

实验报告

学号:

班级:

syntax.y 代码

```
1  %{
2      #include "lex.yy.c"
3      void yyerror(const char*);
4  %}
5
6  %token LC RC LB RB COLON COMMA
7  %token STRING NUMBER
8  %token TRUE FALSE VNULL
9
10 %nonassoc ERR
11 %left RB COMMA
12 %%
13
14 Json:
15     Value
16     ;
17 Value:
18     Object
19     | Array
20     | STRING
21     | NUMBER
22     | TRUE
23     | FALSE
24     | VNULL
25     ;
26 Object:
27     LC RC
28     | LC Members RC
29     | LC Members RC STRING error { puts("Extra value after close, recovered"); }
30     ;
31 Members:
32     Member
33     | Member COMMA Members
34     | Member COMMA RC error { puts("Extra comma, recovered"); }
35     | Member COMMA error { puts("Comma instead if closing brace, recovered"); }
36     ;
37 Member:
38     STRING COLON Value %prec ERR
39     | STRING COMMA Value error { puts("Comma instead of colon, recovered"); }
40     | STRING COLON NUMBER NUMBER error { puts("Numbers cannot have leading zeroes, recovered"); }
41     | STRING Value error { puts("Missing colon, recovered"); }
```

```

42 | STRING COLON COLON Value error { puts("Double colon, recovered"); }
43 |
44 Array:
45     LB RB
46     | LB Values RB %prec ERR
47     | LB Values error { puts("Unclosed array, recovered"); }
48     | LB Values RB COMMA error { puts("Comma after the close, recovered"); }
49     | LB Values RB RB error { puts("Extra close, recovered"); }
50     | LB Values RC error { puts("mismatch, recovered"); }
51 |
52 Values:
53     value
54     | value COMMA values
55     | value COLON values error { puts("Colon instead of comma, recovered"); }
56     | value COMMA error { puts("extra comma, recovered"); }
57     | value COMMA COMMA error { puts("double extra comma, recovered"); }
58     | COMMA values error { puts("missing value, recovered"); }
59 |
60
61 %%
62
63 void yyerror(const char *s){
64     printf("syntax error: ");
65 }
66
67 int main(int argc, char **argv){
68     if(argc != 2) {
69         fprintf(stderr, "Usage: %s <file_path>\n", argv[0]);
70         exit(-1);
71     }
72     else if(!(yyin = fopen(argv[1], "r"))) {
73         perror(argv[1]);
74         exit(-1);
75     }
76     yyparse();
77     return 0;
78 }
79

```

通讨截图

```
echokovo@k: /mnt/c/Users/k x + v
(PoJail) echokovo@k:/mnt/c/Users/k/Desktop/lab4/BUPT-Compiler-Lab4-2025Fall-master/jp$ make jp
flex lex.l
bison -t -d syntax.y
gcc syntax.tab.c -lfl -ly -o jp.out
(PoJail) echokovo@k:/mnt/c/Users/k/Desktop/lab4/BUPT-Compiler-Lab4-2025Fall-master/jp$ python jsonparser_test.py
For file fail02.json:
-----
["Unclosed array"]
-----
syntax error: Unclosed array, recovered
#####
For file fail04.json:
-----
["extra comma",]
-----
syntax error: extra comma, recovered
#####
For file fail05.json:
-----
["double extra comma",,]
-----
syntax error: double extra comma, recovered
#####
For file fail06.json:
-----
[ , "<-- missing value"]
-----
syntax error: missing value, recovered
#####
For file fail07.json:
-----
["Comma after the close"],
-----
syntax error: Comma after the close, recovered
#####
For file fail08.json:
-----
["Extra close"]]
-----
syntax error: Extra close, recovered
#####
For file fail09.json:
-----
{"Extra comma": true,}
-----
syntax error: Extra comma, recovered
#####
For file fail10.json:
-----
{"Extra value after close": true} "misplaced quoted value"
-----
syntax error: Extra value after close, recovered
#####
For file fail13.json:
-----
{"Numbers cannot have leading zeroes": 013}
-----
syntax error: Numbers cannot have leading zeroes, recovered
#####
For file fail19.json:
-----
{"Missing colon" null}
-----
syntax error: Missing colon, recovered
#####
For file fail20.json:
-----
{"Double colon"::: null}
-----
syntax error: Double colon, recovered
#####
```

```

syntax error: Double colon, recovered
#####
For file fail21.json:
-----
{"Comma instead of colon", null}
-----
syntax error: Comma instead of colon, recovered
#####
For file fail22.json:
-----
["Colon instead of comma": false]
-----
syntax error: Colon instead of comma, recovered
#####
For file fail32.json:
-----
{"Comma instead if closing brace": true,
-----
syntax error: Comma instead if closing brace, recovered
#####
For file fail33.json:
-----
["mismatch"]}
-----
syntax error: mismatch, recovered
#####
Recovered/Total: 15/15
(PoJail) echokovo@k:/mnt/c/Users/k/Desktop/lab4/BUPT-Compiler-Lab4-2025Fall-master/jp$ |

```

设计思路

对于情况 {"Extra value after close": true} "misplaced quoted value"

Object:

```

    LC RC
  | LC Members RC
  | LC Members RC STRING error { puts("Extra value after close, recovered"); }
;

```

直接构造对应形式

对于情况 {"Extra comma": true,}

Members:

```

    Member
  | Member COMMA Members
  | Member COMMA RC error { puts("Extra comma, recovered"); }
  | Member COMMA error { puts("Comma instead if closing brace, recovered"); }
;

```

Member:

同样直接构造对应形式

对于未闭合的形式 {"Comma instead if closing brace": true,

Members:

```

    Member
  | Member COMMA Members
  | Member COMMA RC error { puts("Extra comma, recovered"); }
  | Member COMMA error { puts("Comma instead if closing brace, recovered"); }
;

```

Member:

不能直接在非终结符 Object 中直接构造 LC Members COMMA 的形式 会出现冲突

故在非终结符 Member 中构造对应的错误形式

其余情况同理

对于以下特殊的情况

```
Member:
| STRING COLON Value %prec ERR
| STRING COMMA Value error { puts("Comma instead of colon, recovered"); }
| STRING COLON NUMBER NUMBER error { puts("Numbers cannot have leading zeros"); }
```

```
Array:
| LB RB
| LB Values RB %prec ERR
| LB Values error { puts("Unclosed array, recovered"); }
| LB Values RB COMMA error { puts("Comma after the close"); }
```

使用 `%prec ERR` 将该推导式优先级降低以解决 shift/reduce 冲突