# 词语识别

以c语言子集为例

DELIM-> , | ; | ( | ) | { | } | [ | ]

NL -> \n

SPACE -> \t|\s

EMPTY -> (NL|SPACE) EMPTY | (NL|SPACE)

BIN\_DIGIT -> 0 | 1

BIN\_NUMBER -> BIN\_DIGIT BIN\_NUMBER |

DEC\_DIGIT -> 0 1 2 3 4 5 6 7 8 9

DECILMAL\_NUMBER -> DIGIT DECILMAL|DIGIT

HEX\_DIGIT -> DEC\_DIGIT | a-f|A-F

INT\_NUMBER -> DIGIT

NUM

LETTER -> a-z| A-Z

OP->- | + | \* | /

UNDERSCORE\_WORD -> (LETTER|\_|DIGIT) UNDERSCORE\_WORD| (LETTER|\_|DIGIT)

ID-> (\_|LETTER) UNDERSCORE\_WORD

EXPR->

注：E表示空

## 步骤1：识别二进制，十进制，十六进制数

## 步骤2：编写通用自动机

## 步骤3：实现预处理

## 步骤4：正确输出文法单词序列

# 正则表达式实现

首先最好基于Perl的风格 ： <http://www.pcre.org/>

google re2库参照 ： <https://github.com/google/re2>

一个关于NFA DFA的争论： <http://stackoverflow.com/questions/3978438/dfa-vs-nfa-engines-what-is-the-difference-in-their-capabilities-and-limitations>

一个声称不使用回溯的NFA实现 : <https://swtch.com/~rsc/regexp/regexp1.html>

结果是，多数语言都乐于使用NFA的实现，尽管它可能运行指数时间。

# 附录：LEX的使用