Automatic Identifying Rhythm of Arabic Poem

Mohamed Y. Dahab
Computer Science Department, Faculty of Computing & Information Technology
King Abdulaziz University, Jeddah, Saudi Arabia

Ablah AlAmri
Jeddah Community College JCC,
King Abdulaziz University, Jeddah, Saudi Arabia

Bayan Bagasi
Computer Science Department, Faculty of Computing & Information Technology
King Abdulaziz University, Jeddah, Saudi Arabia

Ebtesam AlMalki
Computer Science Department, Faculty of Computing & Information Technology
King Abdulaziz University, Jeddah, Saudi Arabia

Ola AlBeshri
Computer Science Department, Faculty of Computing & Information Technology
King Abdulaziz University, Jeddah, Saudi Arabia

ABSTRACT
The poem in the Arabic language is a composition of verse to describe the sense of people to effect and share the feeling with listeners, the poem written in separate lines, that usually has figurative language and has a repeated rhythm and rhyme. Al-Khaleel Bin Ahmad Al-Frahide is the inventor of 'AlOrood' (العروض) poem template science (PTS). He invented weighting of Arabic poem which identifies poem. The Arabic poem contains one or more poem segments (PS) (البيت). The PS is a correct construction lyrics group, weighted according to the knowledge of the rules and RS, be in the same unit matching by a specific 'TafEelAh' (تفعيلة) musical pattern (MP). The Arabic PS classified according to how it matches with (بحر) poem template (PT).

This paper provides set of algorithms to identify Arabic poem template of Arabic poem segments that rely on poem template science.

General Terms
Algorithms and Natural Language Processing.

Keywords
Text Mining and Arabic Morphological Analyzer.

1. BACKGROUND
he poems in the Arabic language is one of the most popular literature, old Arabian poets gave a great attention to poems, the poem is a verse of poetry or a collection of correction structure words. The poem consists of many lines where each line has two halves clearly distinct, each line called PS.

Al-Khaleel Bin Ahmad Al-Frahide (**الخليلبنأحمدالفراهيدي**) invented PTS to measure and define a set of specific music templates. Arabic philologists define PTS as the science of the rules by means of which one distinguishes correct meters from faulty ones in ancient poetry. Al-Frahide determined the part of templates in which they can be increased or decrease without taking the tune out of balance. These musical templates are classified into variant sets of types, each of which is PT.

In order to understand the way how a PT is assigned to a poem, there is a need to clarify some terms and definitions:

MP is a repeated musical pattern which organizes in one PS. Diacritic/HarKah (حركة) is short vowel marks by Fatah (ا)(الفتح) represents a short /a/, Dammaا)(لضمة)) represent a short /u/ or Ksraahِ)(الكسرة)) represents a short /i/ Motaharek(متحرك) is the letter has HarKah as diacritic SaKen(ساكن) which indicates that the consonant to which it is attached is not followed by a vowel (the letter has diacritic by SoKOon (اْ) (سكون)).

TaqtEah(تقطيعة) Splitting is the process of mapping diacritic (movements and consonants) uttered to the equivalent character (/) movement or (٥) consonant.

To identify Rhythm of Arabic Poem need to analyze PS to MP. It depends on diacritics of each letter. These letters may be movement (متحرك) or consonant (ساكن). Also, every MP has a Splitting. To make a character in Splitting, keep spoken word without written word. Then if a letter dialect is 'HarKah' use (/) , and if 'SoKOon' use (٥). Sometimes the MP is changed by decreasing one or more letter this called ZeHAF (1] (زحاف].

2. INTRODUCTION
In old Arabic, Literature poems are the most popular, the poem in the Arabic language built to describe the sense of people to effect and share the feeling with listeners. One tool of effect poem is the way that the poetry put the words together, the way the words look and sound together which give a reader feel that the poem is a music template. PTS is invented to determine if the poems are balanced or not [1, 2].

The number of rhythms is 16 by agreeing on all scientists. These rhythms are shown in the table (2). The manual way to identify PT (rhythm) are writing the spoken for PS, then putting symbol instead each character. If character dialect is 'HarKah' put (/) and if 'SoKOon' put (**٥**), then choosing the match MP for each Splitting show in the table (1), then determine rhythm from a collection of MPs.

By used Artificial Intelligent techniques in text recognition, this paper develops a computational model to define the exact PT of a poem. The rest of the paper is organized as follow, present some related work in section 3. Then, section 4

present the stages of proposed algorithm. In Section 5 provide the tracing of the proposed algorithms. Section 6, provide enhancement of the proposed algorithms. Finally, conclusion and future workare provided in section 7.

Table 1.MP and its Splitting

MP	Splitting
'FaOoOLoN'	٥/٥//
'FaAEeLoN'	٥//٥/
'MaFaAEeELoN'	٥/٥/٥//
'FaAEeLaAToN'	٥/٥//٥/
'MoSTaFEeLoN'	٥//٥/٥/
'MoFaAAaLaToN'	٥///٥//
'MoTaFaAEeLoN'	٥//٥///
'MaFOoOLaATo'	/٥/٥/٥/

3. RELATED WORK

This section presents some previous works or programs done on this area. There is no paper had been written about this topic. But there are two programs worked to identify rhythm. First one is ALAroud Program [3]. It is determined 'Al Wzen' and PT(rhythm) for any PS that enter by the user and enter the diacritics for each word manually. It provides only PT that invented by Al-Khaleel Bin Ahmad Al-Frahide. The second one is Malik Alsheir program [4]. It provides the same function of the previous program but it's huge than the first one.

Table 2.Rhythm of Poems

Ite.	MP	Rhythm
4	'FaOoOLoN''MaFaAEeELoN'	'AtweeL'
4	'FaAEeLaAToN''FaAEeLoN'	'AlMaDed'
4	'MoSTaFEeLoN' 'FaAEeLoN'	'AlBasEeT'
6	'MoFaAAaLaToN'	'AlwaAfer'
6	'MoTaFaAEeLoN'	'AlKaAMeL'
4	'MaFaAEeELoN'	'AlHaZJ'
6	'MoSTaFEeLoN'	'ARaJz'
6	'FaAEeLaAToN'	'ARaMeL'
2	'MoSTaFEeLoN' 'MoSTaFEeLoN' 'MaFOoOLaATo'	'ASareEa'
2	'MoSTaFEeLoN' 'MaFOoOLaATo''MoSTaFEeLoN'	'AlMunSaRh'
2	'FaAEeLaAToN''MoSTaFEeLoN' 'FaAEeLaAToN'	'AlkhFeEf'
2	'MaFaAEeELoN'''FaAEeLaAToN'	'AlMudaAreA'
2	'MaFOoOLaATo''MoSTaFEeLoN'	'AlMuQTadeB'
2	'MoSTaFEeLoN' ''FaAEeLaAToN'	'AlMuJTath'
8	'FaOoOLoN'	'AlMuTqaAReb'

4. THE PROPOSED ALGORITHMS

In this section, present steps of proposed works to Automatic identify rhythm of Arabic Poem. These steps are:

1. Convert PS to (1 and 0).
2. Convert the segment of (0/1) to MP.
3. Identify type of rhythm.

4.1 Convert PS to (1 and 0)

In order to convert each PS to 1 and 0, first split PS to Individual character then save these character to an array A, then looping over array A to determine the sequence of 0 and 1 by the following cases:

Case 1: if the first character is the letter followed by diacritics ('Fathah ', 'Damma' or 'Kasraah') then the looping counter incremented by 2 and put 1 in the sequence array.

Case 2: if the character is the letter followed by a letter also then the looping counter incremented by 1 and put 0 in the sequence array.

Case 3: if the character is the letter followed by a diacritic ('SoKOon') then the looping counter incremented by 2 and put 0 in the sequence array.

Case 4 : if the character is the letter followed by a diacritic ('Shaddah') then check the third character , if its diacritics ('Fathah 'or 'Damma' or 'Kasraah') then the looping counter incremented by 3 and if it is a letter then the looping counter incremented by 2 and . Anyway, put 10 in the sequence array for each case.

Case 5: if the character is the letter followed by diacritic 'TanweenFatha' then the loop counter incremented by 3 and put 01 in the sequence array.

Case 6 : if the character is 'ALEF' followed by 'LAM' then check the third character , if its letter then-then the looping counter incremented by 3 and put 10 in the sequence array, else if the third character is ('Fathah 'or 'Damma' or 'Kasraah') then do case 2 .

After finish, check if the last element in sequence array is 1, add 0 into sequence array.

After analysis first step, draw the flowchart that shown in figure (1). In the flowchart, after entering PS, split the string into the list of characters. Every Sequence array muststart by 1, because the first word in any PS is start by 'Harkah' ('Fathah' or 'Damma' or 'Kasraah') which is represented by 1. Then check the following two characters or three characters if the case gets 0 or 1. The flowchart shows all possible cases.

4.2 Convert the segment of (0/1) to MP

The first step in this stage has split the sequence of 0 and 1 to segments (7 or 5 subsequences of complete sequence). By using FSA that shown in figure (2) which produce MP and table (3) shows that MPs with its final state can convert the segment to MP. However, There is a problem tosplit the sequence by the best way such as the segment does not match any known MP but the proposed algorithm will try to get the nearest similar MP. The following algorithm proposed way to get segment and find MP

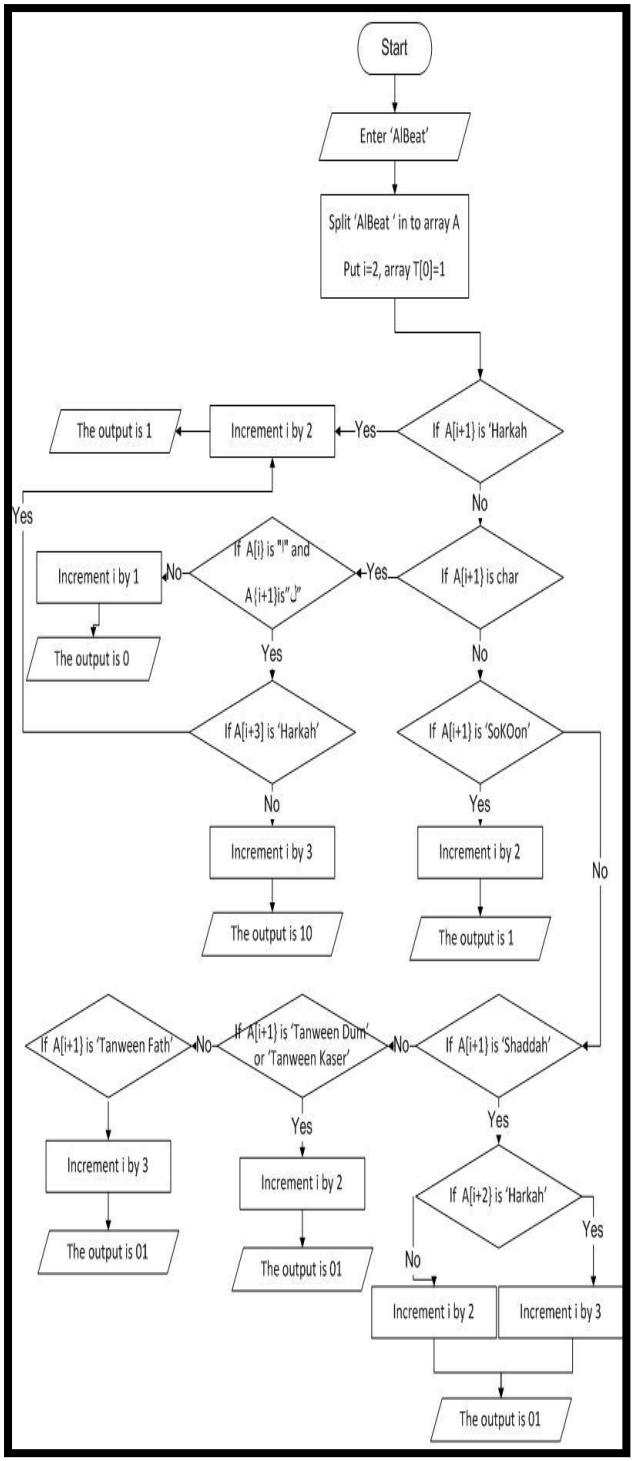

Fig. 1. Flowchart for the process of Step1

The Algorthim:

ST is segment of 0 and 1(7 or 5 subsequence) from S which is complete sequence that get from the previous stage.

Q = [q0 - q27], states of FSA

F is final state which has shown in table (3).

Use T(qi, STt) as transaction function, i is the number of state, t is index of ST and STt has a single value (0 or 1) .

Always start with segment has 7 subsequence if it doesn't match decrease to 5 subsequence if it's also it doesn't match decrease to 4 subsequence and get the nearest final state.

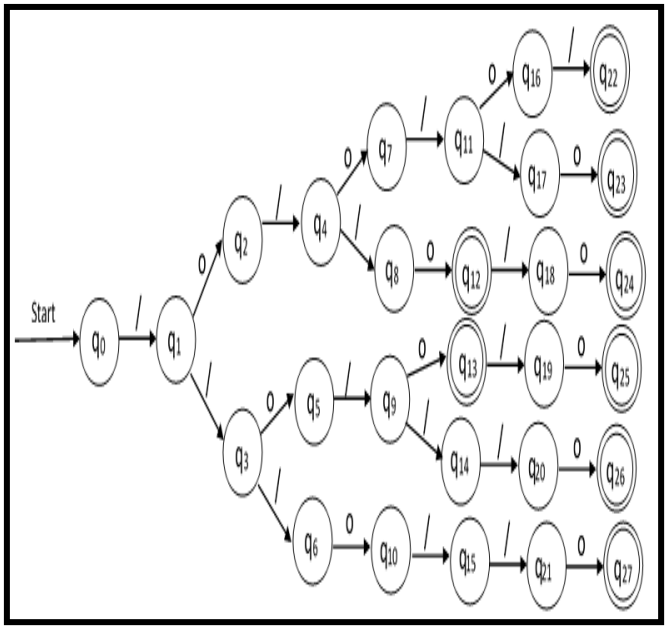

Fig. 2. Finite State Automata (FSA) that produce MP.

Table 3mp And Splitting And Final State On The (Fsa)

MP	Splitting	Final State
'FaOoOLoN'	٥/٥//	q13
'FaAEeLoN'	٥//٥/	q12
'MaFaAEeELoN'	٥/٥/٥//	q25
'FaAEeLaAToN'	٥/٥//٥/	q24
'MoSTaFEeLoN'	٥//٥/٥/	q23
'MoFaAAaLaToN'	٥///٥//	q26
'MoTaFaAEeLoN'	٥//٥///	q27
'MaFOoOLaATo'	/٥/٥/٥/	q22

Y is the index of subsequence stop there.

P is the state that function stop there.

(Y, P)= Convert_Segment(ST, Q, F) :

t←0

z←0

P ← Qz

while (STt≠ ∅)

T (Qz, STt)→Qz

ifQz ∈ F and STt+1 = ∅

P←Qz

Y ← t

return Y, P

else if Qz= ∅

Y ← t

return Y, P

else

P ← Qz

t ←t+1

Y ← t-1

return Y, P

At first time, If Y is 6 and P is final state that's mean the segment is matching with certain MP then start the next subsequence from Y+1. Or if Y is less than 6 and P is not final state then decrease the number of subsequence to 5 and send to function. Then, if Y is 4 and P is final state then that's mean segment is matching with certain MP then start the next subsequence from the Y+1. Or if Y is less than 4 and P is not a final state then decrease the number of subsequence to 4, choose the nearest final state from P to match with this subsequence and start the next subsequence from the Y+1.

After finished all subsequence of 0 and 1, the output is the group MP then identify PT using these MPs in the next stage.

4.3 IdentifyingRhythm's Type

Rhythms are 15 types. Each type has a different group of MP shown in the table (2). To identify the rhythm,the Finite State Automata (FSA) is used. Figure (3) shows the Finite State Automata (FSA) that produces rhythm and table (4) shows rhythm and its group of MP and final state that achieved by building on the (FSA). In the following, the proposed algorithm will present and its flowchart will show in Figure (4).

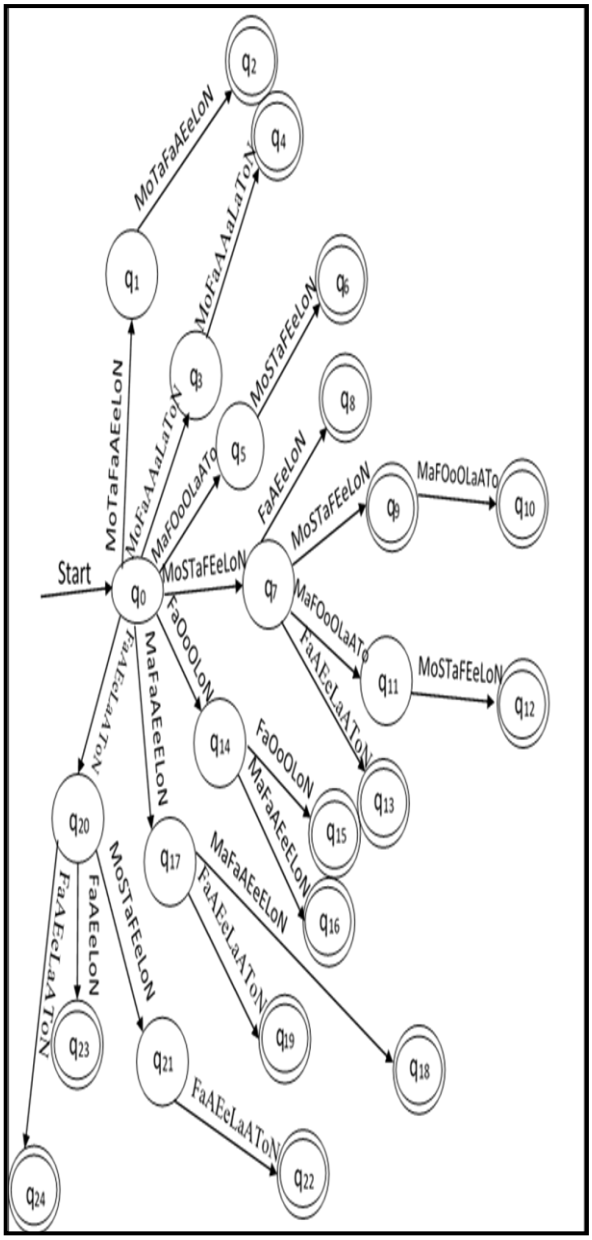

Fig. 3. Finite State Automata (FSA) that produce rhythm.

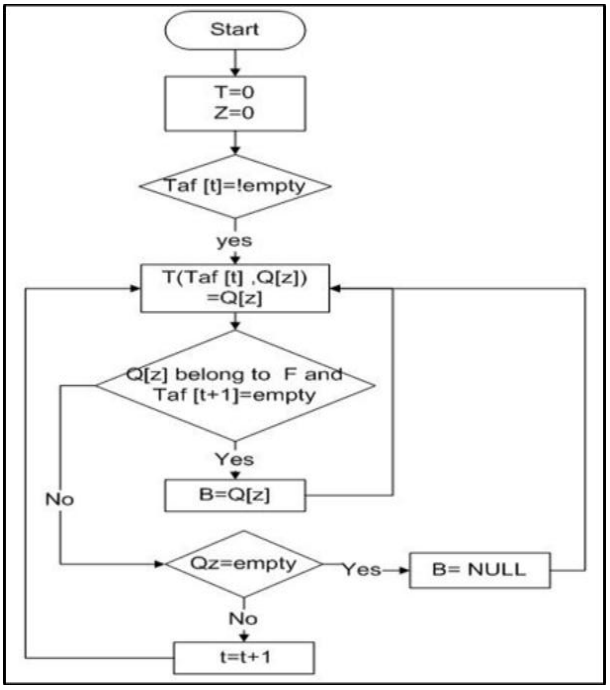

Fig. 4. Flowchart of previous algorithm that identify type of rhythm

The Algorithm:

Taf is the group of MP from the previous stage

Q = [q0 - q23], state of FSA

F is set of final states which shown in table (4)

T(Qz, Taft) as transaction function, z is the number of state, t is index of Taf and Taft has a value cetrin MP .

B is the final state which represents one type of rhythm

B=Identify_Rhythm(Taf,Q,F) :

t ←0

z ←0

while (Taft ≠∅)

T (Qz, Taft)→ Qz

if (Qz∈F) and (Taft+1 = ∅)

returnQz

else if (Qz = ∅)

return Null

else

t ←t+1

Table 4 Rhythm And It's Mps And Final State On The (Fsa)

Ite.	MP	Rhythm	Final State
4	‘FaOoOLoN’ ‘MaFaAEeELoN’	‘AtweeL’	q16
4	‘FaAEeLaAToN’ ‘FaAEeLoN’	‘AlMaDed’	q23
4	‘MoSTaFEeLoN’ ‘FaAEeLoN’	‘AlBasEeT’	q8
2	‘MoFaAAaLaToN’	‘AlwaAfer’	q4
6	‘MoTaFaAEeLoN’	‘AlKaAMeL’	q2
6	‘MaFaAEeELoN’	‘AlHaZJ’	q18
6	‘MoSTaFEeLoN’	‘ARaJz’	q9
6	‘FaAEeLaAToN’	‘ARaMeL’	q24
2	‘MoSTaFEeLoN’ ‘MoSTaFEeLoN’ ‘MaFOoOLaATo’	‘ASareEa’	q10
2	‘MoSTaFEeLoN’ ‘MaFOoOLaATo’‘ MoSTaFEeLoN’	‘AlMunSaRh’	q12
2	‘FaAEeLaAToN’‘ MoSTaFEeLoN’ ‘FaAEeLaAToN’	‘AlkhFeEf’	q22
2	‘MaFaAEeELoN’‘ ‘FaAEeLaAToN’	‘AlMudaAreA’	q19
4	‘MaFOoOLaATo’‘ MoSTaFEeLoN’	‘AlMuQTadeB’	q6
4	‘MoSTaFEeLoN’ ‘‘FaAEeLaAToN’	‘AlMuJTath’	q13
8	‘FaOoOLoN’	‘AlMuTqaAReb’	q15

5. TRACING THE PROPOSED ALGORITHMS

Steps of tracing algorithms that discussed previously:

Test case 1:

Will apply the previous algorithms on the following PS:

اذا طمعٌ يحل بقلب عبدٍ علته مهانة وعلاه وهن

Step1: Convert previous PS to (0) or (1). The sequence that's will be produced by using the flowchart in the Fig (1) and only convert the first verse of PS

اذا طمعٌ يحل بقلب عبدٍ

01110110110110111011

Step2: depending on the developed algorithm always take the first seven sequences of course from right to left.

011101101110110111011

0111011 ---> the trace on the graph leads to final state q26 as shown in figure (5). From table (3),q26 corresponding to "MoFaEeLaToN".

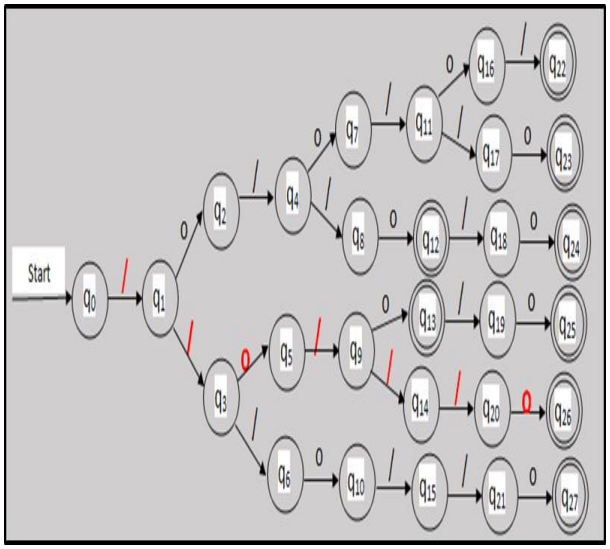

Fig. 5. Trace of algorithm on first seven sequences of Test case 1

The next, will complete and take the next seven sequences:

0111011<u>0111011</u>0111011

0111011 ---> the trace on the graph lead to final state q26. From table (3), q26 corresponding to "MoFaEeLaToN".

Then take the next seven sequences also until finish the first part of PS:

<u>0111011</u>01110110111011

0111011 ---> the trace on the graph leads also to final state q26. From table (3), q26 corresponding to "MoFaEeLaToN".

Step3: now go to algorithm that finds Rhythm to make trace over Fig (3). Finite State Automata (FSA) that produce rhythm of the following sequence of MP:

"MoFaEeLaToN" "MoFaEeLaToN" "MoFaEeLaToN"

From figure (6), the final state that produced from trace over FSA is q4 and in table (4) seeing that's q4 is corresponding to PT 'AlwaAfer' which is the correct PT.

Test case 2:

Will apply the previous algorithms on the following PS:

السامع الذم شريكٌ له ومطعم المأكول كالأكل

Step1: Convert previous PS to (0) or (1). The sequence that's will be produced by using thealgorithm in the Fig (1) and only convert the first verse of PS

السامع الذم شريكٌ له

11011010111010110101

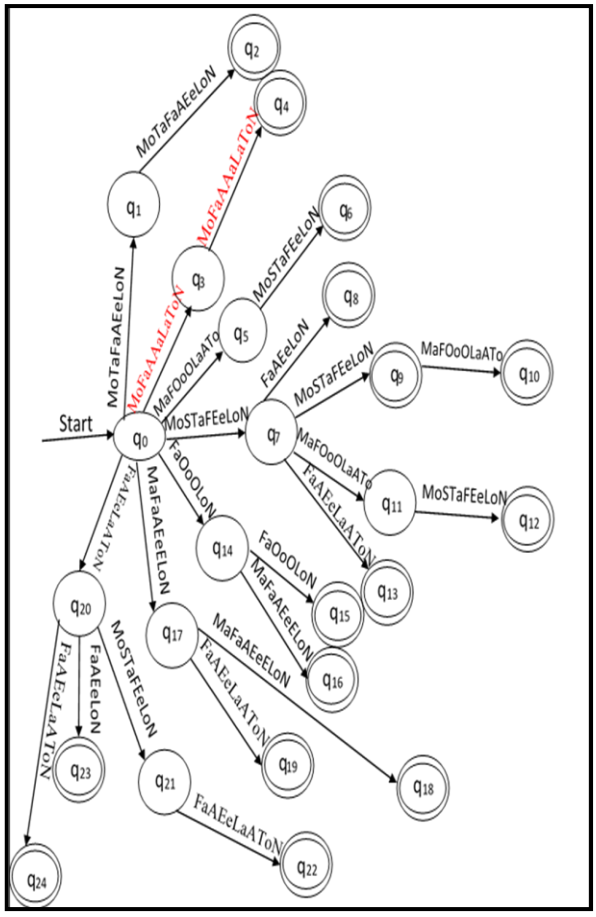

Fig. 6. Trace over FSA of Fig. 3 Test case 1.

Step2: depending on proposed algorithm always take the first seven sequences of course from right to left.

11011010111010<u>0110101</u>

0110101---> the trace on the graph lead to final state q23 as shown in figure (7). From table (3), q23 corresponding to "MoSTaFEeLoN".

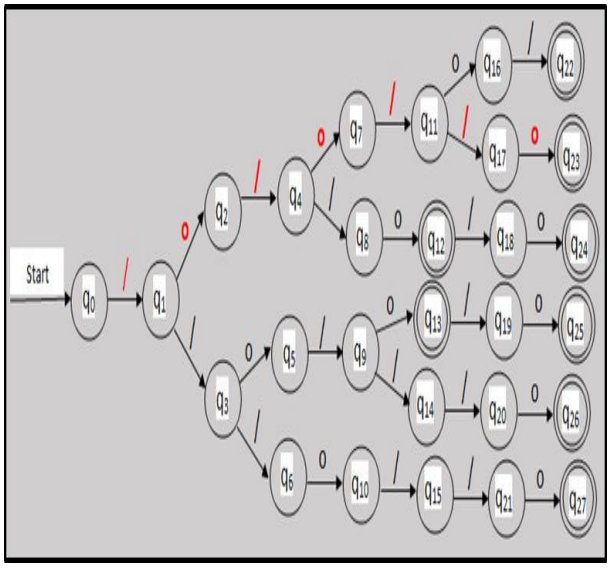

Fig. 7. Trace of algorithm on first seven sequences of test case 2

The next, will complete and take the next seven sequences:

1101101011101010110101

1011101 ---> no final state even when reduce the number of sequences. This situation produced a type of Zehaf called AltAEe shown in Appendix A by details.

6. ENHANCING THE PROPOSED ALGORITHMS

All MP presented the previous sections not necessary appear in its original shape, but it can be changed. This change can be decreasing one or more letter or change HaRaKaH to SoKoOn this type of change called ZeHAF (الزحاف). Or can be decreasing or increasing one or more letter this type of change called EaLah (العلة).

ZeHAF has two types in Arabic poet the first one is single ZeHAF while the second one is compound ZeHAF.

In the single the change only in one place in the MP like removing one letter from the MP. The compound ZeHAF the change can be in two places in the MP.

APPEDEX A describes all ZeHAF conditions and show tables of ZeHaF and EaLah in MP for each Rhythm.

6.1 Solving Zehaf Problem

To solve the problem of 'Zehaf,' analysis all cases of zehaf for each rhythm and arrange all possible changing on origin MP for each Rhythm then develop the previous algorithms which worked with correct MP without any Zehaf.

The main concept of development algorithm is that the algorithm will start by taking the sequence of zeros and ones which produced from the first step (Convert PS to machine language (1 or 0)), then the algorithm will trace over the FSA in figure 2 to reach the final state. The trace over FSA will depend on the first seven sequence of zeros and ones .The first case in the algorithm, if the trace over FSA in figure 2 reach final state then this state will help the FSA in figure (3) to find the accurate path that's will give the correct PT. The second case, if the trace over FSA in figure (2) does not reach final state then the next nearest final state will be chosen to be used in finding the path of the correct PT. This case have extra stage which is using the tables of 'zehaf' (Table 5 to Table 19 in Appendix B) in order to find the correct path depending on the original 'Tafeallah' if the correct path was not found in the FSA in figure (3) then the final state which has been used to find the path should be changed via in order traversal through the FSA in figure (2).

The algorithm used both finite state machine (figure (2), figure (3)) and tables of 'Zehaf' for each Rhythm (Table 5 to Table 19). The Algorithm also contains main functions start from Taf_First_subSequance() function in figure 8 then find_Rhythm() function in figure 10. In addition, there are two sub-function first one is find_Taf1() which called in Taf_First_subSequance() and second one is find_Taf2() which called in find_Rhythm().

The following are four figures of main functions and sub-functions with explained details for each function.

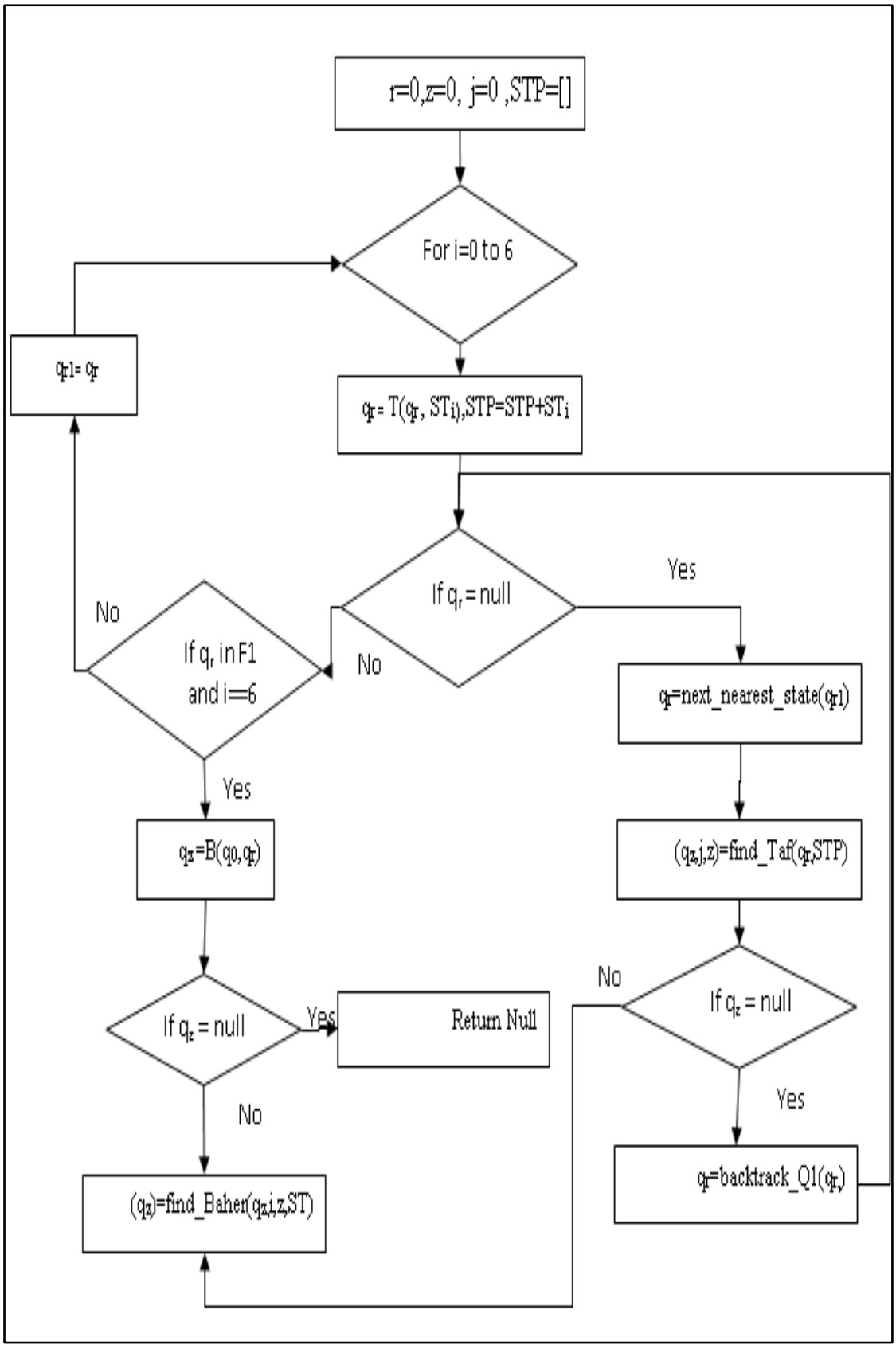

Fig. 8.qz= Taf_First_subSequance()

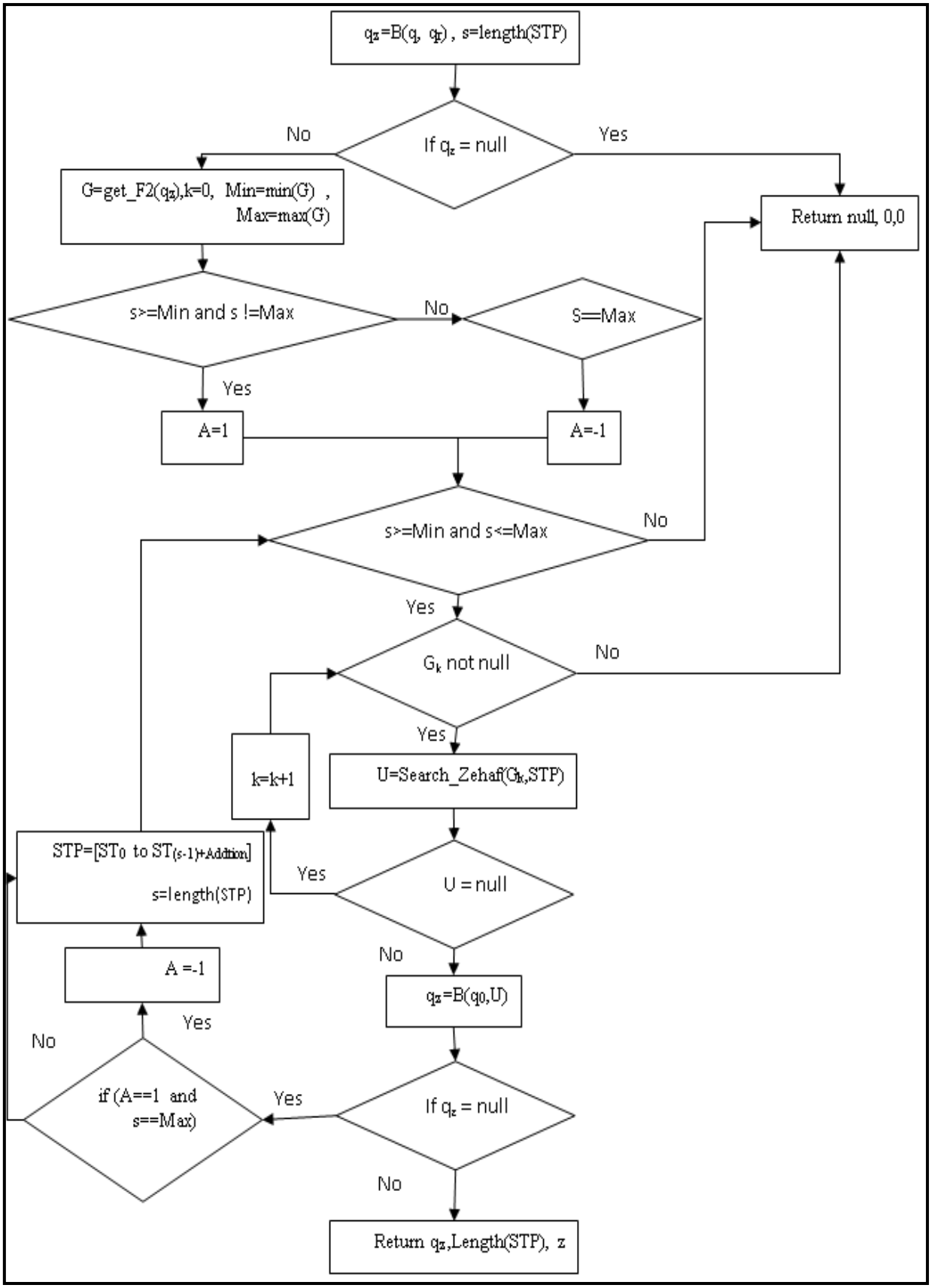

Fig. 9. (qz,j,z)=find_Taf1(qr,STP)

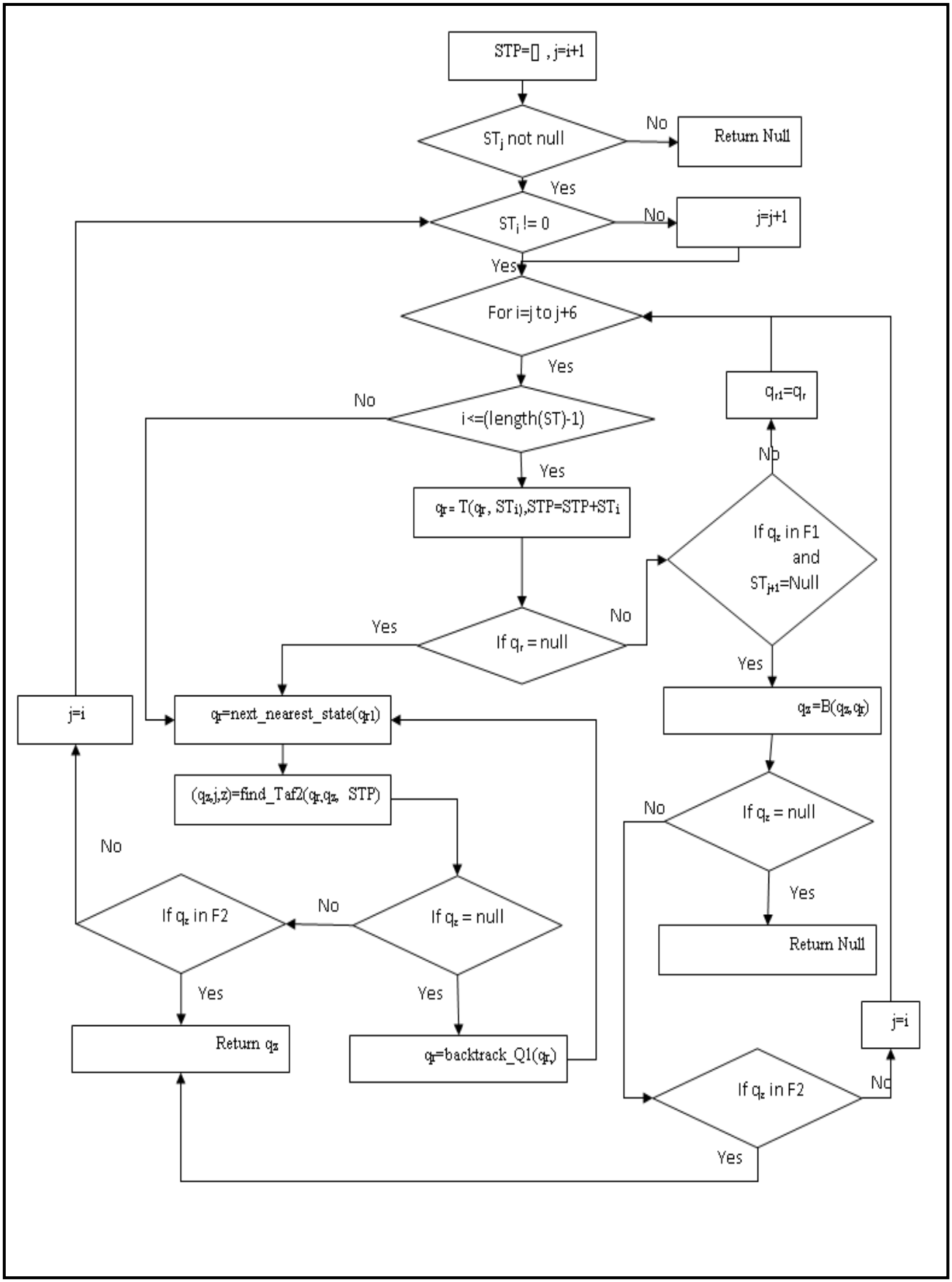

Fig. 10.qz=find_Rhythm(qz,i,z,ST)

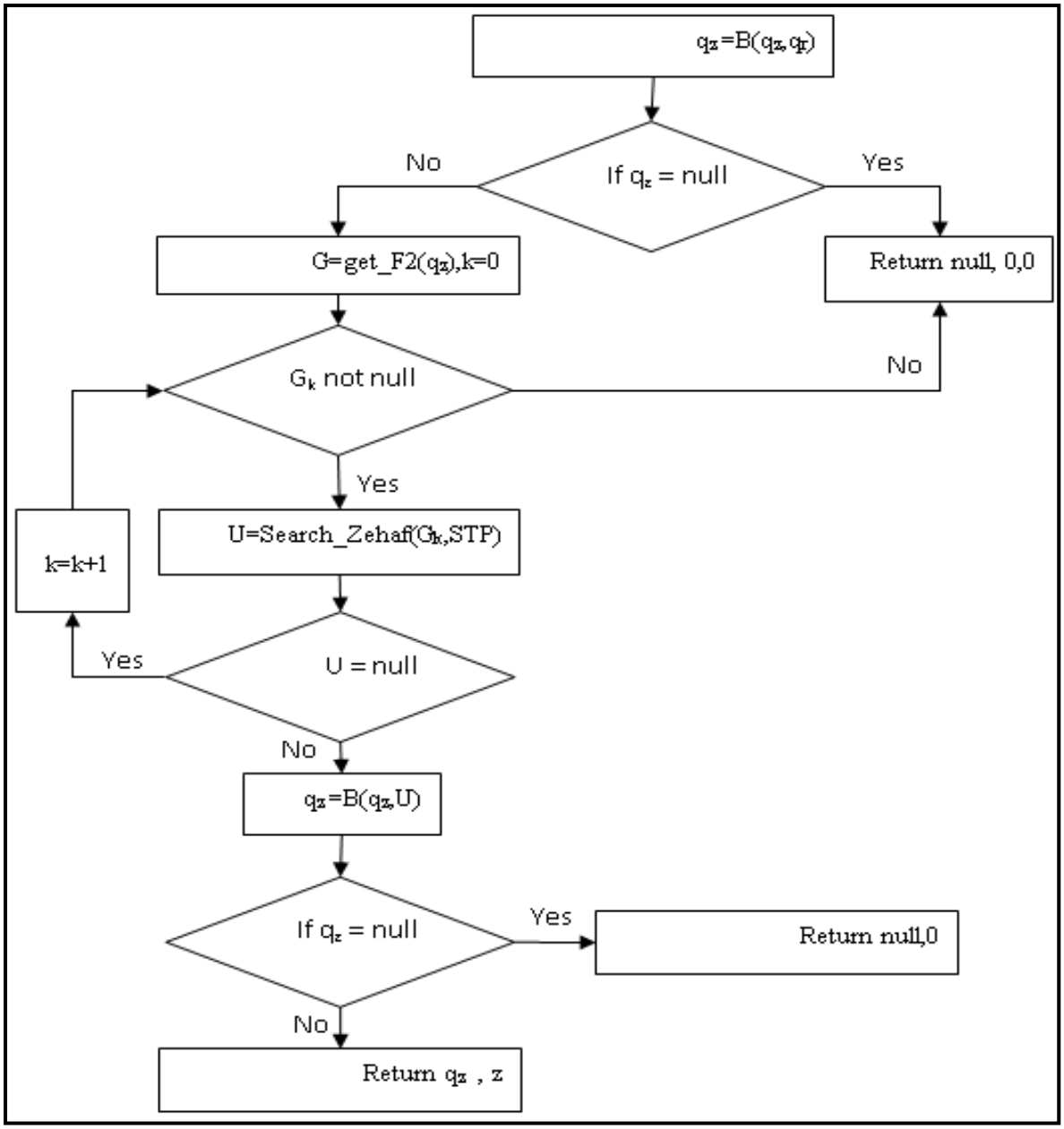

Fig. 11. (qz,j,z)=find_Taf2(qr, qz, STP)

From figure 8, the algorithm starts from the first seven sequences of zero and one. It used the finite state machine of in figure 2 to find the MP. If the state is dead state, then take the next near final state and used to test if these sub sequence is a 'Zehaf' of any rhythm by used find_Taf1() function shown in figure 9 and explain it later. If the result of function is null , then used backtrack_Q1() function which is used inorder traversal on finite state machine on figure 2 from the stop state to find the finial state if the visiting state is final state return it and repeat the step. Otherwise, if the state is finalTheThestate then used this state in the finite state machine in figure 3, if the result is null then the 'Wazen' is incorrect. Otherwise, if it's not null, save the number of states and the number of sequence to send to the find_Rhythm() to determine rhythm.

From figure 9, this function called from Taf_First_subSequance(). Also, it used the finite state machine of find rhythm in figure 3 and the result is a state. If it is a null then return null. Else if it's not null then found the following final states from this state that represent a specific group of rhythms. Then used these final states to search of 'Zehaf' of these rhythms by matching the subsequence with each Splitting on 'Zehaf' table of the rhythm and return the origin MP of matching Splitting. Then used these MP as a state to find the path in finite state of rhythm in figure 3 to reach to correct rhythm. Otherwise, if it's null, return null.

From figure 10, this function is the last step ofthe algorithm. It used to find the rhythm from the next subsequence from the previous function stop. It starts with next seven subsequence and used the same steps of the Taf_First_subSequance() function except used the find_Taf2() function in figure 11 which explain later. This function used to follow the path of the finite state machine of 'Rhythm '. Finally, if the state from find_Taf2() or finite state machine of rhythm is final state machine then this state represented the correct rhythm of this PS.

From figure 11, it used the finite state machine of find rhythm in figure 3 and the result is a state if it is a null then return null or if it's not null then found the following final states from this state that's final state represent the type of rhythm. Then used these final states to search of 'Zehaf' of these rhythms and return the origin MP. Then used these MP as a state to find the rhythm in figure 3. Otherwise, if it's null, decrease or increase the subsequence of zero and one and repeat the step.

6.2 Additional Enhancement

This part can be add it to algorithm to administrate the problem of the confliction of define rhythm 'AlTweel' or 'AlMuTqaAReb' and define rhythm 'AlMudaAreA' or 'AlHaZJ' as they have the start of sequence 0 and 1 :

STP2=STi to STi+6

U=Search_Zehaf(Gk,STP)

If U = Null ⟶ Return U

Else qz = B(qz,U)

If qz = Null ⟶ Return Null

Else ⟶ Return qz

The idea of algorithm is to check the next sequence on path if the next 'TafEalAh' of next sequence completed the path of FSA or not. If it's not in a path then repeat the step of the algorithm to find another path.

6.3 Tracing The Enhancement Algorithm

Will apply the development algorithm in test case 2 from previous algorithm trace which had 'Zehaf'.

السامع الذم شريكٌ له

1101101011101**0110101**

First, use the algorithm in Fig 8 Depending on proposed algorithm always take the first seven sequences of course from right to left.

0110101---> the trace on the graph lead to final state q23 as shown in figure (7). From table (3), q23 corresponding to "MoSTaFEeLoN".

Second, enter "MoSTaFEeLoN" into FSA of identifying PT by transaction function to checked if there is PT start with this. The result is q7 from FSA in figure3.

The next, will complete and take the next seven sequences:

110110**1011101**0110101

1011101 ---> no final state when it stopped q8 which isn't final state then will take the nearest final state which is in this case q12 and the subsequence here will be (1101). Then used an algorithm in Figure 9. It will be searched about the matching of this sub sequence in the table of Zehaf of the final state in the same path of q7 which are q8,q9,q10,q12,qnd q14 from Figure 3. In this subsequence, it's not found the matching then increase the subsequence by one extra character from the original complete sequence which becomes (11101) then will search about it in the table of Zehaf again. Also, this sub sequence not found in all table of Zehaf.Then also the sub sequence will be increased with the extra character from the original sequence which becomes 011101. Finally,this sub sequence found in the table of Zahaf (Appendix A- Table 16) of q10 which is a path to Rhythm 'ASareEa'in.

Next, the division here is change because the Zehaf so the new Splitting will be as follow to complete the trace first seven sequence then sequence:

11011010111010110101

After that will take the next seven sequence which is 1101101 and also the trace over FSA in Fig (2) (produce MP') will be done then will take the next sequence which is 1101101 by trace over FSA in Fig 2 by trace reach dead end before q18 by algorithm will take the next nearest final state which is q24 which is 'FaAEeLaAToN' from Table. It's not matching with the next of path from q7 and q9 in second FSA of find Rhythm, here as the algorithm say will make backtracking and the previous final state in in-order traversal will be q12 which are 'FaAEeLoN'. Also, it's not matching with the next state of the path from q7 and q9 in second FSA of find Rhythm, then used in-order traversal in FSA and the result is q23 which is 'MoSTaFEeLoN'. It's not matching with the next of path state from q7 and q9 in second FSA of find Rhythm again. Then, take the previous finial state by in-order traversal, will be q 22 which is 'MaFOoOLaATo'. It's matching with the next state of path from q7 and q9 in second FSA of find Rhythm. Then search about the matching with sub sequence 01101. When to search the table of Zehaf in (Appendix A – table 16).

So the complete sequence of MP at the end of the trace will be:

MoSTaFEeLoN". 'MoSTaEeLoN' 'MaFOoLaA'

When to tracing this sequence of MP on FSA in Fig3 (produce rhythm) reach the final state q10. After searching about q10 in Table 4 found Rhythm ASareEa which is the correct rhythm as shown in figure (12).

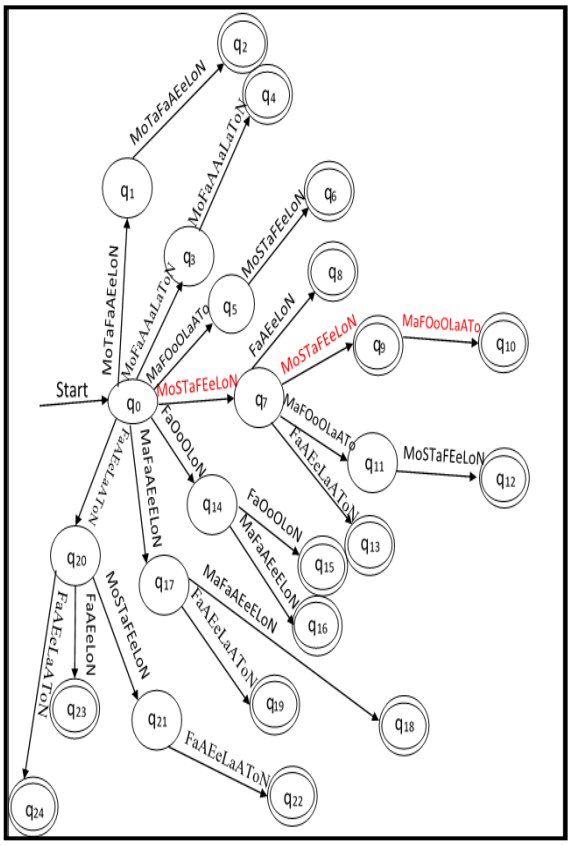

Fig. 12. Trace over FSA of Fig. 3 test case 2

7. CONCLUSION AND FUTURE WORK

The algorithm is testing using all possible cases of various rhythm. The accuracy percentage on correct balance PS is 100%. Where, the accuracy percentage on PS include Zehaf is 90%. Because, the correct balance of PS is follow clear path or sequence of MP but if PS has Zehaf may be it conflicted path where Zehaf has many similarity between different Rhythm.

Many readers find a pleasure in reading Literature specially poems, because the way that the poetry put the words together, the way the words look and sound together which give a reader feel that the poem is a music templates, in Arabic language PTS is used to determine if the poems are balance or not, and if it is balance specify the PT that the poem is belong to. The algorithm that describe in this paper, is used to identify rhythm of Arabic Poem that rely under Al-Frahide's 'Wzen'.

This algorithm divides in to three stages. First stage has been converting PS to sequence zero and one depending in the dicrieates. It was checked all possible cases and get the correct output if zero or one. Second stage has been extracted the 'TafEalah' from the sequence that's result from first stage. In this stage, using first finite state automate in figure (2) to specify 'TafEalah' by segment from sequence. The last stage, it has been identified rhythm from group of MP. Also, this stage used the second finite state automata in figure (3) to specify rhythm. There is a problem when poem has imperfection called 'Zehaf'. Then, improved the second stage and third stage to one stage which is used the both FSAs to detect the 'Zehaf' and used the origin MP to find rhythm. From improving algorithm, it can be identify rhythm where ever the 'Zehaf' in PS.

To achieve high accuracy, test on more number of PS from different sources of poem which include various types of Zehaf.

To improve this work, add the all other commoner rhythm for slang poem. These rhythms are over the 50 type. Then, it must need more analysis and studying to form the FSA of these rhythms. Also, it will add the function extract the fault in 'Wazen' of poem and help the user to correct that.

8. ACKNOWLEDGMENTS

We would like to Thanks to Ms. NahedGzali – graduated from Tanta University in faculty of Arts Arabic department - whose experience was very helpful for us in the understating and analysis the all cases of poem and PTS.

9. REFERENCES

[1] Encyclopedia of AlArood. (elibrary4arab) Retrieved 10 15, 2014, from electronic libraryvforarab: http://www.elibrary4arab.com/ebooks/arabic/arood-qafeya/index.htm

[2] Alison Booth, K. J. (2010). The Norton introduction to literature, tenth edition.W.W.Norton. alwasel, s. A. (n.d.).

[3] AlHussain, A. (2009, 5 5). ALAroud Program website. Retrieved 11 3, 2014, from ALAroud Program website: http://azahou45.free.fr/

[4] AlQaied, S. (2014, 2). Malik Alsheir. Retrieved 11 1, 2014, from Malik Alsheir : http://www.knsite.com/

[5] Ghanim, M. H. (2013, 2 11). Eng. Mohammad blog. Retrieved 11 15, 2014, from blogspot: http://mhmdhmdy.blogspot.com/2013/02/blog-post_1516.html

[6] Saleh, M. S. (n.d.). Two way to alArood tag. Retrieved 11 25, 2014, from Alfaraheedy: http://www.faraheedy.com/lessons/lesson2

[7] Dictionary. (2014). dictionary. Retrieved 12 2, 2014, from http://dictionary.reference.com/browse/poem

[8] Dictionary. (2014). dictionary. Retrieved 12 2, 2014, from http://www.merriam-webster.com/dictionary/poem

[9] WEIL, Gotthold& MEREDITH-OWENS, G. (1960) "Arud.

10. APPENDIX (A)

ZeHAF is change of MP by decreasing one or more letter. This appendix describes all ZeHAF.

1. AlAssb (العصب)

Making the fifth letter of the MP as SAkEn here is single ZeHAF because the change only in one place.

Example:

MoFaAaLaToN :MoFaAaLToN

2. AL3AQL(العقل)

Removing the fifth letter of the MP if it is not SAkEn here is single ZeHAF because the change only in one place.

Example:

MoFaAaLaToN :MoFaAaToN

This type of ZeHAF consider as bad type of ZeHAF.

3. ALNaQSS (النقص)

It is the process of making the fifth letter of the MP as SAKen and removing the seventh letter.

Example:

MoFaAaLaToN: MoFaAaLTo

4. ALADB (العضب)

Just removing the first letter

Example:

MoFaAaLaToN: FaAaLaToN

5. ALAQSS (العقص)

Example:

MoFaAaLTo: FaAaLTo

6. ALQaSM(القصم)

Removing the first letter from MP that's have the fifth letter as SAKeN

Example:

MoFaAaLToN: FaAaLToN

7. ALGaMaM(الجمم)

Also this type is compound ZeHAF because we applying the rule on the result we have get it from applying AL3AQL

Example:

MoFaAaLaToN :MoFaAaToN: FaAaToN

8. ALQaTF (القطف)

Making the fifth letter SAKeN

Example:

MoFaAaLaToN :MoFaAaLToN

9. ALKaBN(الخبن)

Removing the second SAKeN letter

Example:

FaEEeLoN :FaEeLoN

MoSTaFEeLoN :MoTaFEeLoN

FaEEeLaAToN: FaEeLaAToN

MoSTaFEeLoN: MoTaFEeLoN

10. ALTaEE (الطي)

Removing the fourth SAKeN letter

Example:

MoSTaFEeLoN: MoSTaEeLoN

MaFOoLaTo: MaFOLaTo

11. ALKabl(الخبل)

Removing the second and forth SAKeN letter

Example:

MoSTaFEeLoN: MoTaEeLoN

This kind is bad and not usable.

12. ALWaKF(الوقف)

Making the seventh letter SAKen

Example:

MaFOoLaTo: MaFOoLa

13. ALKaSHF(الكشف)

Removing the seventh letter if it is MoTaHReK

Example:

MaFOoLaTO: MaFOLaT

14. ALKaFF(الكف)

Removing the seventh SAKeN letter

Example:

FaEEeLaAToN: FaEEeLaATo

MoSTaFEeLoN: MosTaFEeLo

MFaAEeLoN: MFaAEeLo

FaAEeLaAToN: FaAEeLaATo

15. ALSHaKel (الشكل)

Removing the second and seventh SAKeN letters

Example:

FaEEeLaAToN: FaEeLaATo

MoSTaFEeLoN: MoTaFEeLo

16. AlGabd (القبض)

Removing the fifth SAKEN letter

Example:

MFaAEeLoN: MFaAELoN

FaOoLoN: FaOoLo

17. AlKharm (الخرم)

Remove first letter (watad) in first MP

Example:

In FaOoLoN :

If the MP is in the original shape (FaOoLoN) then this called THALAMA (ثَلْمًا)

FaOoLoN: OoLoN

If the MP is after applying AlGabd in the original shape (FaOoLo)this is called THARMA (ثَرْمًا)

Example:

FaOoLo: OoLo

In MaFaAEeELoN:

MaFaAEeELoN: FaAEeELoN

18. ALKarb (الخرب)

Removing M letter from MaFaAEeELoN after applying ALKaFF

Example :

MFaAEeLo :FaAEeELo

19. AlShater (الشتر)

Removing M letter from MaFaAEeELoN after applying AlGabd

Example:

MFaAEeLoN: FaAEeELoN

20. AlEdmar (الإضْمار)

Making the second letter SAKen

Example:

MoTaFaAEeLoN: MoTFaAEeLoN

21. AlWags(الوقص)

Removing the second Motaharek letter

MoTaFaAEeLoN: MoFaAEeLoN

22. ALKazl(الخزل)

Making the second letter SAKen and removing the seventh SAKeN letters

MoTaFaAEeLoN: MoTFaAEeLo

23. AlCut (القطع)

Remove last letter and making the previous letter SAKen

FaAEeLoN: FaAEeL

11. APPENDIX (B)

Table 5 ALZEHAF AND ALEALAH FOR Baher 'Arajz'

MP	Splitting
'MoSTaFEeLoN' (origin)	٥//٥/٥/
'MoTaFEeLoN'	٥//٥//
'MoSTaEeLoN'	٥///٥/
'MoTaEeLoN'	٥////

Table 6 ALZEHAF AND ALEALAH FOR Baher 'AlMuTqaAReb'

MP	Splitting
'FaOoOLoN' (origin)	٥/٥//
'FaOoO'	٥//
'FaOoOLo'	/٥//
'OoOLoN'	٥/٥/
'OoOLo'	/٥/

Table 7 ALZEHAF AND ALEALAH FOR Baher 'AlHaZJ'

MP	Splitting
'MaFaAEeELoN' (origin)	٥/٥/٥//
'MaFaAEeLoN'	٥//٥//
'MaFaAEeELo'	/٥/٥//
'FaAEeELoN'	٥/٥/٥/
'FaAEeELo'	/٥/٥/
'FaAEeLoN'	٥//٥/
MaFaAEeE	٥/٥//

Table 8 ALZEHAF AND ALEALAH FOR Baher 'AlKaAMeL'

MP	Splitting
'MoTaFaAEeLoN' (origin)	٥//٥///
'MoTFaAEeLoN'	٥//٥/٥/
'MoFaAEeLoN'	٥//٥//
'MoTFaEeLoN'	٥///٥/
'MoTaFaA'	٥///

Table 9 ALZEHAF AND ALEALAH FOR Baher 'ARaMeL'

MP	Splitting
'FaAEeLaAToN' (origin)	٥/٥//٥/
'FaEeLaAToN'	٥/٥///
'FaAEeLaATo'	/٥//٥/
'FaEeLaATo'	/٥///
'FaAEeLaA'	٥//٥/
'FaEeLaA'	٥///

Table 10 ALZEHAF AND ALEALAH FOR Baher 'AlMuQTadeB'

MP	Splitting
MaFOoOLaATo(origin)	/٥/٥/٥/
'MaOoOLaATo'	/٥/٥//
'MaFOoLaATo'	/٥//٥/
'MoSTaEeLoN'	٥///٥/

Table 11 ALZEHAF AND ALEALAH FOR Baher 'AtweeL'

MP	Splitting
'MaFaAEeELoN' (origin)	٥/٥/٥//
'MaFaAEeELo'	/٥/٥//
'MaFaAEeLoN'	٥//٥//
FaOoOLoN (origin)	٥/٥//
'FaOoOLo'	/٥//
'OoOLoN'	٥/٥/
'OoOLo'	/٥/

Table 12 ALZEHAF AND ALEALAH FOR Baher 'AlMaDed'

MP	Splitting
'FaAEeLaAToN' (origin)	٥/٥//٥/
'FaEeLaAToN'	٥/٥///
'FaAEeLaA'	٥//٥/
'FaEeLaA'	٥///
'FaAEeLaATo'	/٥//٥/
'FaEeLaATo'	/٥///

Table 13 ALZEHAF AND ALEALAH FOR Baher 'AlBasEeT'

MP	Splitting
'MoSTaFEeLoN' (origin)	٥//٥/٥/
'MoTaFEeLoN'	٥//٥//
'FaAEeLoN'	٥//٥/
'FaEeLoN'	٥///
'MoSTaEeLoN'	٥///٥/
'MoTaEeLoN'	٥////
'MoSTaFEeL'	٥/٥/٥/
'MoTaFEeL '	٥/٥//

Table 14 ALZEHAF AND ALEALAH FOR Baher 'AlMuJTath'

MP	Splitting
'MoSTaFEeLoN' (origin)	٥//٥/٥/
'MoTaFEeLoN'	٥//٥//
'MoSTaFEeLo'	//٥/٥/
'MoTaFEeLo'	//٥//
'FaAEeLaAToN' (origin)	٥/٥//٥/
'FaEeLaAToN'	٥/٥///
'FaAEeLaATo'	/٥//٥/
'FaEeLaATo'	/٥///
'FaALaAToN'	٥/٥/٥/

Table 15 ALZEHAF AND ALEALAH FOR Baher 'AlwaAfer'

MP	Splitting
'MoFaAAaLaToN' (origin)	٥///٥//
'MoFaAAaToN'	٥//٥//
'MoFaAAaL'	٥/٥//
'MoFaAAaLToN'	٥/٥/٥//
'MoFaAAa'	/٥//
'MoFaAAaLTo'	/٥/٥//
'FaAAaLaToN'	٥///٥/
'FaAAaLToN'	٥/٥/٥/
'FaAAaLTo'	/٥/٥/
'FaAAaToN'	٥//٥/

Table 16 ALZEHAF AND ALEALAH FOR Baher 'ASareEa'

MP	Splitting
MaFOoOLaATo (origin)	/٥/٥/٥/
'MaFOoOLaAT'	هه/٥/٥/
'MaOoOLaAT'	هه/٥//
'MaFOoLaA'	٥//٥/
'MaOoLaA'	٥///
'MoSTaFEeLoN' (origin)	٥//٥/٥/
'MoTaFEeLoN'	٥//٥//
'MoSTaEeLoN'	٥///٥/
'MoTaEeLoN'	٥////

Table 17 ALZEHAF AND ALEALAH FOR Baher 'AlMunSaRh'

MP	Splitting
'MaFOoOLaATo'	/٥/٥/٥/
'MaFOoOLaAT'	هه/٥/٥/
'MaOoOLaAT'	هه/٥//
'MaOoLaATo'	/٥///
'MaFOoLaATo'	/٥//٥/
'MaOoOLaATo'	/٥/٥//
'MaFOoOLaA'	٥/٥/٥/
'MaOoOLaA'	٥/٥//
'MoSTaFEeLoN'	٥//٥/٥/
'MoTaFEeLoN'	٥//٥//
'MoSTaEeLoN'	٥///٥/
'MoTaEeLoN'	٥////

Table 18 ALZEHAF AND ALEALAH FOR Baher 'AlkhFeEf'

MP	Splitting
'FaAEeLaAToN' (origin)	٥/٥//٥/
'FaEeLaAToN'	٥/٥///
'FaAEeLaATo'	/٥//٥/
'FaEeLaATo'	/٥///
'FaAEeLaA'	٥//٥/
'MoSTaFEeLoN' (origin)	٥//٥/٥/
'MoTaFEeLoN'	٥//٥//
'MoTaFEeLo'	//٥//
'MoSTaFEeLo'	//٥/٥/

Table 19 ALZEHAF AND ALEALAH FOR Baher 'AlMudaAreA'

MP	Splitting
'FaAEeLaAToN' (origin)	٥/٥//٥/
'FaAEeLaATo'	/٥//٥/
'MaFaAEeELoN' (origin)	٥/٥/٥//
'MaFaAEeELo'	/٥/٥//
'MaFaAEeLoN'	٥//٥//
'FaAEeELo'	/٥/٥/
'FaAEeLoN'	٥//٥/