

Recognition Algorithm Research on Query Language Based on Operator Precedence Grammar

Haoyin Lv^{a, b} Tao Guo^{a, c} Zhengqi Liu^a

School of Information Engineering Long Dong University Gansu Qingyang, China.

*562755330@qq.com

Keywords: query statement; operator precedence grammar; automaton

Abstract. The end of query is the composition of SQL statements, completed by using the graphical interface. Operator precedence grammar was combined with segmentation to identify query statement in graphic interface, this paper customized the semantic action and added correctness judgment "constraints()", realized the composition of SQL and added fuzzy operator, multinational language library, process control, fault location and other language technology to make it more powerful, which was the expansion for SQL.

1 Introduction

Data management is not only to store and manage data, but also a system that focuses on information system implementation level. Language provided by DBMS is used to operate the database, and it is called Database Language. Structured Query Language (SQL) is now widely used. The supporting degree of DBMS to SQL is a key factor to evaluate whether it is successful, and an efficient and secure SQL compiler is one of the core and foundation of excellent databases. This paper used compilation technology and operator precedence grammar to compose query statement, customized relevant semantic actions, added correctness judgment "constraints()", realized the composition of SQL. While fuzzy operator, multinational language library, process control, fault location and other language technology were added, and the identification and interpretation process from natural language (take Chinese as an example) to SQL language were realized.^[1]

Keywords and limits for different application systems are different and the priority relation is also complex for the mixed operation of logic statements. The main elements of words are keyword, logic operator, comparison operators, text and parentheses. Operator precedence grammar can effectively deal with the semantic recognition and construction and has the characteristic of error locating, multi-language recognition, and automatic error correction of similar signs.^[2]

2 The Lexical Recognition Process Analysis of Searching Statements

2.1 Expression analysis

Expression compromises the following two kinds of subclauses. Field comparison operator "content" Field comparison operator numerical value These two kinds of clauses are defined as text field and numerical value field.

2.2 The contents of logic relation

The logic relation of subclauses of Where includes "and", "or" and "not".

2.3 The contents of operator

The conditional operator of subclauses of Where includes >, <, >=, <=, <>, =, !>, !<.

2.4 Pattern match operator

Pattern match operator includes “like” and “not like”.

2.5 The lexical recognition process

The diagram of lexical recognition is shown in fig.1.

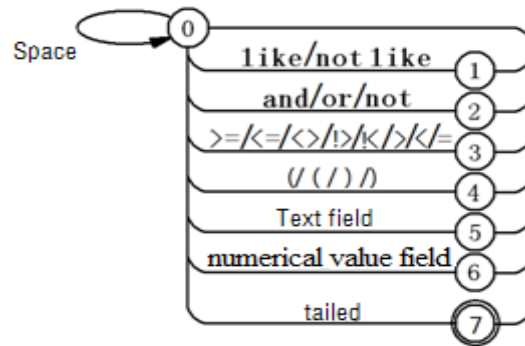


Fig.1. the diagram of lexical recognition

2.6 The construction of transferred meaning words

“and” ≡ ”and”

“or” ≡ ”or”

“not” ≡ ”not”

“greater or equal to” ≡ ”>=”

“Less than or equal” ≡ ”<=”

“not equal to” ≡ ”<>”

“no greater than” ≡ ”!>”

“not less than” ≡ ”!<”

“more than ” ≡ ”>”

“Less than ” ≡ ”<”

“equal” ≡ ”=”

To satisfy the needs of users, lexical grammar can recognize the words defined by Chinese Character (or other self-defined languages) and are not limited. The basic definition is shown above.^[3]

2.7 Intelligent error correction definition

To increase the error correction ability, lexical analyzer can recognize the parenthesis of full angle and half angle and the amount defined is not limited. The basic definition is shown below:

“(” ≡ ”(”

“)” ≡ ”)”

3 The Lexical Definition of Where Clauses in SQ

3.1 The lexical definition

Condition → condition and condition | condition or condition | Not condition | (condition) | Text field comparison operator “content” | Text field mode match operator “content” | Numerical value field comparison operator value “and” can be represented by “a”, with “or” by “o” and “not” by “n”, “(“ by “(“, “)” by “)”, “text field” by “k”, “comparison operator” by “q”, “mode match operator” by “p”, “numerical field” by “c”, “content” by “t”, “value” by “t”. The lexical grammar of Where clauses can be simply expressed as:

$$E \rightarrow EaE | EoE | nE | (E) | kpt | kqt | cqt$$

3.2 the fault-tolerance of lexical grammar

Every keyword that users use is transferred to small letter by Lcase. Users don't need to add single quote mark to text field and numerical value field. If the lexical grammar is recognized as text field, it will automatically output contents with single quote mark. ^[4]

3.3 The types of lexical grammar

$$E \rightarrow EaE | EoE | nE | (E) | kpt | cpt$$

If there are no sequential (coordinate) non-end marks on the right of the grammar, then the grammar is called operator grammar. The priority sequence of logic operators are not \triangleright and \triangleright or. Other expressions are shown below.

$$P \rightarrow \dots ab \dots$$

$$P \rightarrow \dots aQb \dots$$

$$P \rightarrow \dots aR \dots$$

$$P \rightarrow \dots Rb \dots$$

In the four expressions, there is a arithmetic priority relation between all the end marks, then this grammar is called operator priority grammar.

4 Construct Priority Relation Table from Operator Priority Grammar

From the definition of grammar, the set of end marks is $\{a, o, n, (,), k, p, t, c, q, \#\}$. The priority relation of end marks is shown in figure 1.

TABLE I. OPERATOR PRIORITY RELATION TABLE

	a	o	n	()	k	p	t	c	q	#
a	\triangleright	\triangleright	\triangleleft	\triangleleft	\triangleright	\triangleleft			\triangleleft		\triangleright
o	\triangleleft	\triangleright	\triangleleft	\triangleleft	\triangleright	\triangleleft			\triangleleft		\triangleright
n	\triangleright	\triangleright	\triangleleft	\triangleleft	\triangleright	\triangleleft			\triangleleft		
(\triangleleft	\triangleleft	\triangleleft	\triangleleft	\equiv	\triangleleft			\triangleleft		
)	\triangleright	\triangleright			\triangleright						\triangleright
k							\equiv			\equiv	
p								\equiv			
t	\triangleright	\triangleright	\triangleright		\triangleright						\triangleright
c										\equiv	
q								\equiv			
#	\triangleleft	\triangleleft	\triangleleft	\triangleleft		\triangleleft			\triangleleft		\equiv

5 Construct Priority Function

In shed priority function $f()$ and comparison priority function $g()$ are shown in table 2, The value in the shade is the results adjusted.

TABLE II. PRIORITY FUNCTION TABLE

	a	o	n	()	k	p	t	c	q	#
f()	5	3	5	1	5	7	7	8	7	7	0
g()	4	2	6	6	1	6	7	7	6	7	0

6 The Arithmetic Analysis of Operator Priority Function^[5].

```

k = 1;
s = "#";
Do
    read the next input mark into a;
If (Vt(s(k)))
    j = k;
Else
    j = k-1;
While( f(s(k))>g(a))
    { Q = s(j);
      If (Vt(s(j-1)))
        j = j-1;
      Else
        j = j-2;
      While (f(s(j))>=g(Q))
        { Let S[j+1]...S[k] as N;
          k = j + 1;
          s(k)= N;}
      If (f(s(k))<=g(a))
        { k = k + 1;
          s = s & a;}
      Else
        return False;}
While (a <> "#")
  If (s = "#E#") return True;

```

7 Lexical Recognition and Target Code Generation

The results of recognition are produced in SQL. It's produced with grammar navigation translation method to recognize the lexicon and produce the target code.^[6]

7.1 The lexical motion of logic operator

“a”, “o”, “n”, “p” are output as “space and space”, “space or space”, “space not space”, “space p space” when recognizing the grammar.

7.2 E→kpt generative semantic action

when E→kpt::

{if not constraints(k,t) then it's false to let k equal t;t="" & t & ""}

7.3 E→kqt generative semantic action

when E→kqt:

{if not constraints(k,t) then it's false to let k equal t.;t="" & t & ""}

7.4 E→cqt generative semantic action

when E→cqt:

{if not IsNumeric(t) then c it's false to let c equal t }

7.5 Semantic correctness judgment

When the conditions to be recognized maybe numerical value, text, date, some fields should be limited in value range. The semantic correctness is judged by the constraints(k,t)process. Constraints(k,t)process is Boole type used to judge the format of date field and value range of text field.
[7]

```

select case k
  case date field 1, date field 1, .....date field m
    return IsDate(t)
  case text field 1
    if t<> value 1 or t<> value 2.....or t<> value i then return false
    :
    :
  case text field n
    if t<> value 1 or t<> value 2.....or t<> value j then return false
  case else
    return true
end select

```

8 Experimental data

8.1 Natural language analysis

Community Name equal Changqing Road Community and (Name equal Zhang San and sex equal Man)

Fig.2. Chinese string

As shown in Fig 2, Chinese string has been automatically identified and distinguished by different colors in graphical interface, and the left and right parentheses were mixed input with full-width and half-angle. Through segmentation, Chinese string can turn into the form shown in Fig 3:

//Community Name// = //Changqing Road Community// and// (//Name// =//Zhang San// and// sex//=//Man//)//

Fig.3. String after segmentation

Construct grammar: kqta (kqtakqt), the analysis is as follows:

#	k
#k	p
#kp	t
#kpt	a
#E	a
#Ea	(
#Ea (k
#Ea (k	p
#Ea (kp	t
#Ea (kpt	a
#Ea (E	a
#Ea (Ea	k
#Ea (Eak	p
#Ea (Eakp	t
#Ea (Eakpt)
#Ea (EaE)

```

#Ea(E      )
#Ea(E)     #
#EaE       #
#E          #
#E#

```

According to $s \rightarrow "\# E \#"$, the analysis result is correct, the final SQL is as follows:

community like 'Changqing Road Community ' and (JumingXX_Tab.name like 'zhangsan' and JumingXX_Tab.sex like 'man')

8.2 Fault localization analysis

Natural language query is as follows:community like Changqing Road community and (sex like N and name like Daisy).Analysis procedure:

```

#          k
#k         p
#kp        t
#kpt       a
#E         a
#Ea        (
#Ea(       k
#Ea(k      p
#Ea(kp     t
Sex  cannot be N
#Ea(kpt    a
#Ea(E      a
#Ea(Ea     k
#Ea(Eak    p
#Ea(Eakp   t
#Ea(Eakpt  )
#Ea(EaE    )
#Ea(E      )
#Ea(E)     #
#EaE       #
#E          #
#E#

```

Analysis results:community like 'Changqing Road community' and (residentXX_sex like 'N' and residentXX_name like 'Daisy').Experiment results show that this algorithm's fault location is accurate, has strong ability to identify, and can add prefix automatically for relational database multi-table identification interpretation.

9 Conclusion

Based on SQL-3 standard, this paper adopts compilation technology and operator precedence grammar to compose query statement and identifies the query language close to natural language through key font library and correlative restriction to make the segmentation and semantic of the input query language with automatic recognition function.

When using the operation, all applications can be assembled compile module to communicate with SQL Server after translation. By testing practice, this algorithm provides a new approach for correct understanding and composition of query language, which is fast and convenient.

Acknowledgment

Funding: start-up funding project for the talents needed of Longdong University (XYBY140206)

Archival science and technology project of Gansu Province (2014-07)

Corresponding author:Zhengqi LIU (1963-),805115827@qq.com.

References

- [1] Li Hanjing. Spacial modeling research based natural language treatment [D] Harbin Institute of Technology. 2007.
- [2] Xu Yongdong. The key technique research of automatic abstract of multi-documents[D]. Harbin Institute of Technology. 2007.
- [3] Wang Huibin. The research and realization of special structured searching language SSQL[A]. The fourth seminar on GIS between both sides of Taiwan Straits and the tenth symposium of China GIS society.[C].2006.
- [4] Liu Shuqi, Li Zhengfan. A searching optimization algorithm based on distributed database syste.[J]Journal of Beijing Union University. 2005(01).
- [5] Liu Wei. The detection of doubted code and bad programmable habits based on original codes.[D]Beijing University of Post and Telecommunications. 2009.
- [6] Liu Zhengqi, Guo Tao, multi-dimension combined coding algorithm research based on logical operation.[J],Networking techniques.2011[09].
- [7] Compilers Principles techniques and Tools, Alfred V.Aho,Ravi Sethi Jeffrey D.Ullman, China Machine Press 2004.