‘t’)
ba = Tree(‘S’)
ba.add_subtree(b)
ba.add_subtree(a)
da = Tree(‘S’)
da.add_subtree(d)
da.add_subtree(a)
bada = Tree(‘W’)
bada.add_subtree(ba)
bada.add_subtree(da)
```

4.1 Representing Position-based Segmental Changes by DTDTT

As the syllable structure of reduplicative Mandarin onomatopoeia is comparatively simple, either CV(e.g. [ta]) or CVN(e.g. [tiŋ]), and the segmental changes are dependent on the position of reduplicant in a word, a set of top-down finite-state tree transducers (dtdtt1 -- dtdtt7) are defined to process reduplication. The starting state for all the top-down transducers is set to be “qf”.

4.1.1 Prepositional Reduplication

Dtdtt1 and dtdtt2 are created for prepositional reduplication. The composition of them maps A to BA or AB to CDAB. Examples are shown in Figure 2. Dtdtt1 takes a word with one or two syllables and makes a prepositional copy while dtdtt2 takes in the output of dtdtt1 and changes the vowel in the reduplicants (with state of “qi”) to [i]. According to **Differ-BR-N,** the vowel of the input of dtdtt2 cannot be [+high] and [-back] ([i], [y], [ɿ]), which is specified in the input alphabet.

$$\text{[pa]} \xrightarrow{\text{dtdtt1}} \text{[pa } \underline{\text{pa}}\text{]} \xrightarrow{\text{dtdtt2}} \text{[pi } \underline{\text{pa}}\text{]}$$

$$\text{[pa ta]} \xrightarrow{\text{dtdtt1}} \text{[pa ta } \underline{\text{pa ta}}\text{]} \xrightarrow{\text{dtdtt2}} \text{[pi ti } \underline{\text{pa ta}}\text{]}$$

Figure 2. Examples of Prepositional Reduplication

4.1.2 Postpositional Reduplication

Dtdtt3 and dtdtt4 are used for postpositional reduplication. Postpositional reduplication differs from prepositional one in that it aligns every foot, rather than prosodic word, with a morpheme from stem at the left edge. Instead of mapping A with AA and AB with ABAB in dtdtt1, dtdtt3 changes A to AA and AB to AABB. Correspondingly, dtdtt4 changes the initial consonant in reduplicants to [l], the most sonorant consonant in Mandarin. Composition of dtdtt3 and dtdtt4 makes A into AB or AB into ACBD as is shown in Figure 3. Similar to dtdtt2, dtdtt4 excludes [l] from the input alphabet to satisfy precondition.

$$[\text{p}^{\text{h}}\text{a}] \xrightarrow{\text{dtdtt3}} [\underline{\text{p}^{\text{h}}\text{a}}\ \text{p}^{\text{h}}\text{a}] \xrightarrow{\text{dtdtt4}} [\underline{\text{p}^{\text{h}}\text{a}}\ \text{la}]$$

$$[\text{p}^{\text{h}}\text{i p}^{\text{h}}\text{a}] \xrightarrow{\text{dtdtt3}} [\underline{\text{p}^{\text{h}}\text{i}}\ \text{p}^{\text{h}}\text{i}\ \underline{\text{p}^{\text{h}}\text{a}}\ \text{p}^{\text{h}}\text{a}] \xrightarrow{\text{dtdtt4}} [\underline{\text{p}^{\text{h}}\text{i}}\ \text{li}\ \underline{\text{p}^{\text{h}}\text{a}}\ \text{la}]$$

Figure 3. Examples of Postpositional Reduplication

Dtdtt7 is also created for postpositional reduplication. It receives the quadrisyllabic output of reduplication by dtdtt3 twice, which is AAAA, and turns it into ABAC. Different from dtdtt4 which only changes the consonant of reduplicants, dtdtt7 changes the consonant of each reduplicant to [l] and also changes vowel of the first reduplicant to [i], which is depicted by the example below.

$$\text{[xua]} \xrightarrow{\text{dtdtt3}} [\underline{\text{xua}}\ \text{xua}] \xrightarrow{\text{dtdtt3}} [\underline{\text{xua}}\ \text{xua}\ \underline{\text{xua}}\ \text{xua}] \xrightarrow{\text{dtdtt7}} [\underline{\text{xua}}\ \text{li}\ \underline{\text{xua}}\ \text{la}]$$

Figure 4. An example of ABAC transformation

4.1.3 Partial Reduplication

As previous transducers cannot deal with partial reduplication, dtdtt5 and dtdtt6 are created accordingly. Dtdtt5 takes a disyllabic word BA, which is the output of composition of dtdtt1 and dtdtt2, and makes a postpositional copy after the first syllable (<u>B</u>B<u>A</u>). Likewise, dtdtt6 takes the disyllabic output of composition of dtdtt3 and dtdtt4 (AB) and makes a copy after the second syllable ($\underline{\text{AB}}$B). Examples are shown below.

$$[\text{ta}] \xrightarrow{\text{dtdtt1}} [\text{ta } \underline{\text{ta}}] \xrightarrow{\text{dtdtt2}} [\text{ti } \underline{\text{ta}}] \xrightarrow{\text{dtdtt5}} [\underline{\text{ti}} \text{ ti } \underline{\text{ta}}]$$

$$[\text{xua}] \xrightarrow{\text{dtdtt3}} [\underline{\text{xua}} \text{ xua}] \xrightarrow{\text{dtdtt4}} [\underline{\text{xua}} \text{ la}] \xrightarrow{\text{dtdtt6}} [\underline{\text{xua la}} \text{ la}]$$

Figure 5. Examples of Partial Reduplication

4.2 Representing Coarticulation by DBUTT

Since the vowel in prepositional reduplicant becomes [i], the dorsal consonants [k], [k^h], [x] before [i] become palatalized ones [tɕ], [$tɕ^h$], [ɕ]. Meanwhile, [tʂ], [$tʂ^h$], [ʂ] become [ts], [ts^h], [s] with feature of [+anterior]. These accompanied changes in consonants have already been realized in transitions of dtdtt2. Nevertheless, due to phonotactic rules in Mandarin, it is [ɿ] rather than [i] could occur after [ts], [ts^h], [s], so [i] in the reduplicant after [ts], [ts^h], [s] needs to be further changed to [ɿ]. This conditional segmental change dependent on adjacent segments cannot be realized by top-down tree transducer because it assigns state to each position independently. Therefore, a bottom-up tree transducer dbutt2 is introduced as a complement of dtdtt2.

Dbutt2 takes the output of dtdtt2 as input. It assigns state “qz” to segments of [ts], [ts^h], [s] and assigns “qi” state to [i]. In its transitions, the state of a syllable is “qz” if it has both “qz” and “qi” as children, and changes it to a tree with the second child of [ɿ]. Therefore, the complete prepositional reduplication is the composition of dtdtt1, dtdtt2 and dbutt2, as is shown in Figure 6.

$$[\text{ʂa}] \xrightarrow{\text{dtdtt1}} [\text{ʂa } \underline{\text{ʂa}}] \xrightarrow{\text{dtdtt2}} [\text{si } \underline{\text{ʂa}}] \xrightarrow{\text{dbutt2}} [\text{sɿ } \underline{\text{ʂa}}]$$

Figure 6. Example of Prepositional Reduplication with Coarticulation

5 Testing with Monosyllabic and Disyllabic Onomatopoeia

5.1 Compositional Functions for Patterns

Before testing, 14 functions were defined for each reduplication pattern with monosyllable as input. For instance, the function generating CDAB word from A was realized by composition of postpositional and prepositional reduplication, as is illustrated in the following figure. It returns empty string if transformation is undefined. Otherwise, show() method will be applied to the output tree.

$$[\text{xua}] \xrightarrow{\text{dtdtt3}} [\underline{\text{xua}}\ \text{xua}] \xrightarrow{\text{dtdtt4}} [\underline{\text{xua}}\ \text{la}] \xrightarrow{\text{dtdtt1}} [\text{xua la}\ \underline{\text{xua la}}] \xrightarrow{\text{dtdtt2}} [\text{ɕi li}\ \underline{\text{xua la}}] \xrightarrow{\text{dbutt2}} [\text{ɕi li}\ \underline{\text{xua la}}]$$

Figure 7. Example of transformation by CDAB function

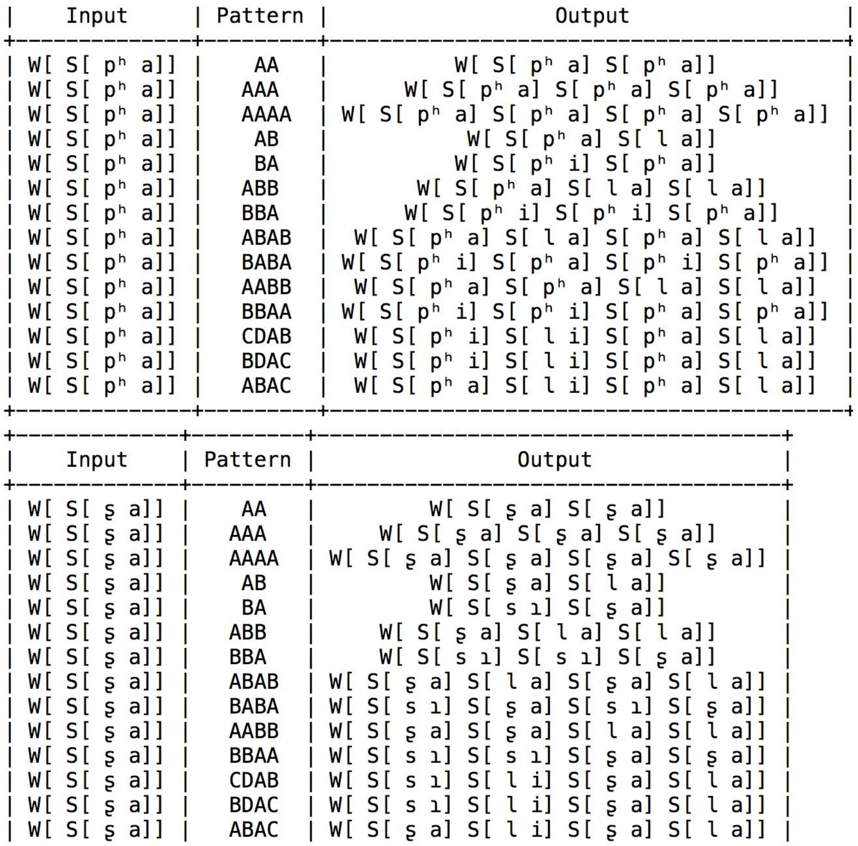
For disyllabic base AB, only CDAB, ABAB and AABB patterns are allowed in reduplication, with the precondition of rhyming for CDAB and continuous meaning for AABB. AA function for monosyllabic input can be applied to generate ABAB pattern with AB as input. Similarly, BA function can be used for generating CDAB from AB.

5.2 Lists of Input Trees

Monosyllabic input list includes 20 tokens of monosyllabic Mandarin onomatopoeia in tree structure, which covers most of the typical monosyllabic onomatopoeia. Disyllabic lists, however, are defined separately for continuous sound and temporary sound since the precondition of AABB pattern from AB requires continuous meaning in AB. Both rhyming and non-rhyming tokens are contained in disyllabic lists.

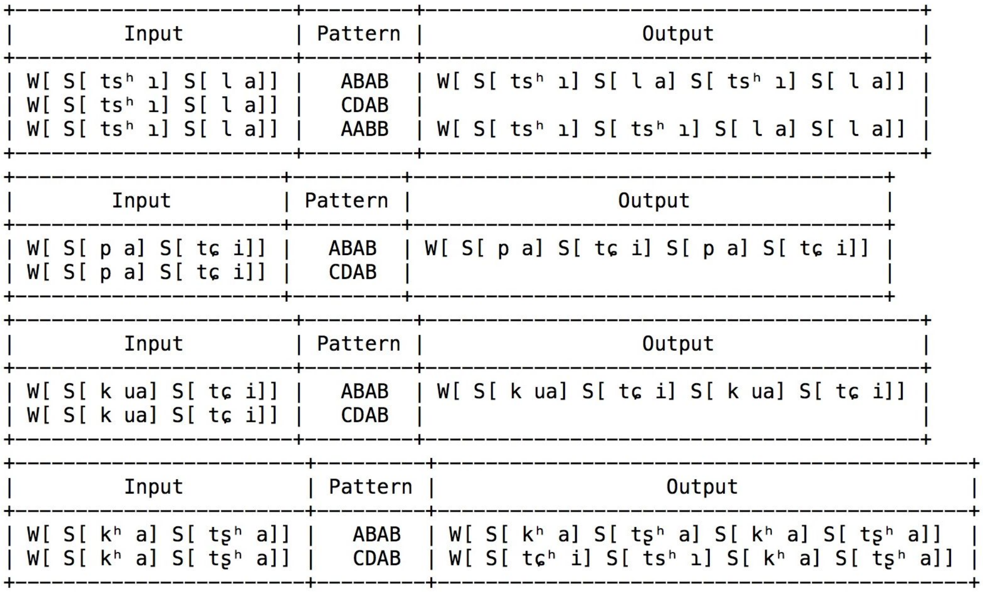
5.3 Results

In terms of monosyllabic input, all the 14 possible patterns are shown in the output, with empty indicating unsatisfied precondition. As for disyllabic input, 3 patterns are applied to input with continuous meaning, 2 for temporary ones. The excerpt of output is shown as follows.



```
|    Input     | Pattern |                 Output                  |
+--------------+---------+-----------------------------------------+
| W[ S[ pʰ a]] |   AA    |          W[ S[ pʰ a] S[ pʰ a]]          |
| W[ S[ pʰ a]] |   AAA   |     W[ S[ pʰ a] S[ pʰ a] S[ pʰ a]]      |
| W[ S[ pʰ a]] |  AAAA   | W[ S[ pʰ a] S[ pʰ a] S[ pʰ a] S[ pʰ a]] |
| W[ S[ pʰ a]] |   AB    |          W[ S[ pʰ a] S[ l a]]           |
| W[ S[ pʰ a]] |   BA    |          W[ S[ pʰ i] S[ pʰ a]]          |
| W[ S[ pʰ a]] |   ABB   |      W[ S[ pʰ a] S[ l a] S[ l a]]       |
| W[ S[ pʰ a]] |   BBA   |     W[ S[ pʰ i] S[ pʰ i] S[ pʰ a]]      |
| W[ S[ pʰ a]] |  ABAB   |  W[ S[ pʰ a] S[ l a] S[ pʰ a] S[ l a]]  |
| W[ S[ pʰ a]] |  BABA   | W[ S[ pʰ i] S[ pʰ a] S[ pʰ i] S[ pʰ a]] |
| W[ S[ pʰ a]] |  AABB   |  W[ S[ pʰ a] S[ pʰ a] S[ l a] S[ l a]]  |
| W[ S[ pʰ a]] |  BBAA   | W[ S[ pʰ i] S[ pʰ i] S[ pʰ a] S[ pʰ a]] |
| W[ S[ pʰ a]] |  CDAB   |  W[ S[ pʰ i] S[ l i] S[ pʰ a] S[ l a]]  |
| W[ S[ pʰ a]] |  BDAC   |  W[ S[ pʰ i] S[ l i] S[ pʰ a] S[ l a]]  |
| W[ S[ pʰ a]] |  ABAC   |  W[ S[ pʰ a] S[ l i] S[ pʰ a] S[ l a]]  |
+--------------+---------+-----------------------------------------+
+-------------+---------+-------------------------------------+
|    Input    | Pattern |               Output                |
+-------------+---------+-------------------------------------+
| W[ S[ ʂ a]] |   AA    |         W[ S[ ʂ a] S[ ʂ a]]         |
| W[ S[ ʂ a]] |   AAA   |     W[ S[ ʂ a] S[ ʂ a] S[ ʂ a]]     |
| W[ S[ ʂ a]] |  AAAA   | W[ S[ ʂ a] S[ ʂ a] S[ ʂ a] S[ ʂ a]] |
| W[ S[ ʂ a]] |   AB    |         W[ S[ ʂ a] S[ l a]]         |
| W[ S[ ʂ a]] |   BA    |         W[ S[ s ɹ] S[ ʂ a]]         |
| W[ S[ ʂ a]] |   ABB   |     W[ S[ ʂ a] S[ l a] S[ l a]]     |
| W[ S[ ʂ a]] |   BBA   |     W[ S[ s ɹ] S[ s ɹ] S[ ʂ a]]     |
| W[ S[ ʂ a]] |  ABAB   | W[ S[ ʂ a] S[ l a] S[ ʂ a] S[ l a]] |
| W[ S[ ʂ a]] |  BABA   | W[ S[ s ɹ] S[ ʂ a] S[ s ɹ] S[ ʂ a]] |
| W[ S[ ʂ a]] |  AABB   | W[ S[ ʂ a] S[ ʂ a] S[ l a] S[ l a]] |
| W[ S[ ʂ a]] |  BBAA   | W[ S[ s ɹ] S[ s ɹ] S[ ʂ a] S[ ʂ a]] |
| W[ S[ ʂ a]] |  CDAB   | W[ S[ s ɹ] S[ l i] S[ ʂ a] S[ l a]] |
| W[ S[ ʂ a]] |  BDAC   | W[ S[ s ɹ] S[ l i] S[ ʂ a] S[ l a]] |
| W[ S[ ʂ a]] |  ABAC   | W[ S[ ʂ a] S[ l i] S[ ʂ a] S[ l a]] |
```

Figure 8. Results of Transformation with Monosyllabic Input



```
+------------------------+---------+-----------------------------------------+
|         Input          | Pattern |                 Output                  |
+------------------------+---------+-----------------------------------------+
| W[ S[ tsʰ ɹ] S[ l a]]  |  ABAB   | W[ S[ tsʰ ɹ] S[ l a] S[ tsʰ ɹ] S[ l a]] |
| W[ S[ tsʰ ɹ] S[ l a]]  |  CDAB   |                                         |
| W[ S[ tsʰ ɹ] S[ l a]]  |  AABB   | W[ S[ tsʰ ɹ] S[ tsʰ ɹ] S[ l a] S[ l a]] |
+------------------------+---------+-----------------------------------------+
+-----------------------+---------+---------------------------------------+
|         Input         | Pattern |                Output                 |
+-----------------------+---------+---------------------------------------+
| W[ S[ p a] S[ tɕ i]]  |  ABAB   | W[ S[ p a] S[ tɕ i] S[ p a] S[ tɕ i]] |
| W[ S[ p a] S[ tɕ i]]  |  CDAB   |                                       |
+-----------------------+---------+---------------------------------------+
+------------------------+---------+-----------------------------------------+
|         Input          | Pattern |                 Output                  |
+------------------------+---------+-----------------------------------------+
| W[ S[ k ua] S[ tɕ i]]  |  ABAB   | W[ S[ k ua] S[ tɕ i] S[ k ua] S[ tɕ i]] |
| W[ S[ k ua] S[ tɕ i]]  |  CDAB   |                                         |
+------------------------+---------+-----------------------------------------+
+-------------------------+---------+--------------------------------------------+
|          Input          | Pattern |                   Output                   |
+-------------------------+---------+--------------------------------------------+
| W[ S[ kʰ a] S[ tʂʰ a]]  |  ABAB   | W[ S[ kʰ a] S[ tʂʰ a] S[ kʰ a] S[ tʂʰ a]]  |
| W[ S[ kʰ a] S[ tʂʰ a]]  |  CDAB   | W[ S[ tɕʰ i] S[ tsʰ ɹ] S[ kʰ a] S[ tʂʰ a]] |
+-------------------------+---------+--------------------------------------------+
```

Figure 9. Results of Transformation with Disyllabic Input

6 Conclusion

This project designs a generator to transform monosyllabic and disyllabic single-morpheme onomatopoeia into all the well-formed polysyllabic words by reduplication. After analyzing the derivational relationship between patterns and constructing precondition hierarchy for trisyllabic and quadrisyllabic patterns, top-down and bottom-up finite-state tree transducers are applied to process two kinds of segmental changes, which are those dependent on word structure and variations influenced by adjacent segments. Prepositional reduplication is ultimately realized by composition of two kinds of transducers.

As the yield of the output trees is what we are concerned about, string transducer, instead of bottom-up tree transducer, can be applied to the yield after reduplication to make segmental change of coarticulation. Moreover, two-way string transducer could also be used for reduplication to save the complexity of composition of tree transducers.

References

Shao Jingmin. (1981). 拟声词初探. 语言教学与研究, (4), 57-66.

Shi Yuzhi. (1995). 论汉语的大音节结构. 中国语文, (3), 230-240.

Wang Hongjun. (1996). 汉语语音词的韵律类型. 中国语文, (3), 167-171.

Zhu Dexi. (1982). 潮阳话和北京话重叠式象声词的构造——为第十五届国际汉藏语言学会议而作. 方言, (3), 174-180.