I.J. Intelligent Systems and Applications, **2011, 3, 19-25**
Published Online May 2011 in MECS (http://www.mecs-press.org/)

Building Sequence Span Attribute Model and Example Analysis

Yuqiang Sun, Yuwan Gu, Guodong Shi*

Changzhou university International Institute of Ubiquitous Computing, Jiangsu, Changzhou213164, China
Sunyuqiang0@126.com

Abstract-**Parallel parsing is one of the key technologies of parallel system. Grammatical character affects the efficiency of parallel parsing and degree of difficulty of implement. Existing methods have problems as follows: Parallelism of grammar that adapt different data object is difference, if there is a large difference between considering attribute and analysis object structure, then affect efficiency. Specific grammar parallel parsing is systematically studied. Scanning parallel parsing methods from the new angle of sequence span after word lattice distortion. Considering sequence span attribute between some specific grammars makes parsing without changing structure and data of CYK table based on the structure of word lattice CYK initialization table; In passing item of the form [i , j , B→η·] in parallel parsing item table memory structure in circle structure is adopted chain breaking technology; When indexed optimize analysis, the key algorithms of increasing the feasibility and validity of sequence span attribute、reusing parsing tree、calculating of d space function and node separating are further studied, then unification and optimize effect between analysis table middle structure and data object structure is reached. New algorithm and implement strategy of parallel parsing of specific grammar is proposed.**

Index Terms -**Parsing, Algorithm design, Parallelization**

Parallel parsing is one of the key technologies of parallel system, but grammatical character and description directly affects the efficiency of parallel parsing and degree of difficulty of implement. Further study on the aspect is help to renovate method and improve technology of parallel parsing. Scanning parallel parsing methods from considering the new angle of sequence span attribute of grammar, especially paying attention on the process of parallel parsing of specific grammar. Constructing middle structure similar with analysis object, then reaching form unification is propitious to the purpose of optimize and improving efficiency. There are theory and practice meaning about the analysis method for improved parallel parsing.

Ⅰ. INTRODUCTION

Parallel parsing technology is key technology of Parallel Compilation[1]. The processing of large log data files arising in “problem determination” in today's IT computing environments. An approach that uses Grid services to efficiently parallelize the IBM's Generic Log Adapter (GLA) is proposed. GLA is a generic parsing engine shipped with the IBM's Autonomic Computing Toolkit that has been conceived to convert proprietary log data into a standard log data event-based format in real time. However, in order to provide generic support for parsing the majority of today's unstructured log data formats the GLA makes heavy use of regular expressions that incur in performance limitations. Until now all the approaches that have been proposed to increase GLA's performance have revolved around fine-tuning the set of regular expressions used to configure the GLA for a particular log data format or writing specific parsing code. A new approach consisting in transparently parallelizing the GLA by taking advantage of its internal architecture and the fact that structuring log data is a task that lends itself very well to parallelization is proposed. A Master–Worker strategy that uses Grid services to parallelize GLA efficiently and in a completely transparent way for the user is presented[2-3]. In this paper [4] consider regular grammars with attributes ranging over integers, providing an in-depth complexity analysis. We identify relevant fragments of tractable attribute grammars, where complexity and expressiveness are well balanced. In particular, we study the complexity of the classical problem of deciding whether a string belongs to the language generated by any attribute grammar from a given class $\mathcal{C}$ (call it $\mathrm{PARSE}[\mathcal{C}]$).In this paper [5-8] propose an approach to quantitative structural information for inferring context free grammars. First, we construct derivative capacity of nonterminal symbols in context free grammar, concomitant indicator and embedded dimensional number of strings in samples set, which are called as quantitative structural information; Then, we rewrite Cocke-Younger-Kasami (CYK) algorithm for parsing in the form of the derivative set; Third, we present the construction of new production rule and the descriptive procedure for inferring with an extended CYK algorithm by the quantitative structural information. Finally, we discuss the extended CYK algorithm for inferring context free grammars[9].

Scripting languages such as R and Matlab are widely used in scientific data processing. As the data volume and the complexity of analysis tasks both grow, sequential data processing using these tools often becomes the bottleneck in scientific workflows. We describe pR, a runtime framework for automatic and

* Corresponding Author: shisungu@126.com

transparent parallelization of the popular R language used in statistical computing. Recognizing scripting languages' interpreted nature and data analysis codes' use pattern[10]. Automatic speech recognition (ASR) has become a valuable tool in large document production environments like medical dictation. While manual post-processing is still needed for correcting speech recognition errors and for creating documents which adhere to various stylistic and formatting conventions, a large part of the document production process is carried out by the ASR system. In this paper [11] propose a method for automatically reconstructing them from draft speech-recognition transcripts plus the corresponding final medical reports. The main innovative aspect of our method is the combination of two independent knowledge sources: phonetic information for the identification of speech-recognition errors and semantic information for detecting post-editing concerning format and style. Speech recognition results and final reports are first aligned, then properly matched based on semantic and phonetic similarity, and finally categorised and selectively combined into a reconstruction hypothesis. Recently, FPGA chips have emerged as one promising application accelerator to accelerate the CYK algorithm by exploiting a fine-grained custom design. In this paper [12] propose a systolic-like array structure including one master PE and multiple slave PEs for the fine-grained hardware implementation on FPGA to accelerate the CYK/inside algorithm with Query-Dependent Banding (QDB) heuristics.

Above all, parallel parsing of specific grammar has not reached applied demand. Such as speech recognition、text(Scripting) recording including that there is a large improvement space on parallel system design. A number of theory problems worth discussing.

Study scheme of the paper as figure 1 :

(1) Analyzing for attribute parameter character of sequence span, building topology model, and analyzing for parallelism of sequence span model. (2) Determining the grammar possess sequence span character or not, if falling short of it and how about the degree of nearness. (3) Recording the information which accord with the demand or reach certain degree of nearness, and constructing middle structure, designing parallel optimize parsing algorithm.

Ⅱ. THE CONVERSION FROM COMMON GRAMMAR TO CHOMSKY NORMAL FORM GRAMMAR

For common non-chomsky normal form grammar, namely right side of each production (or rule) is consisted of more than two non-terminals or terminals mingled with non-terminals, which is carried through parsing and recognition by conversion from common grammar to Chomsky normal form grammar. The purpose of conversion is that parallel parsing and recognition method apply to common grammar (include chomsky normal form and non- chomsky normal form).

Common grammar is carried through parallel parsing and recognition, first conversed from non- Chomsky normal form grammar to Chomsky normal form grammar, then carried on parallel parsing and recognition, for example: non- Chomsky normal form grammar, the rule is S→ABC and S→A+C, it can be conversed to Chomsky normal form grammar, it become the condition that right side of each production (or rule) is consisted of two non-terminals or one terminal, conversion method accord to the type of right symbolic of production, there is two conversion pattern, the one is:

For right side of production (or rule) is consisted of terminals mingled with non-terminals, namely, conversion rule is :

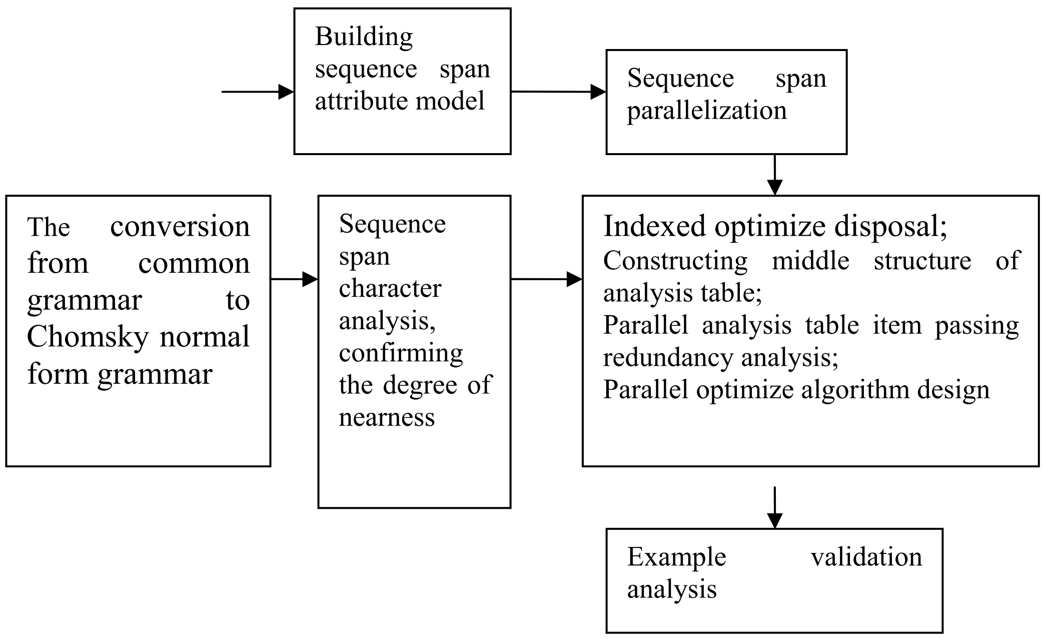
Building sequence span attribute model
Sequence span parallelization
The conversion from common grammar to Chomsky normal form grammar
Sequence span character analysis, confirming the degree of nearness
Indexed optimize disposal;
Constructing middle structure of analysis table;
Parallel analysis table item passing redundancy analysis;
Parallel optimize algorithm design
Example validation analysis

Figure 1. Study scheme of the paper

P→ A1 A2 …Ak(1) a11 a12 … a1m(1)Ak(1)+1 …Ak(2)a21 a22 … a2m(2)Ak(2) +1 …Ak(i) ai1 ai2 … ai m(i) Ak(i)+1 …Ak(s)as1 as2 … asm(s) An

Thereinto: ai∈VT Ai∈VN

Then, the rule is rewritten as follow:

P→ A1 A2 …Ak(1) B1Ak(1)+1 …Ak(2) B2Ak(2) +1 …Ak(i) Bi Ak(i)+1 …Ak(s) Bs An

B1 → a11 a12 … a1m(1)

B2 → a21 a22 … a2m(2)

… … … … …

Bi → ai1 ai2 … aim(i)

… … … … …

Bs → as1 as2 … asm(s)

In this way, it can be conversed to the first case, then it is conversed according to the first conversion method, finally, obtained grammar is Chomsky normal form, also the form is equivalent to primary form. So it can be conversed from non- Chomsky normal form grammar to Chomsky normal form grammar through rewriting grammar.

III. BUILDING MODEL STRUCTURE

An example implemented by word-lattice is given in graph 1[9]. Starting at the most left node in word-lattice, ending at the most right node in word-lattice, and each path corresponding to a recognizable sentence. These sentences conform with syntax rules.

N={S,X,Y,Z},
Σ={a,b,c,d},
P={
S—>XZ|XY
X—>XY|YX
Y—>ZX
Z—>ZY
X—>a
Y—>b
Z—>a
X—>c
Z—>d }

Figure 2 is produced in the case of the same path depth.

When dealing with word-lattice, the distinctness with original presented CYK-algorithm (view Graph 1)is not only that N(i,1) has set initial value in the step of initialization but also that N(i, j) which word-lattice form showed out with word may present in any position of CYK-table(view Graph 3).

In order to illustrate this, considering the example of word-lattice conversion given in graph 1, including six sentences between them “a a b b”, “a b d b”, “a b b a”, “b b a a”, “b c b a” and “b c d b”.

First it is regulation to increase node depth according to lattice nodes. The regulation is that the node corresponding to related time sequence t1, t2 …, in the case of the lattices produced by speech recognition machine（view figure 3）.

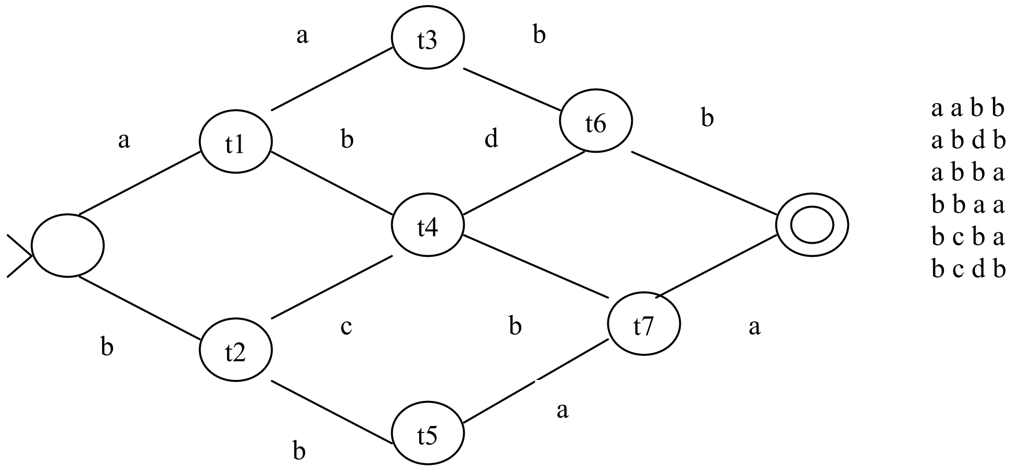

Figure 2. Word-lattice conversion including six kinds of different sentences

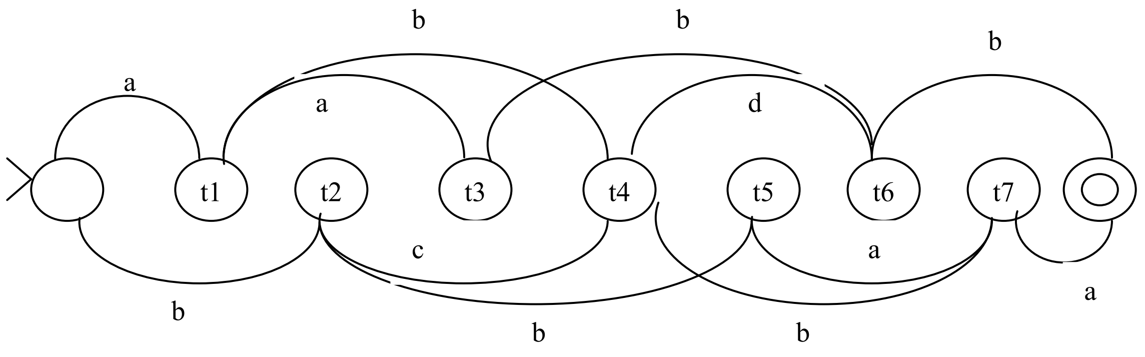

Figure 3. word-lattice graph after distortion

TABLE 1. CYK-TABLE ACCORDING TO FIGURE 3

8								
7								
6								
5								
4								
3		Y	Y	Y	Y			
2	Y	X,Z	X		Z	X,Z	Y	
1	X,Z							X,Z
j ∕ i	1	2	3	4	5	6	7	8

New representation form of word-lattice is projected to CYK-table, in table 1 and figure 3:
(1)Interval of continuous nodes correlating to corresponding column of parsing table.
(2)If arc of lattice (tm,tn)(n>m) tagged with w, then N (m+1,n-m) corresponding to row m+1, column n-m of lattice of CYK-table, filling {A:(A→w) ∈P} in the lattice (view table 1).

In table 1, i represents node t_{i-1} , j represents span a crossed, N (i,j) represents the content corresponding arc from t_{i-1} to t_{i-1+j} .

Practical steps of forming table 1:
(1) i=1,j=1 represent a arc from start node t_0 to t_1 , a is corresponded with the arc in figure 3, as well as X—>a , Z—>a, so filling X,Z in N(1,1).
(2) i=1,j=2 represent a arc from start node t_0 to t_2 , b is corresponded with the arc in figure 3, as well as Y—>b, so filling Y in N(1,2).
(3) i=1,j=3 represent a arc from start node t_0 to t_3 , there are not corresponding arc in figure 3, so filling null in N(1,3).
According to above analysis table 1 is formed.

Ⅳ. ALGORITHM DESIGN AND OPTIMIZATION

A. The basic ideas of algorithm improvement

The design idea of mixed model algorithm is that path corresponding to each sentence is separated from word-lattice syntax graph, each element of the separated graph sentences is found the corresponding position in CYK-table, the positions are not in the same layer, so the first step is to put the positions to the bottommost layer, the same as the projection mapping of stereo to planar. The second step is to parse the sentence with the improved CYK-algorithm, the improved CYK-algorithm will applying the span sum between word-lattice nodes, so that no offset position can occur in CYK-table and the parsing proceeds successfully, finally, if S∈N（1,n）, it means that the sentence can be recognized, otherwise, not.

B. Improved algorithm

The algorithm is as follows:

The first step: m is the decreased number of the nodes of word-lattice graph to one, n is the decreased number of the nodes of word-lattice graph to one, putting the table content to the bottommost layer of the table, and the corresponding row of every position is invariable, and original spans are saved in N_l , which is used to span accumulative in the second step, it is null in the table excepting the bottommost layer.

```
$N_0$=1, l=1
for i=1 to m do
   for j=1 to m do
   {
      if （$M_{i,j} \neq ø$）
            {
               $M_{i,1}$= $M_{i,j}$,
            If（j≠1）Then $M_{i,j}$= ø,
               $N_l$=j,l=l+1
            }
      end if
   }
end for
```

```
    end for
```

The second step: sentence parsing applying span accumulation.

```
    for j=2 to n do
      for i=1 to n-j+1 do
        for k=1 to j-1 do
        {
          $p=（N_0-1）+\cdots+(N_{i-1}-1)$,
          $q=（N_1-1）+\cdots+(N_{i+k-1}-1)$,
          $N（i+N_{i-1}-1,j）= N（i+N_{i-1}-1,j）\cup\{X:（X \rightarrow YZ）\in P$ with $Y\in N( i+p,k)$ and $Z\in N(i+ q +k,j-k)\}$
        }
        end for
      end for
    end for
```

C. Example analysis

Here only giving the practical analysis steps of the sentence bbaa, other sentences could be analyzed with the same method, here not give the practical analysis.

Getting separated locate graph according to the sentence bbaa, and the analysis is as follows:

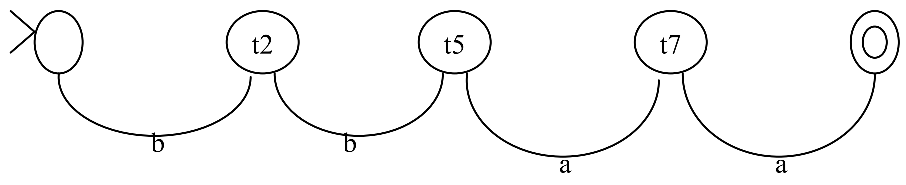

Figure 4. corresponding local graph of bbaa

TABLE 2. CYK-TABLE ACCORDING TO FIGURE 4

8								
7								
6								
5								
4								
3			Y					
2	Y					X,Z		
1								X,Z
j/i	1	2	3	4	5	6	7	8

TABLE 3. CYK-TABLE OF INITIALIZED THE BOTTOMMOST LAYER

8								
7								
6								
5								
4								
3								
2								
1	Y		Y			X,Z		X,Z
j/i	1	2	3	4	5	6	7	8

Table 3 is implemented by the first step of the algorithm, putting the content of table 2 to the bottommost layer of CYK-table, and i correcting is not changed, only j is 1, and putting the content null, where j≠1, and getting N0=1, N1=2, N2=3, N3=2, N4=1, the data is used in step two of the algorithm.

TABLE 4. THE FINAL CYK-TABLE

8								
7								
6								
5								
4	S							
3	X		S					
2	ø		X			S,Y		
1	Y		Y			X,Z		X,Z
j/i	1	2	3	4	5	6	7	8

Table 4 is got through the second step of the algorithm; the practical steps are as follows:

The first step:

j=2, i=1, k=1, then p=0, q=1

$N（1,2）= N（1,2）\cup\{X: X\rightarrow YZ\in P, Y\in N（1,1）, Z\in N（3,1）\}$

The expression is not existing in the grammar, so $N（1,2）=\{\Phi\}$;

j=2, i=2, k=1, then p=1, q=3

$N（3,2）= N（3,2）\cup\{ X: X\rightarrow YZ\in P, Y\in N$

（3,1）,Z∈N（6,1）}

There is the expression X—>YX in the grammar, so N（3,2）={X}；

j=2, i=3, k=1, then p=3, q=4

N（5,2）= N（5,2）∪{ X：X—>YZ∈P,Y∈N（6,1）,Z∈N（8,1）}

There are the expressions S—>XZ,Y—>ZX in the grammar, so N（5,2）={S,Y}.

The second step:

j=3, i=1, k=1, then p=0, q=1

N（1,3）= N（1,3）∪{X：X—>YZ∈P,Y∈N（1,1）,Z∈N（3,2）}

There is the expression X—>YX in the grammar, so N（1,3）={X}；

j=3, i=1, k=2, then p=0, q=3

N（1,3）= N（1,3）∪{X：X—>YZ∈P,Y∈N（1,2）,Z∈N（6,1）}

There is not the expression in the grammar, so N（1,3）={X}∪{Φ}={X}；

j=3, i=2, k=1, then p=1, q=3

N（3,3）= N（3,3）∪{X：X—>YZ∈P,Y∈N（3,1）,Z∈N（6,2）}

There is not the expression in the grammar, so N（3,3）={Φ}；

j=3, i=2, k=2, then p=1, q=4

N（3,3）= N（3,3）∪{X：X—>YZ∈P,Y∈N（3,2）,Z∈N（8,1）}

There is the expression S—>XZ in the grammar, so N（3,3）={Φ}∪{S}={S}.

The third step:

j=4, i=1, k=1, then p=0, q=1

N（1,4）= N（1,4）∪{X：X—>YZ∈P,Y∈N（1,1）,Z∈N（3,3）}

There is not the expression in the grammar, so N（1,4）={Φ}；

j=4, i=1, k=2, then p=0, q=3

N（1,4）= N（1,4）∪{X：X—>YZ∈P,Y∈N（1,2）,Z∈N（6,2）}

There is not the expression in the grammar, so N（1,4）={Φ}∪{Φ}={Φ}；

j=4, i=1, k=3, then p=0, q=4

N（1,4）= N（1,4）∪{X：X—>YZ∈P,Y∈N（1,3）,Z∈N（8,1）}

There is the expression S—>XZ in the grammar, so N（1,4）={Φ}∪{S}={S}.

The Graph 7 is formed according to the above analysis.

If S is in N（1,4）of table 4, it means that the sentence can be recognized.

Syntax tree from table 4 :

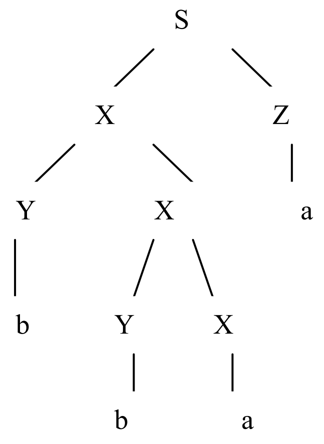

Figure 5. Parsing tree of sentence bbaa

Ⅴ. CONCLUDING REMAKES

The grammar, whose right side of each production (or rule) is consisted of more than two non-terminals or terminals mingled with non-terminals, is equivalently conversed, and the equivalence、adaptability、efficiency and recognition analysis process of grammar after conversation is analyzed. On the condition of not increasing the cost of system, non- chomsky normal form grammar can be recognized and analyzed by algorithm. It is analyzed thoroughly for the generating algorithm about initial CYK-table of word-lattice structure, and CYK-algorithm is improved by the attribute of span of time sequence after word-lattice distortion, a kind of word-lattice parsing algorithm based on improved CYK-algorithm is proposed. Without modified the structure or the data of CYK-table, parsing is proceed with software algorithm design based on initial CYK-table after word-lattice distortion. At last, the algorithm’s feasibility, running process and results, which consist with the results of theoretical analysis, are explained through an instance. Making preparations for indexed tree optimize and parallel parsing next step.

ACKNOWLEDGMENT

Supported by The project of general office of Broadcasting and Television (GD10101) and Natural Science Fund in ZheJiang(Y1100314)

REFERENCES

[1]Shen Zhiyu etc. Parallel Compilation Method. Beijing: National defence industry press.2002.

[2]Chen Huowang etc. Programming Language Compilation Theory. Beijing: National defence industry press.2003.

[3]Fatos Xhafa, Claudi Paniagua, Leonard Barolli, Santi Caballé. Using Grid services to parallelize IBM's Generic Log Adapter Journal of Systems and Software, Volume 84, Issue 1, January 2011, Pages 55-62

[4] M. Manna, F. Scarcello, N. Leone .On the complexity of regular-grammars with integer attributes Journal of Computer and System Sciences, Volume 77, Issue 2, March 2011, Pages 393-421

[5]Cristian.Ciressan,Eduardo.Sanchez,Martin.Rajman. An FPGA-based coprocessor for the parsing of context-free grammars. Proceeding of the 2000 IEEE Symposium on Field-Programmable Custom Computing Machines.

[6]Kadri Hacioglu, Wayne Ward. DIALOG-CONTEXT DEPENDENT LANGUAGE MODELING COMBINING N-GRAMS AND STOCHASTIC CONTEXT-FREE GRAMMARS. Pages:537-540,2001.

[7] A. Giannone , M. R. A. Eltantawi, P. Maresca. Recursive Parser Optimization by Rewriting Context-free Grammars.Pages:104-126,2002.

[8] Chaiyaporn Chirathamjaree. The use of context-free grammars in isolated word recognition. pages: 140-143,2004.

[9] Ming-Heng Zhang. Quantitative structural information for inferring context free grammars with an extended Cocke-Younger-Kasami algorithm Pattern Recognition Letters, In Press, Accepted Manuscript, Available online 7 January 2011

[10]Jiangtian Li, Xiaosong Ma, Srikanth Yoginath, Guruprasad Kora, Nagiza F. Samatova. Transparent runtime parallelization of the R scripting language Journal of Parallel and Distributed Computing, Volume 71, Issue 2, February 2011, Pages 157-168

[11] Stefan Petrik, Semantic and phonetic automatic reconstruction of medical dictations Computer Speech & Language, Volume 25, Issue 2, April 2011, Pages 363-385

[12] Fei Xia, Yong Dou, Dan Zhou, Xin Li .Fine-grained parallel RNA secondary structure prediction using SCFGs on FPGA Parallel Computing, Volume 36, Issue 9, September 2010, Pages 516-530