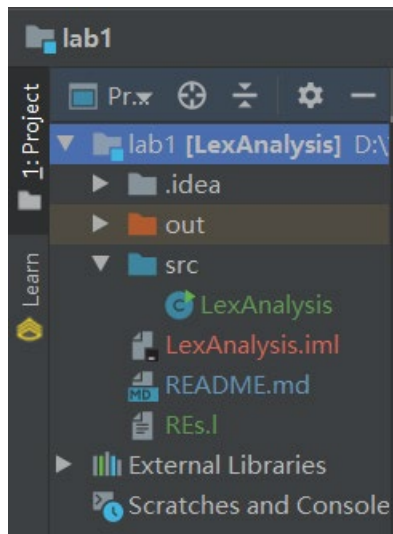


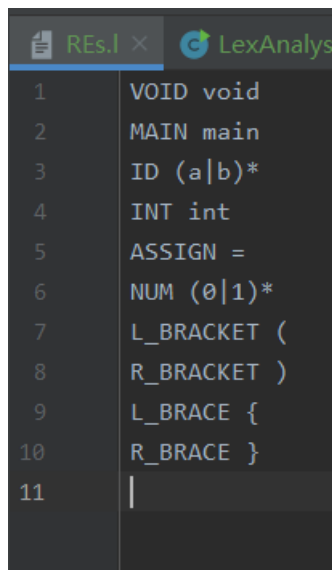
Lex 实验报告模板

学号: 171250592 姓名: 王有鑫

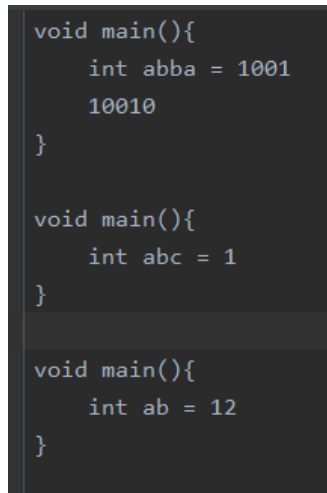
源文件目录截图: (请一定将提交的 src 文件与截图对应)



REs. 1 资源文件截图: (具体内容怎么定义都可以)



输入文件/流内容截图: (Stream of characters)



输出 token 的截图：（下图只是助教自己的实现，你们怎么输入输出都可以）

```
void main(){
    int abba = 1001
    10010
}
```

```
<VOID,void>
<MAIN,main>
<L_BRACKET,(>
<R_BRACKET,>
<L_BRACE,{>
<INT,int>
<ID,abba>
<ASSIGN,=>
<NUM,1001>
<NUM,10010>
<R_BRACE,}>
```

```
LexAnalysis x
D:\jdk\jdk1.8.0_221\
void main(){
    int abc = 1
}
```

```
<VOID,void>
<MAIN,main>
<L_BRACKET,(>
<R_BRACKET,>
<L_BRACE,{>
<INT,int>
Invalid Input!
```

```
void main(){
    int ab = 12
}<VOID,void>
<MAIN,main>
<L_BRACKET,(>
<R_BRACKET,>
<L_BRACE,{>
<INT,int>
<ID,ab>
<ASSIGN,=>
Invalid Input!
```

回答以下问题：

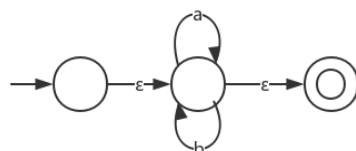
- 简要说明在实验中实现词法分析需要哪几个具体的步骤？（2-3 行说明就行了）
 - 自己定义将要实现的能够分析的 REs
 - 将每个小的 RE 转换为小的 NFA
 - 将这些 NFA 合并为一个大的 NFA
 - 将这个 NFA 转化为 DFA
 - 将 DFA 优化为 DFA0。
 - 根据 DFA0 进行编程，识别输入的字符流，并返回对应的 token 序列
- 具体说明其中重要的转化步骤怎么实现（当然可以用你自己的方式说明，越详细越好）

- RE -> NFA

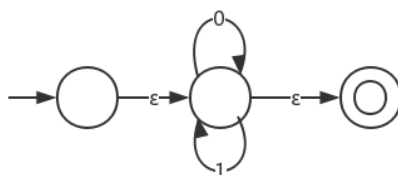
- VOID



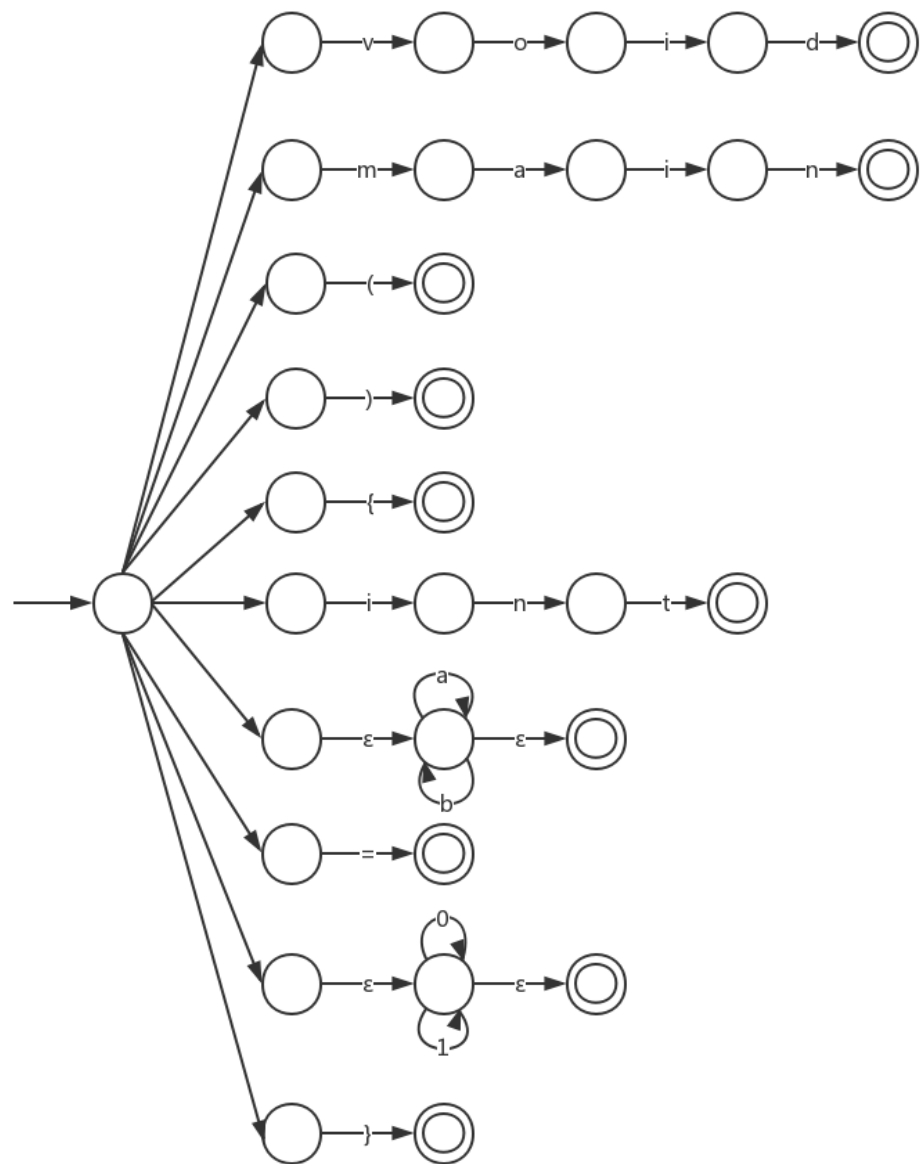
- ID (以 ab 代替)



- NUM (以 01 代替)



② 合并 NFA



③ NFA \rightarrow DFA0
- 转换表

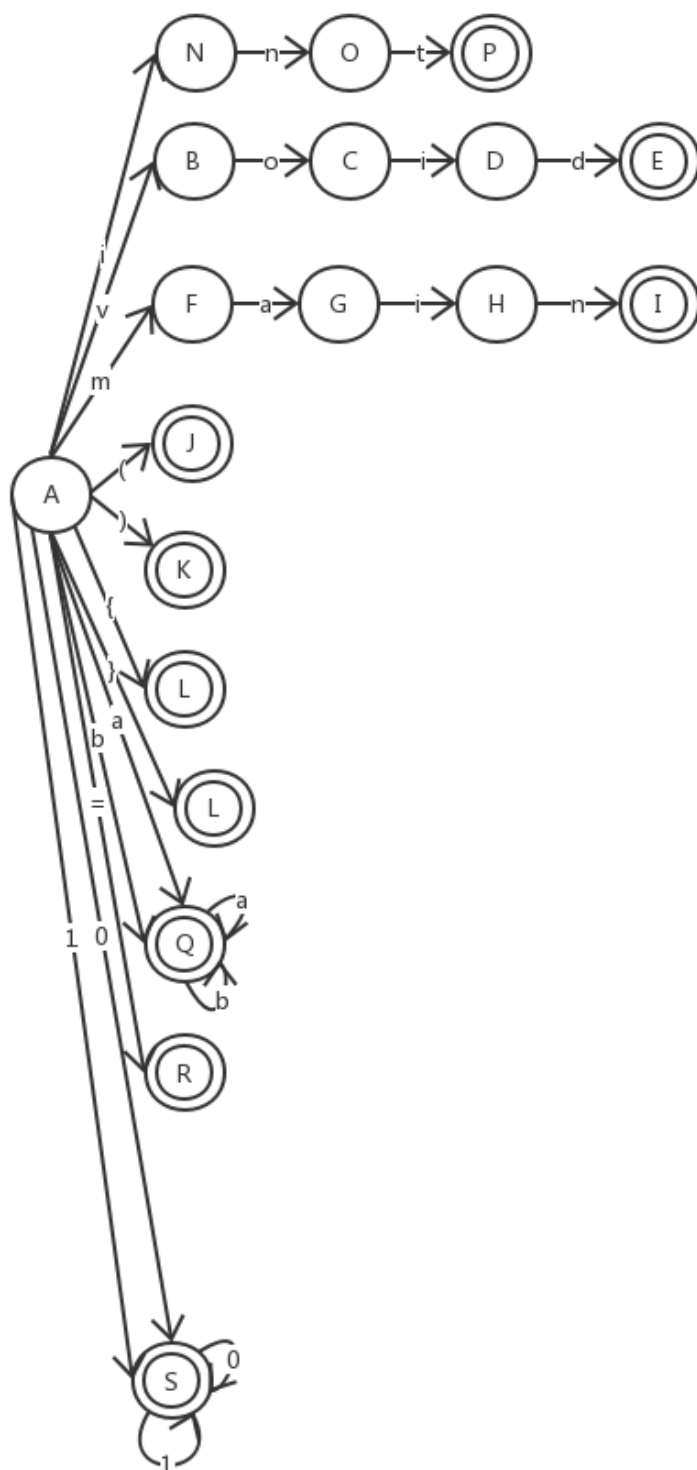
	v	o	i	d	m	a	n	(
I0={0,1,6,11,13,15,17,19,23,24,25,26,28,29,30}	I1={2}		I14={20}		I6={7}	I17={24,25}		I10={12}
I1={2}	ø	I2={3}						
I2={3}			I4={4}					
I4={4}				I5={5}				
I5={5}								
I6={7}						I7={8}		
I7={8}			I8={9}					
I8={9}				I9={10}				
I9={10}								
I10={12}								
I11={14}								
I12={16}								
I13={18}								
I14={20}							I15={21}	
I15={21}								
I16={22}								
I17={24,25}						I17={24,25}		
I18={27}								
I19={29,30}								

(续表)

)	{	}	t	b	=	0	1
I0={0,1,6,11,13,15,17,19,23,24,25,26,28,29,30}	I11={14}	I12={16}	I13={18}		I17={24,25}	I18={27}	I19={29,30}	I19={29,30}
I1={2}								
I2={3}								
I4={4}								
I5={5}								
I6={7}								
I7={8}								
I8={9}								
I9={10}								
I10={12}								
I11={14}								
I12={16}								
I13={18}								
I14={20}								
I15={21}				I16={22}				
I16={22}								
I17={24,25}								
I18={27}					I17={24,25}			
I19={29,30}							I19={29,30}	I19={29,30}

- DFA0

最后得到优化后的状态为: {I0}、{I1}、{I2}、{I4}、{I5}、{I6}、{I7}、{I8}、{I9}、{I10}、
{I11}、{I12}、{I13}、{I14}、{I15}、{I16}、{I17}、{I18}、{I19}, 分别对应于 A-S



3. 是否有 Error Handling，具体如何实现的？

- 对于不符合规定语法要求的输入，通过报错输出 “Invalid Input!” 来警示，并结束接

下来的字符流读入、

4.实验中出现的和相应的解决办法。

- 问题：代表 ID 和 NUM 的字母和数字比较复杂，求解 NFA 和 DFA 的时候比较困难
- 利用 a 或 b 代表 ID，利用 0 或 1 代表 NUM

5.对实验的评价和感觉。(简短一些就好)

通过这次实验，利用程序实现将输入字符流识别为对应的 token 序列，加深了自己对于如何编译给定语法的输入字符流的理解，也进一步加深了由一门语言转化为最终程序的中间算法的理解和认识。