Search Applications Based Data Analysis Method Boyer Moore

A Mulyana[1], I Purwanto[1]
[1.] Information System, Bandar Lampung University, Indonesia

1. Introduction

1.1 Background

Can not be denied the technology has evolved very fast today. Almost all modern human use of technology to facilitate the work and life. Each technology has a storage type that is different for each of the data, but the text remains the main form of data storage (Purwoko, 2006).
Problems *string searching* the data store is to find some characters*(pattern)*in a large amount of text (Hartayo Vembrina, & Meilana, 2011). It can be concluded that the search *string* or *string searching* is the search for a *pattern* of text. Often operating in the text search process involves the emergence of a *string,* and location of the text in question (Sulistyo, 2006). Because it was the search *string* is one thing that is needed primarily in search of data. Boyer-Moore algorithm is a string searching algorithm, published by Robert S. Boyer and J. Strother Moore in 1977. Boyer-Moore algorithm is considered to have the best results in practice this algorithm, amatching algorithm moves *string* on directions right to left. Systematically, the steps undertaken Boyer-Moore algorithm when matching strings are:
1. Algorithm Boyer-Moore began to match the pattern at the beginning of the text.
2. From right to left, this algorithm would match the character-by-character pattern with the character in the corresponding text, until one of the following conditions are met:
1) The characters in the pattern and in the text compared to mismatch (mismatch).
2) All of the characters in pattern matching. Then the algorithm will inform the discovery in this position.
The algorithm then shift pattern to maximize the value of good-suffix shift and bad-character shift, and then repeat the steps until the pattern at the end of the text. This algorithm is then developed so as to create the Turbo Boyer-Moore algorithm, Tuned Boyer-Moore and algorithms Zhu-Takaoka. In this paper resume *reviewer* searchanalyzing the data (string matching) using the Turbo Boyer-Moore algorithm, Tuned Boyer-Moore and Algorithm Zhu-Takaoka on all three journals to know each generated data.

2. Basis Theory

2.1 Literature

Overview The libraries used by the authors in the study are as follows.

2.1.1 Application

According Nazrudin Safaat H (2012: 9) The application software is a subclass of computer software that uses computer skills directly to a task that the user wants.

2.1.2 Data

According to Turban (2010, P41), the data is base description of objects, events, activities and transactions are recorded, are grouped and stored but not yet organized to convey a certain meaning.

2.1.3 Algorithm Boyer-Moore

According to (Helmi, 2013) *Boyer-Moore algorithm* is a string searching algorithm published by Robert S. Boyer and J. Strother Moore in 1977. This algorithm is considered as the most efficient algorithms in common applications .Not like string searching algorithm discovered earlier, *the algorithm Boyer-Moore* began to match characters from the right pattern (pattern in the search). The idea behind this algorithm is that with the start of matching characters from the right, and not from the left, it will be more information is obtained.

2.2 Study Overview

The review studies used by the author is as follows.
a) A Suffix Tree Based Handwritten Chinese Address Recognition System
Ninth International Conference on Document Analysis and Recognition IEEE 0-7695-2822-8. Author Yan Jiang and Zheng Ren Siemens AG, Xiaoqing Ding Department of Electronic Engineering Tsinghua University, Beijing, China and Konstanz, Germany An introduction to the address automatically (optical character recognition (OCR) is so important to read the address written by hand, especially in oriental languages such as China and Japan. By using methods algorithm Zhu Takaoka.
b) Computational Complexity in Language String Processing and theoryof halting problem in Deterministic Turing Machine Accepting Context Sensitive Language, Context Free Language or Regular Language
2nd International Conference on Computing for Sustainable Global Development (IndiaCom 978-9-3805-4416-8 / 15 2015 IEEE author Chetan Sharma and Rinku from Chitkara University, Himachal Pradesh INDIA, form data using a searchTuring Machine **algorithm complexity**algorithm considers context-free grammar in Chomsky normal form (CNF) and find a list of variables for all substring from string and finally check the list of variables, for *parent string* with Turbo Boyer-Moore method.
c). Bond Graph Modeling and State Estimation of Stringing Machine
7th International Conference on Modeling, Identification and Control (ICMIC 2015) 978-0-9567157-5-3 © 2014 IEEE author Rim Attafi and Nadia Zanzouri *Université de Tunis El Manar / Tunis-Tunisia* Use BG models are graphical methods of modeling tools in varied fields (electrical, mechanical, hydraulic,) using the Boyer-Moore Tuned.

3. Analysis Of Results And Discussion

3.1 Analysis Results

In this study the authors analyzed the three journals with different cases include results and discussion using *the Boyer-moore*algorithm.Results and discussion journals have been the author of an analysis of the four journals as follows.
The result has been the author analyzes the following three fourth.
3.1.1 Journal A Suffix Tree Based Handwritten Chinese Address Recognition System This journal is *the International Journal of Engineering Research and General Science* (2014) pages 177-181, 2014. ISSN 2091-2730 © Journal written by Srikrishna S, Sreenivasulu Reddy, Vani S of *Department of Mechanical Engineering, Sri Venkaeswara University College of Engineering, Tirupati,*India. This journal has a case that is a recognition address automatically (optical character recognition (OCR) is so important to read the address written by hand, especially in oriental languages such as Chinese and Japanese writing style differs from person to person, many characters are very similar in shape, structure and writing. This causes difficulties in distinguishing character lines and text, Solutions for the introduction character based criteria, using algorithms Boyer-Moore namely Tzu Takaoka. Tthe calculation process starts from the stage of system process is divided into two main procedures (Figure 1), the first is an image processing, including elimination of noise, extracting zip code and text lines and segmentation lines of text. there is no recognition or address information are involved. Both are guided by the recognition and address data base.

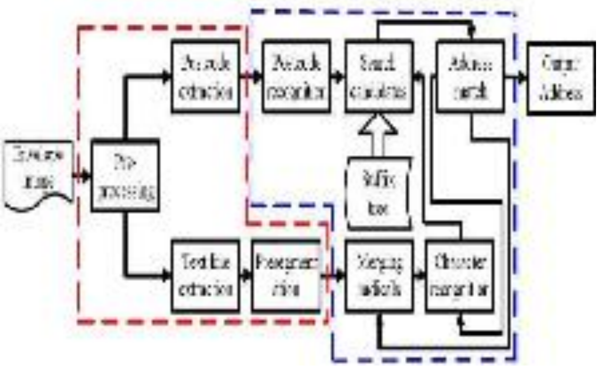


Figure 1 The Process Of Reading The Address

Figure 2 Achina Envelope

3.2 Handwritten Address Recognition

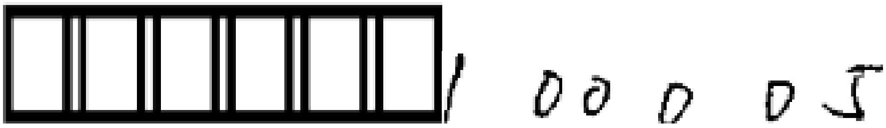
An Envelope typical Chinese handwritten.

3.2.1 Extraction Postal Code And The Recognition

Process is written the way in six separate boxes on postal codes using this template is applied to filter out redundant items in hypothetical address

a) Postcode region in Fig.2

b) Slant correction of (a)



(c) Template for the area code post

(d) Taken digit

Figure 3. TheText Line Taken FromFig.2

3.3 Text Extraction Line

Envelopelayout is classified into four types: noise, graphics, text and images. Just as the text label components are combined as a line of text. The text line pre-segmentation By using segmentss_1, s_2, ..., to show a combination of radical $s_{i,1}$ $s_{i,1}$ + 1. ..$s_{i,2}$, the task of character segmentation turned to find integers m and an array $\{a_{i,1}, i, 2\}_{1\le i\le m}$ to determine the number of characters which are combined and the corresponding radical way combination. give the correct segmentation and mark the beginning and end index radical for each respective character images.

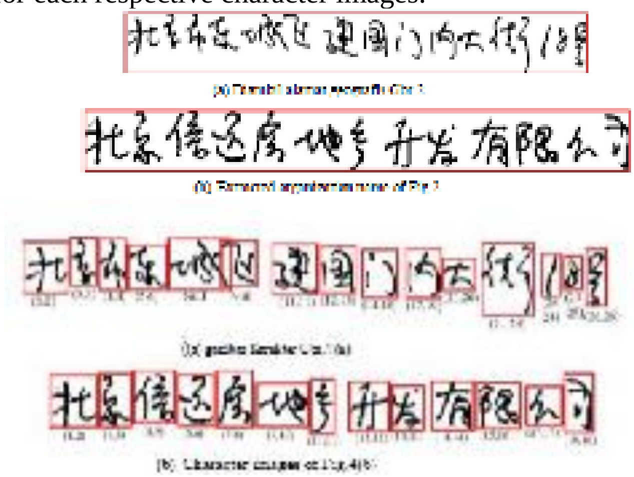
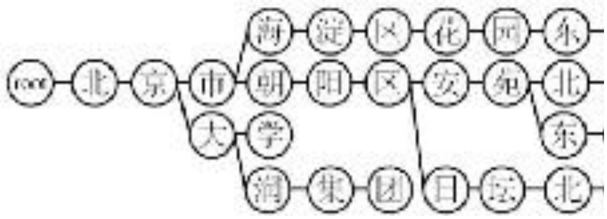


Figure 4 Character

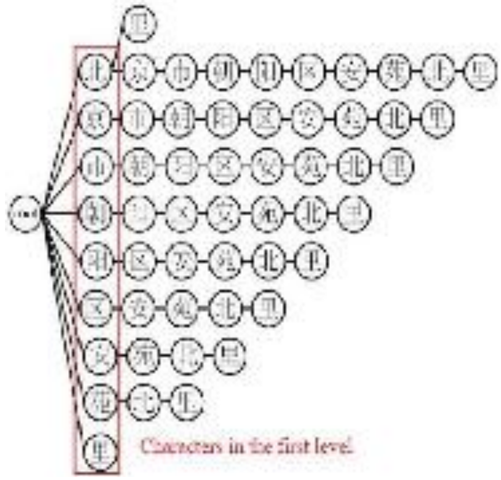
3.4 A Suffix Tree

Tree is a multi-way tree structure to a string. It is widely adopted to save a large dictionary of English. The idea is that all the strings share a common stem or prefix hang off a common node. ,*A Suffix Tree* allows us to find a string of sub-string common share and access any character in the address quickly, a *Tree* and *A Suffix Tree* illustrated. In the structure of the *Tree*,we save six address differences and one address is stored.



(a) The Structure Of Tree Illustration

Figure 5. Picture Illustration of the structure of the Tree



(b) An illustration of The Structure SuffixTree

Figure 6. An Illustration Of The Structure Suffixtrie

3.5 Proposed Algorithm Substrings Generation

Step 1: For all the possible combinations of radical Ip, q, $1 \le p \le q \le n$,provide an assessment of what is potentially a valid character.

Step 2: All valid combinations classified radical and M candidate Ck (p, q), $1 \le k \le M$ was given to *me*, q.

Step 3.1: For each recognized Ip, q, if there is another picture I recognize, s that satisfies $q + 1 = r$,check the candidate set of two pictures. For every Ck (ip, q), $1 \le k \le M$, we identify a character node in the first level of the tree suffix (leaf nodes directly Then we are looking for all the child nodes and check if there are any nodes that contain the characters in the candidate set Ir, s. If Cl (Ir, s) is found, we add a <2,p, s, Ck (p, q) Cl (Ir, s), * *pNode*>recording this 2 character from substring set S. node p Noting pointer to a child node containing a Cl (Ir, s).

Step 3.2: for <d, p, s, c1 c2 ... cd, * *pNode*>s,we check all recognized the combination of radical Is1, t, $s + 1 \le t \le n$.If there is a character cd1 in candidate SETIs1, t, which is also contained in a child node of pNode, we add(d + 1) -character substring <d + 1,p, t, c1 c2 ... cd *1* * *pNewNode*>to S,where pNewNode record pointers to child nodes that contain cD+ 1, Step 3.2 is repeated until there is no substring newly added.

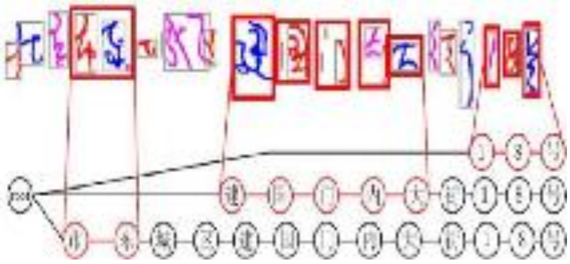


Figure 7 algorithm process

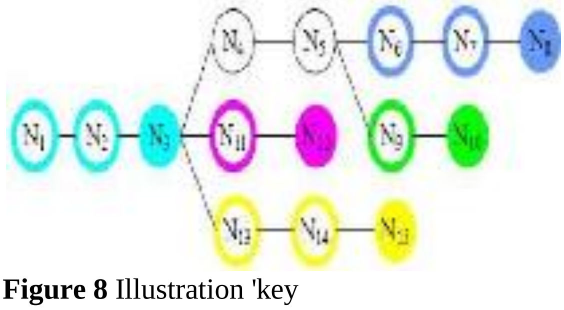


Figure 8 Illustration 'key code' Node

Candidates lookup address FFI x tree by examining all the records substring in S.If the number of characters in the string exceeds a pre-defined scan all nodes descent to prepare dates address-The results of recognition can also be considered in finding steps. If the address is added as a candidate, at least one 'major character' of the address must occur on the posterior radical candidate set combination. A 'key character' is defined in the paper show that the character is very important to distinguish one string to the other when the two strings share a long common substring. As figure colored nodes is the end of a particular address. Figure shows 5 addresses ending N3, N8, N10, N12 and N13. From the end of the address, tracing back all the nodes that antecedent to us.

Meet node crotch multi-address lines. All nodes are visited (including end nodes but without node groin) is characterized as the main character for an address. In Fig.9, N1, N2 and N3 node key characters to represent address at N3. Node is marked with the same color (including solid and hollow) is the 'key code' node. Node crotch in Fig.9 is N3 and N3. N4 and N5 is not 'the key characters' node because they use less-uniquely identify an address. To further reduce the candidates address, zip code referred to in this process.

3.6 Address MatchesFunction

The address comparison we compare the input with the radicals each candidate addresses to optimize the function C as shown below. As we discussed in 2.3, the goal is to find an array of 1)$a_{i,1} \leq a_{i,2}$, (2)$a_{i,2} < a_{j,1}$, for all $i<j$ and $a_{i,2} = 0, a_{j,1} = 0$

$C = 1 \leq i \leq N,\ a_{i,1} = 0\ P\ (C_i\ |I a_{i,1},\ a_{i,2})$ - $w1\ \#\{i: a_{i,1} = 0\}$

$-w2(a_{i,1}$ - max $a_{k,2}-1) +(n-$ max $a_{i,2})$

$a_{i,1} = 0k < i\ a_{i,2} = 0$ (1)

in the above equation, the address string c1 c2 ... cn, Trust character first item is the average (see for further details of the estimate p and the cost of matching represented in two of the last item: the middle shows the number of characters that can not be matched with a combination of radicals and the last one is air-num radicals who do not fit in with the character . The algorithm is illustrated, we treat each string sub as a unit, as we see, by improving the previous substring, the search space to optimize the above functions greatly reduced. Although it failed to recognize some of the characters, also can get results matched correctly because all the important characters are recognized correctly. Computational Complexity in Language String Processing and theoryof halting problem in Deterministic Turing Machine Accepting Context Sensitive Language, Context Free Language or Regular Language *2nd International Conference on Computing for Sustainable Global Development (IndiaCom* 978-9-3805-4416-8 / 15 2015 IEEE author Chetan Sharma and Rinku from Chitkara University, Himachal Pradesh INDIA In this journal have a case that is the process of determining Turing machine can be designed to receive these types of languages such as regular, free language context, the language of regular etc. Turing machine will stop if the string is invalid. this problem can be avoided by providing input as appropriate, indicates the acceptance of the string is valid. In this paper, a simple demonstration of checker brackets designed the Turing machine is deterministic introduced to understand the reception and the halting problem of Turing machine is deterministic in reception if the invalid or string invalid belongs ordinary, context free or context-sensitive language, using algorithms Turbo Boyer -Moore in this journal algorithm.

The Turing machine is a general model of computation, can potentially execute algorithms exist various modes such Turing machine accepts the language, computation functions, and carry out other types of calculations. Variants such as multi-tape Turing machine is briefly discussed. Non 16 deterministic Turing machine is introduced, as well as the universal Turing machine, which anticipates a stored-program computer machine. The machine will stop when the language or string context sensitive, context of language.

3.7 The Algorithm Complexity Cyk

Algorithm considers context-free grammar in Chomsky normal form (CNF) and find a list of variables for all substring from a string and finally check the list of variables, for the *parent*string.If the end of the variable list contains start symbol then existing membership to be declared null string. Consider the string w of length n and a context-free grammar G in CNF. We know that the decidability of the question "w EL (G)?" Can be determined exactly 2n - 1 steps, namely, membership w can be determined in the 2n-1 steps, As we know that the total number of substring of w is n (n-1) / 2, where n is the length of w. Now, as the complexity of T (w) = Total substring X number of steps per substring = nx 1 + (n-1) x 3 + (n-2) x 5 + ... + 1 x (2n-1), where n is the length w Therefore, T (w) = O (n3) cyk algorithm has better performance than the previous membership of the algorithm has exponential time complexity.

3.8 Turing Machine Computational Complexity

Computational complexity is also known as the complexity of the Time-Space. Computational Complexity of Turing Machine is based on two factors:
The amount of time it takes (Time complexity) and the amount of space required in storage or processing tasks (Space Complexity).

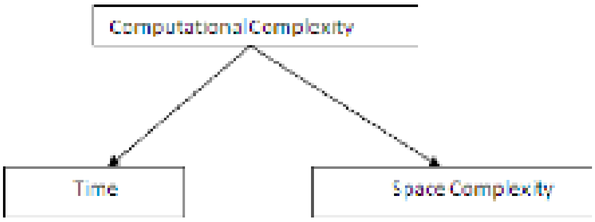


Figure 9 The Complexity Of Computing Turing Machine

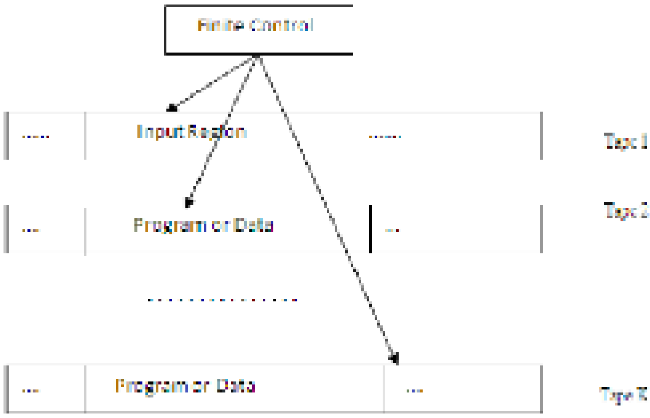
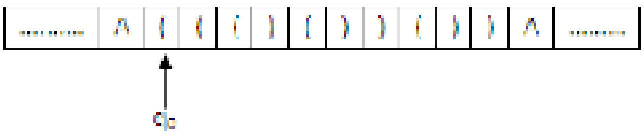


Figure 10 Figure KTM

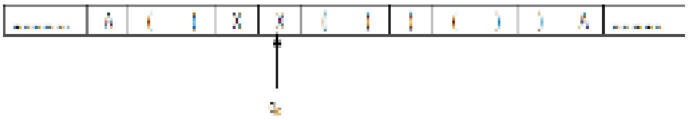
3.9 Time Complexity
KTM Consider(Turing machines) M shown in Figure 2.13 k KTM has limited two-way cassette, one cassette contains a cassette including the input and all input records can be written. The time complexity is defined as: "If for any input of length n (n> = 0), M makes most T (n) moves before stopping them M Said to T (n) time bounded Turing Machine (or the time complexity T (n)). Language is recognized or function is calculated by M is said to be the time complexity T (n).
3.9.1 Space Complexity Of Algorithm Complexity Cyk
algorithm cyk performing well. consider the string w of length n and a context-free grammar G in CNF. We know that decidability of the question "w EL (G)?" can be specified exactly 2n - 1 steps, namely, membership w can be determined in the 2n-1 steps, As we know that the total number of substring of w is n (n-1) / 2, where n is the length of w. Now, the time complexity T (w) = Total substring x number of steps per substring = nx 1 + (n-1) x 3 + (n-18 2) x 5 + ... + 1 x (2n-1), where n is the length of w.
3.9.2 Examples of Expression
Z (Z (Z (Z) (Z) Z) (Z) Z) Z
Tape Initially we were on the beginning p0 with the state q0 side as shown in the picture below or Tape.

Table 1 Mapping Tape Turing Machine



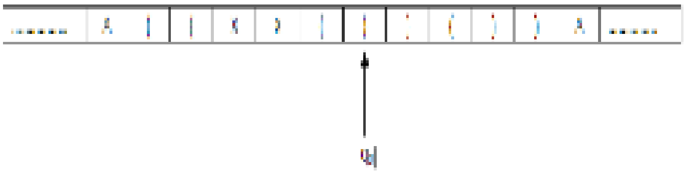
Procedure:
Mapping to tape over the Turing Machine
(q0, () = (q0, (R)
(q0, Z) = (q0, Z, R)
By applying the above mapping engines will be moved to the side left and state changes. When the state q1 met (bracket in q1 state symbol will be converted to X and change the state to q0 and move to the right. As in this suit) bracket we find (the bracket that shows one of the brackets balanced.

Table 2 Mapping Tape Turing Machine



(q0, x) = (q0, X, R)

Table 3. mapping tape Turing Machine



(q1, X) = (q1, X, L)
(q0, A) = (q2, A, L) (2)

Now according to the above mapping if we find a (kind of marker) in state q0 we will move to the state q2. This shows that there is no) bracket fixed and the head will move to the left side of the tape.

(q2, X) = (q2, X, L)
(q1, A) = (H, N) (3)

By applying the above mapping if we find A in the state q1 then the machine will stop and print or indicates that the find additional) cause to fail. This means unbalanced brackets.

Table 4 MappingTape Turing Machine

(q2, () = (H, N) (4)

By applying the above mapping if we find (in a state q2 then the machine will stop and print or indicates that it find an extra (bracket. Its unbalanced brackets.

For example, :

Table 5 Mapping Tape Turing Machine

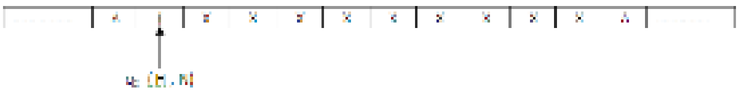


Table 6 Mapping Tape Turing Machine

(q2, A) = (H, Y) (5)

by applying the above mapping if we find A in the state q2 then the machine will stop and print or indicates that the brackets are given balanced.

3.10 Time Complexity

Turing machineabove check also string nested and organized to check the brackets within the solution the machine will start from the left side and moving toward the far right end of the string until he found the opposite symbol ")" corresponding to "(" . As first met ")" rewrite X in place and change the direction of movement of the head to read / scan the string. head moving machine and process symbols until she find ultimate in "(" and ")" among the symbols that have not crossed over the pair as "(" and ")" remained. If the remaining symbols then the machine will display the halting state. To find the time complexity of the engines driving the number of steps required when strings are also nesting is provided as input. Generally string (... (... () ...) ....) With a length n is fed as an input. The number of steps required are as follows: n / 2 + 1 + 2 + 3 +...... + (n-1) + n + n = n2 / 2 + 2n. In equation n / 2, 1,2,3, ..., n-1 are the steps required to process the symbol in accordance with the brackets left to right and the next two n is the number of steps to check every parenthesis left still not acceptable and give the state machine. The figure below shows the steps of head movement while also string have a nest and organized fed as input engine. On the other hand when a series of specific form () () ... . () of length n is provided as an input to the engine, the number of steps required to process that rope symbol is: 2 + 4 (n / 2-1) + 2 + n = 3n in the above equation 2 was the first to process the first left and right brackets and 4 (n / 2-1) is to (n / 2-1) another pair of left and right brackets each requires 4 steps to process. 2 Next to check that there are no fixed right parenthesis, and n is the end for which no remains left parenthesis. So the maximum number of steps that the machine uses the input from a string of length n is given by (n2 / 2) + 2n.

3.11 *Complexity Room*

Complexityspace of the string if the sequence of characters representing a balanced-parenthesis expressions

- Start with an empty stack.
- For every left parenthesis, push.
- For each right parenthesis, pop.
- For any non-parenthesis character, doing nothing.
- Expression balanced if the stack is empty and there are no more characters to process.
- It is not balanced if the well after the last character of the stack is not empty (too many left parenthesis), or if stackempty and right parenthesis encountered (right without left) For all standard operating the stack (push, pop), worst-case run-time complexity can be O (1). We can say and not is because it is always possible to implement stacks with underlying representation inefficient.

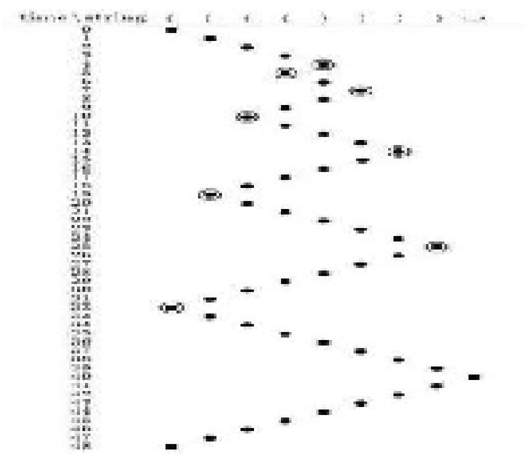


Figure 11 Complexity Room

Complexityspace for the brackets checker is O (n / 2) where n represents the length of the string. Analysis Method approach of using TurboAlgorithm *Boyer-Moore* This algorithm is a development of thealgorithm. *Boyer-Moore*In this algorithm is recording a segment of text that match a suffix to the pattern of the last made to match. The advantage of using this algorithm is possible to reach a 'leap' pass through the segments and can do a *turbo-shift*.In this journal using the *algorithm ofcyk* this algorithm known function As we know that the *algorithm cyk* consider context-free grammar in Chomsky normal form (CNF) and find a list of variables for all substring from a string and finally check the list of variables, for the parent string. Cyk algorithm has better performance than the previous membership of the algorithm has exponential time complexity. on calculating namely:

complexity T (w) = Σ Total substring X number of steps per substring =

nx 1 + (n-1) x 3 + (n-2) x 5 + ... (infinity) + 1 x (2n-1),

where n is the length w

Therefore, T (w) = O (n3) (6)

Given decidability of the question "w EL (G)?"

Can be determined exactly 2n - 1 steps, namely, membership w can be determined in the 2n-1 steps, As we know that the total number of substring of w is n (n-1) / 2, where n is the length of w.

3. Bond Graph Modeling and State Estimation of Stringing Machine
7th International Conference on Modeling, Identification and Control (ICMIC 2015) 978-0-9567157-5-3 © 2014 IEEE author Rim Attafi and Nadia Zanzouri *Université de Tunis El Manar / Tunis-Tunisia.* In this journal have a case that the number of components, we use state approach (not accessible in part by) Many application areas have been realized by the decline; control, diagnosis, synchronization and state estimation, for different types of systems as a single system, not the system is stationary, there is no discrete-time linear systems and linear systems with unknown inputs. By doing calculations with Tuned Boyer-moore algorithm in the journal based approach is considered in order to estimate BG string tension of a racquet stringing machine with Luenberger reduced. Indicators residual (output estimation error) is generated for error detection sensors.

Using the graphical method BG models namely the modeling tools in varied fields (electrical, mechanical, hydraulic,). To facilitate the building of models and exploit a reduced number of components. BG is based on the transfer of power between the different parts of the system. BG's first approach to the design of Luenberger observer. Steps to build Luenberger reduced so that observers use BG approach. Section III proposes observer diagnosis with BG. An illustrative example is developed in section IV and demonstrate the advantages of the method considered.

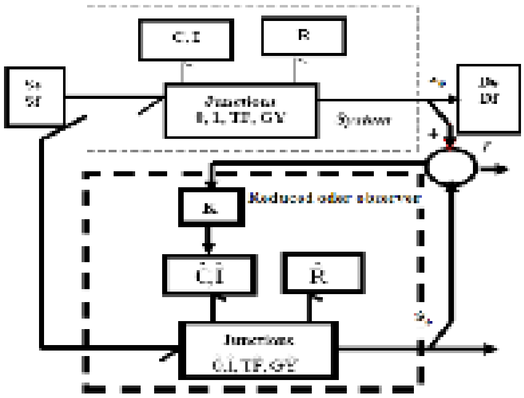
Results

a. Reduced State Estimation Observer for the full sequence was designed to estimate all state vector. While the reduced order observer is used to estimate the part of the state vector. The first step is to separate variables and variable allegedly suspected.

$$\begin{cases} \dot{x} = \begin{bmatrix} \dot{x}_a \\ \dot{x}_b \end{bmatrix} = \begin{bmatrix} A_{aa} & A_{ab} \\ A_{ba} & A_{bb} \end{bmatrix} x + \begin{bmatrix} B_a \\ B_b \end{bmatrix} u \\ y = \begin{bmatrix} C_a & C_b \end{bmatrix} \begin{bmatrix} x_a \\ x_b \end{bmatrix} \end{cases} \qquad (7)$$

b. Reducing Observer based approach to BG. BG becomes a powerful tool that is customized modeling for a firm diagnosis. BG is used for modeling, state estimation and diagnosis of dynamic systems. Becausality concepts used to solve the problem of state estimation. To build the framework of the observer is reduced, BG approach based approach Luenberger algebra. Be causality to avoid using the concept of calculus. This one, summarized by the following steps:
   - Step 1: Check the non existence of output exaggeration
   - Step 2: Verification of observability structural
   - Step 3: Selection, verification (reversibility), and calculation
   - Step 4: Suppression of a dynamic element that is associated with
   - Step 5: Sum term
   - Step 6: Sum of the term

c. Strong diagnosis by reduced order observer
observer structure Luenberger pursuant to modeling is presented BG



(i) Verify that the structural model of the system BG observed;
(ii) Development of observer (which is presented as part II.B) using BG;
(iii) Expression of symbolic of the remaining inferred from the following equation:
(iv) In the form of residual

3.11 Application on Assembling Machine

To develop legal control, monitoring or diagnosis, requires a complete knowledge of the state system. In addition to the partial knowledge To facilitate model building, and exploiting the reduced number of components, we use state approach (not accessible in part by) Many application areas have been realized by the decline; control, diagnosis, synchronization and state estimation, for different types of systems as a single system, not the system is stationary, there is no discrete-time linear systems and linear systems with unknown inputs. Bond Graph (BG) modeling and state estimation by Luenberger reduced so that the observer string tension of a racquet stringing machine. A significant advantage of this approach is that the observer design is achieved by using a graphical model of ***BG**-MooreTurned*Boyer. *Tuned Boyer-Moore algorithm* is the implementation of a simplified version of the Boyer-Moore. As the name implies Tuned This algorithm is widely used in the calculation of the search engine, the algorithm is very fast in practice. String matching algorithm is to check whether the character of the pattern accordingly. In the journal using the calculation BG approach that is based on an algebraic approach uses the concept be causalityLuenberger to avoid calculus. This one, summarized by the following steps:

- Step 1: Check the non existence of output exaggeration
- Step 2: Verification of observability structural
- Step 3: Selection, verification of (reversibility), and the calculation of Ca and Cb
- Step 4: The suppression element dynamic associated with
- Step 5: Sum term Ky
- Step 6: Sum ofterm

a fullas an observer, the calculation of profit from reduced order observer based on pole placement technique proposed by [17]. Indeed, we select reduces orders observer characteristic polynomial and matrix calculations are based on this polynomial coefficients, which are equal to the total gain of the family cycle sequence of causal model BG observer.

$$P_{(\bar{A}_{bb}-K\bar{A}_{ab})}(p) = p^n + \hat{a}_1 p^{n-1} + \hat{a}_2 p^{n-2} \dots + \hat{a}_{n-1} p + \hat{a}_n \quad (8)$$

Data search or S*tring matching* the search for a *pattern* in a text. *String matching* is used to find a *string* called a *pattern* in a *string,* called the text. The working principle ofalgorithm *string matching* is as follows:

1. Scan text with the help of a *window* with the same size length *pattern.*
2. Put the *window* at the beginning of the text.
3. Compare the characters in the *window* with the character of the *pattern.*After the match (whether the results match or do not match) is performed rightward shift in the *window.*This procedure is done repeatedly until the *window* is located at the end of the text. This mechanism is calledmechanism. *a sliding window*

3.12 How it Works String Matching

The obvious way to find a *pattern* that matches the text is to try to find at each starting position of the text and ignore the search as soon as possible if the wrong character is found (Knuth, DE *et*al.1977). The first process is to align the leftmost part of a *pattern* with text. Then compared to the corresponding character of the text and *pattern.*After all matching or not matching of the pattern, *the window* is shifted to the right until the position(- +1) in the text. According to Singh, R. & Verma, HN (2011), the efficiency of the algorithm lies in two stages:

1. Stage preprocessing, this stage is to collect full information about the *pattern* and use this information in the discovery stage.
2. Stage search, *pattern* compared to the *window* from right to left or left to right until a match or a mismatch occurs. With a character value (m <n) to be searched in the text. In thematching

stringalgorithm, the text is assumed to be in the memory, so that when we search for *a string* in an archive, then all the contents of the archive to be read first and then stored in the memory. If the *pattern* appears more than once in the text, then the search will only give the locationoutput of *pattern* foundthe first (Wulan, 2011).

Third Journals above each use the first method of *Zhu Takaoka.* algorithm *Zhu-Takaoka* is amatching algorithm *string* (StringMatching)published by Zhu Rui Feng and Tadao Takaoka in 1986. The modification of the algorithm is an algorithmmatching *Boyer-Moore stringalgorithm* created by RS Boyer and JS Moore has the same characteristicsin the search *string*process.These characteristics, namely the *preprocessing*stage, *Right-to-left*scan, *Bad character*rule,and *Good-suffix*rule.In this journal the introduction of handwriting of address detection using A Suffix Tree is an algorithm method from Zhu-Takaoka with its phase using Tree system. A Suffix Tree allows to find common share string sub-strings and access each character in a fast address. Method Analyst approach using Turbo Boyer-Moore Algorithm is Algorithm is the development of Boyer-Moore algorithm. In this algorithm is done recording segment of text that match with a suffix in pattern last done matching. The advantage of using this algorithm is that it is possible to 'jump' across the segment and be able to do turbo-shift. Turbo-shift can occur if there are substrings of the same patternwith substrings on previously-checked text. In this journal using the algorithm of CYK known algorithm this function find list variable for all substring of string and finally check list variable, for the parent string. The CYK algorithm performs better than the previous membership algorithms that have exponential timing complexity. Tuned Boyer-Moore Method Tuned Boyer-Moore algorithm is an implementation of the simplified version of Boyer-Moore. As the name implies Tuned algorithm is widely used in search engine calculations, this algorithm is very fast in practice. The string matching algorithm is to check whether the characters of the pattern fit In this journal do the calculations with BG approach.

3.12 Advantages and Disadvantages of Journal

Here's an explanation of the advantages and disadvantages of the three journals that review.

4. Conclusions And Suggestions

The conclusions and suggestions in this study are as follows.

4.1 Conclusions

The above three Journals each use the first Zhu Takaoka Method. The Zhu-Takaoka algorithm is a string matching (String Matching) algorithm published by Zhu Rui Feng and Tadao Takaoka in 1986. It is a modification algorithm of the Boyer-Moore Algorithm string algorithm made by Boyer RS and Moore JS having the same characteristics in the string search process. These characteristics are Preprocessing, Right-to-left scan, Bad character rule, and Good-suffix rule. In this journal the introduction of handwriting of address detection using A Suffix Tree is an algorithm method from Zhu-Takaoka with its phase using Tree system. A Suffix Tree allows to find common share string sub-strings and access each character in a fast address. Method Analyst approach using Turbo Boyer-Moore Algorithm is Algorithm is the development of Boyer-Moore algorithm. In this algorithm is done recording segment of text that match with a suffix in pattern last done matching. The advantage of using this algorithm is that it is possible to 'jump' across the segment and be able to do turbo-shift. Turbo-shift can occur if there is a substring of the same pattern as the substring on the previously checked text. In this journal using the algorithm of CYK known algorithm this function find list variable for all substring of string and finally check list variable, for the parent string. The CYK algorithm performs better than the previous membership algorithms that have exponential timing complexity.

Tuned Boyer-Moore Method Tuned Boyer-Moore algorithm is an implementation of the simplified version of Boyer-Moore. As the name implies Tuned algorithm is widely used in search engine calculations, this algorithm is very fast in practice. The string matching algorithm is to check whether the characters of the pattern fit In this journal do the calculations with BG approach.

4.2 Suggestions

Recommendation of journal reviewers is expected that the future will be built a data search application using Boyer-Moore Algorithm to facilitate an organization / company in dealing with problems in search data or string matching, the reviewer will design a data search application with the object of the Institute of education to apply data search or string matching using the Boyer-Moore Algorithm method.

References

[1] Anonim , Algoritma *Tuned boyer moore* ,http://www.igm.univmlv.fr/~lecroq/string/tuned bm.html,diakses pada 10 Februari 2017.

[2] Anonim, Algoritma *boyer-moore* ,http://yodmada.blogspot.co.id/2013/09/algoritma-turbo-boyer-moore_7.html?m+=1,diakses pada 10 Februari 2017.

[3] Anonim, Algoritma zhu takaoka, *http://repository.usu.ac.id/bitstream/123456789/59926/4/ chaptr%2011.pdf diakses, pada 11 Januari 2017.*

[4] Anonim, Aplikasi http://elib.unikom.ac.id/files/disk1/690/jbptunikompp-gdl-beatrixlia-34471-9-unikom_b-2.pdf

[5]Anonim Data, https://library.binus.ac.id/eColls/eThesisdoc/Bab2/2012-1-00631-SI%20Bab2002.pdf

[6] Purwoko, 2006 penyimpanan data , hhttp://library.umn.ac.id/prints/226/2/vina-bab 1-wm.pdf diakss pada 11 Februari 2017.

[7] Sulistyo, 2006 Operasi teks, http://library.umn.ac.id/prints/226/2/vina-bab 1-wm.pdf, diakss pada 13 Februari 2017.

[8] Hartoyo, Vembrina, & Meilana, 2011 Permasalahan *string searching* , http://library .umn.ac.id/prints/226/2/vina-bab 1-wm.pdf, diakss pada 13 Februari 2 017.

[9] Wikipedia, Algoritma boyer-moore https//id.m.wikipedia.org 10 Februari 2017.

[10] Jiang dan Zheng Ren Siemens AG, *A Suffix Tree Based Handwritten Chinese Address Recognition System dari Xiaoqing Ding Ninth International Conference on Document Analysis and Recognition* IEEE 0-7695-2822-8

[11] Chetan Sharma dan Rinku dari Chitkara University, Himachal Pradesh INDIA. *Problem in Deterministic Turing Machine Accepting Context Sensitive Language, Context Free Language or Regular Language 2nd International Conference on Computing for Sustainable Global Development (INDIACom* 978-9-3805-4416-8/15 2015 IEEE.

[12] Rim Attafi dan Nadia Zanzouri *Université de Tunis El Manar/ Tunis-Tunisia Bond Graph Modeling and State Estimation of Stringing Machine 7th International Conference on Modelling, Identification and Control* (ICMIC 2015) 978-0-9567157-5-3 © 2014 IEEE.