

实验二实验报告

实验目的

实验内容

实验环境描述

实验设计思路

核心算法

removeEpsilonProductions

removeUnitProductions()

removeUselessSymbols()

程序中的类、方法和属性

程序的输入格式，输出格式

输入格式

输出格式

程序的测试用例，输入，输出以及执行效果

测试用例一（来自于实验说明）

测试用例二（来自于课本例题）

测试用例三（来自于课本例题）

测试用例四（来自于课本例题）

测试用例五（来自于互联网）

小组成员：

班级	姓名	学号
2021211321	戴鑫旺	2021211016
2021211321	张一达	2021211005
2021211315	赵先哲	2021211004

成员分工：

代码	戴鑫旺、赵先哲
----	---------

调试	张一达、赵先哲
文档	戴鑫旺、张一达

实验目的

编程实现上下文无关文法的变换算法，用于消除文法中的 ϵ 产生式、单产生式、以及无用符号。

实验内容

上下文无关文法对产生式的右部没有限制，这种完全自由的形式有时会对文法分析带来不良影响，而对文法的某些限制形式在应用中更方便。通过上下文无关文法的变换，在不改变文法的语言生成能力的前提下，可以消除文法中的 ϵ 产生式、单产生式、以及无用符号。

要求编程实现消除上下文无关文法中的 ϵ 产生式、单产生式、以及无用符号的算法。输入是一个上下文无关文法，输出是与该文法等价的没有 ϵ 产生式、单产生式、无用符号的上下文无关文法。可使用如下文法中的产生式进行程序的正确性验证。 $S \rightarrow a|bA|B|ccD$

$A \rightarrow abB|c$

$B \rightarrow aA$

$C \rightarrow ddC$

$D \rightarrow ddd$

实验环境描述

- Java软件开发工具：IntelliJ IDEA Ultimate 2023.1.2
- JDK版本：JDK 1.8.0_212
- 操作系统：Windows 11

实验设计思路

- 完成对产生式类、文法变换类的成员变量的定义
- 在主函数中输出提示信息，提示用户输入文法的产生式，在文法变换类中实现根据产生式集合构建文法
- 在文法变换类中实现消除文法的epsilon产生式

- 在文法变换类中实现消除文法的单一产生式
- 在文法变换类中实现消除文法的无用符号
- 在主函数中依次调用文法变换类的三个成员函数实现文法的变换并格式化输出变换后的结果

核心算法

该实验中的核心方法包括 `removeEpsilonProductions`、`removeUnitProductions`、`removeUselessSymbols`，分别用于消除文法的 ϵ 产生式、单一产生式和无用符号。工具方法包括 `removeUselessProductions`（删除无用产生式）、`removeUselessNonTerminals`（删除无用非终结符）、`print`（格式化输出文法）

removeEpsilonProductions

`removeEpsilonProductions()`实现了从上下文无关文法中移除 `epsilon` 产生式。

大致思路为：首先找出能够推导出空串的非终结符集合 N' ，然后根据 N' 来构造新的产生式，最后将新的产生式加入原有的产生式集合中，并删除所有原有的 `epsilon` 产生式。

具体地，使用两个集合 N_0 和 N_1 存储非终结符集合，初始时 N_0 为空集， N_1 包含所有能够推导出空串的非终结符。在循环中，如果当前的 N_1 与上一轮循环中的 N_0 不同，则需要重新计算 N_1 。对于每个产生式，如果其右侧所有符号均在 N' 中，则将该产生式的左侧符号加入到 N_1 中。循环直到 N_1 不再发生变化为止。

在得到了能够推导出空串的非终结符集合后，遍历所有产生式，对于每个产生式，生成所有可能的组合方式并构造新的产生式，其中每个可空符号可以被替换成空串或者不替换。如果起始符号 S 在 N' 中，则需要增加一个新的起始符号 Z 并添加产生式 $Z \rightarrow S|\epsilon$ ；否则直接将起始符号设置为 S 。

最后，将新的产生式集合加入原有的产生式集合，并删除所有原有的 `epsilon` 产生式。

removeUnitProductions()

`removeUnitProductions()`实现了移除单一生成式。

大致思路为：遍历所有产生式，对于每个单一生成式 $A \rightarrow B$ ，将所有能够推导出 A 的非终结符替换成 B ，并构造新的产生式加入到原有的产生式集合中。

算法分为两个步骤：

1. 首先需要找出一个非终结符集合 N_A ，其中 $N_A[A]$ 表示所有能够直接或间接推导出 A 的非终结符。使用两个集合 N_0 和 N_1 存储非终结符集合，初始时 N_0 包含所有单一生成式的左部符号， N_1 为空集。在循环中，对于每个产生式 $B \rightarrow \alpha$ ，如果 B 属于 N_0 ，且 α 不是单一符号，则将 α 中所有可以被替换成

N_0 中的符号的位置全部替换成对应的符号。然后将可以推导出 B 的符号加入到 N_1 中，直到 N_1 不再发生变化为止。

2. 在得到 N_A 后，遍历所有产生式，对于每个单一生成式 $A \rightarrow B$ ，将所有能够推导出 A 的非终结符都替换成 B ，并构造新的产生式加入到原有的产生式集合中。这里需要注意，如果替换后得到的产生式右部仍然是 A ，则不能将其加入到新的产生式集合中，否则会导致无限递归。

removeUselessSymbols()

removeUselessSymbols()用于在上下文无关文法中删除无用符号。

大致思路为：首先删除所有的无用非终结符，即推不出终结字符串的非终结符，再删除无用符号，即不能由起始符号直接或间接推导出的符号。

删除无用非终结符用到了工具函数removeUselessNonTerminals()，其大致步骤如下：

- (1) $N_0 = \emptyset$ ， N_0 为非终结符集合
- (2) $N' = \{A | A \rightarrow \alpha \text{ 且 } \alpha \text{ 属于 } T^*\}$ ， N' 为非终结符集合
- (3) 如果 $N' \neq N_0$ ，则转(4)，否则转(6)
- (4) $N_0 = N'$
- (5) $N' = N_0 \cup \{A | A \rightarrow \alpha \text{ 且 } \alpha \text{ 属于 } (N_0 T)^*\}$ ，转(3)
- (6) $N_1 = N'$ ， N_1 为新的非终结符集合

删除无用符号的步骤大致如下：

- (1) $N_0 = \{S\}$
- (2) $N' = \{X | A \text{ 属于 } N_0 \text{ 且 } A \rightarrow \alpha X \beta\} \cup N_0$ ， N' 为有用符号集合， X 属于 NUT
- (3) 如果 $N' \neq N_0$ ，则转(4)，否则转(5)
- (4) $N_0 = N'$ 转(2)
- (5) $N_1 = N' \cap N$ ， $T_1 = N' \cap T$

程序中的类、方法和属性

```
1 class Production {
2     private final Character left; // 左部
3     private final String right; // 右部
4
5     // 构造函数
6     public Production(Character left, String right) {
7     }
8
9     public Character getLeft() {
10    }
11
12    public String getRight() {
13    }
14
15    @Override
16    public String toString() {
17    }
18
19    @Override
20    public boolean equals(Object obj) {
21    }
22
23    @Override
24    public int hashCode() {;
25    }
26 }
```

```
1 class GrammarTransformer {
2     private Set<Character> nonTerminalSet; // 非终结符集合
3     private Set<Character> terminalSet; // 终结符集合
4     private Set<Production> productionSet; // 产生式集合
5     private Character startSymbol; // 开始符号
6     private Set<Character> nullableSymbols; // 可空符号集合
7
8     // 删除无用产生式
9     public void removeUselessProductions(){
10    }
11
12    // 消除epsilon产生式
13    public void removeEpsilonProductions(){
14    }
15
16    // 消除单一产生式
17    public void removeUnitProductions(){
18    }
19
20    // 删除无用非终结符
21    public void removeUselessNonTerminals(){
22    }
23
24    // 消除无用符号
25    public void removeUselessSymbols(){
26    }
27
28    // 根据产生式集合构造文法
29    public GrammarTransformer(Set<Production> productionSet) {
30    }
31
32    // 输出文法的非终结符、终结符、产生式集合、开始符号
33    public void print() {
34    }
35 }
```

程序的输入格式，输出格式

输入格式

执行程序后，程序输出提示信息如下：

请输入文法的产生式：

1. 每个产生式占一行，产生式的左部和右部用 \rightarrow 分隔
2. 产生式的右部可以用多个“|”分隔，产生式的左部是单个非终结符
3. 非终结符用大写字母表示，终结符用大写字母以外的字符表示
4. 空串用 ϵ 表示，输入以“#”结束

输入：

输入为多个产生式，示例如下：

请输入文法的产生式：

1. 每个产生式占一行，产生式的左部和右部用 \rightarrow 分隔
2. 产生式的右部可以用多个“|”分隔，产生式的左部是单个非终结符
3. 非终结符用大写字母表示，终结符用大写字母以外的字符表示
4. 空串用 ϵ 表示，输入以“#”结束

输入：

$S \rightarrow a|bA|B|ccD$

$A \rightarrow abB|\epsilon$

$B \rightarrow aA$

$C \rightarrow ddC$

$D \rightarrow ddd$

#

输出格式

输入产生式后，首先会根据产生式集合构建文法并输出，示例如下：

非终结符：

[A, B, S, C, D]

终结符：

[a, b, c, d]

开始符号：

S

产生式：

$A \rightarrow abB|\epsilon$

$B \rightarrow aA$

$C \rightarrow ddC$

$D \rightarrow ddd$

$S \rightarrow B|ccD|bA|a$

之后会按照同样格式依次输出消除epsilon产生式后的文法，消除单一产生式后的文法，消除无用符号后的文法，示例如下：

消除epsilon产生式后的文法： 非终结符： [A, B, S, C, D] 终结符： [a, b, c, d] 开始符号： S 产生式： A->abB B->a aA C->ddC D->ddd S->B b ccD bA a	消除单一产生式后的文法： 非终结符： [A, B, S, C, D] 终结符： [a, b, c, d] 开始符号： S 产生式： A->abB B->a aA C->ddC D->ddd S->b ccD bA aA a	消除无用符号后的文法： 非终结符： [A, B, S, D] 终结符： [a, b, c, d] 开始符号： S 产生式： A->abB B->a aA D->ddd S->b ccD bA aA a
---	---	---

程序的测试用例，输入，输出以及执行效果

测试用例一（来自于实验说明）

题目如下

可使用如下文法中的产生式进行程序的正确性验证。
 $S \rightarrow a|bA|B|ccD$
 $A \rightarrow abB|\epsilon$
 $B \rightarrow aA$
 $C \rightarrow ddC$
 $D \rightarrow ddd$

输入	输出		
	消除ε产生式后的文法	消除单一产生式后的文法	消除无用符号后的文法

<pre>S->a bA B ccD A->abB ε B->aA C->ddC D->ddd #</pre>	<pre>消除epsilon产生式后的文法: 非终结符: [A, B, S, C, D] 终结符: [a, b, c, d] 开始符号: S 产生式: A->abB B->a aA C->ddC D->ddd S->B b ccD bA a</pre>	<pre>消除单一产生式后的文法: 非终结符: [A, B, S, C, D] 终结符: [a, b, c, d] 开始符号: S 产生式: A->abB B->a aA C->ddC D->ddd S->b ccD bA aA a</pre>	<pre>消除无用符号后的文法: 非终结符: [A, B, S, D] 终结符: [a, b, c, d] 开始符号: S 产生式: A->abB B->a aA D->ddd S->b ccD bA aA a</pre>
--	---	---	---

测试用例二（来自于课本例题）

题目如下（删除无用符号）

例 1 设上下文无关文法 $G=(N,T,P,S)$, 其中
 $N=\{S,A,B\}$,
 $T=\{a\}$

生成式 P 如下:

$S \rightarrow a,$
 $S \rightarrow AB,$
 $A \rightarrow a$

输入	输出（省略了本题中无关的输出）
	消除无用符号后的文法

$S \rightarrow a$ $S \rightarrow AB$ $A \rightarrow a$ $\#$	消除无用符号后的文法： 非终结符： $[S]$ 终结符： $[a]$ 开始符号： S 产生式： $S \rightarrow a$
--	---

课本答案：

最后得出不存在无用符号的等效文法为

$$G_1 = (\{S\}, \{a\}, \{S \rightarrow a\}, S)$$

程序输出结果与课本答案一致。

测试用例三（来自于课本例题）

题目如下：

例 2 设上下文无关文法 $G=(N,T,P,S)$ ，其中

$$N = \{S\},$$

$$T = \{a,b\}$$

生成式 P 如下：

$$S \rightarrow aSbS \mid bSaS \mid \epsilon$$

G 是有 ϵ 生成的文法，除 $S \rightarrow \epsilon$ 外， S 出现在生成式的右边用上述算法找出无 ϵ 生成的等效文法 $G_1=(N_1,T,P_1,S_1)$ 。

输入	输出（省略了本题中无关的输出）
	消除 ϵ 产生式后的文法

$S \rightarrow aSbS \mid bSaS \mid \varepsilon$
#

消除epsilon产生式后的文法:

非终结符:

[S, Z]

终结符:

[a, b]

开始符号:

Z

产生式:

$S \rightarrow abS \mid baS \mid aSbS \mid bSaS \mid ba \mid ab \mid aSb \mid bSa$

$Z \rightarrow S \mid \varepsilon$

课本答案:

$S_1 \rightarrow \varepsilon \mid S,$

$S \rightarrow aSbS \mid aSb \mid abS \mid ab \mid bSaS \mid bSa \mid baS \mid ba$

程序输出结果与课本答案一致。

测试用例四（来自于课本例题）

题目如下（消除单生成式）:

例 3 设上下文无关文法 $G=(N,T,P,S)$, 其中

$N=\{S,A,B\},$

$T=\{(,),+,* ,a\}$

生成式 P 如下:

$S \rightarrow S+A \mid A,$

$A \rightarrow A*B \mid B,$

$B \rightarrow (S) \mid a$

输入	输出（省略了本题中无关的输出）
	消除单产生式后的文法
$S \rightarrow S+A \mid A$ $A \rightarrow A*B \mid B$ $B \rightarrow (S) \mid a$ $\#$	消除单一产生式后的文法： 非终结符： $[A, B, S]$ 终结符： $[a, (,), *, +]$ 开始符号： S 产生式： $A \rightarrow a \mid A*B \mid (S)$ $B \rightarrow a \mid (S)$ $S \rightarrow S+A \mid A*B \mid (S) \mid a$

课本答案：

(3) 得出文法 G_1 的生成式如下：

$$\begin{aligned}
 S &\rightarrow S + A \mid A * B \mid (S) \mid a, \\
 A &\rightarrow A * B \mid (S) \mid a, \\
 B &\rightarrow (S) \mid a
 \end{aligned}$$

与课本答案一致。

测试用例五（来自于互联网）

题目如下：

$S \rightarrow C \mid aA \mid aB$

$A \rightarrow bB \mid cc$

$B \rightarrow \epsilon \mid d$

$C \rightarrow c \mid ddC$

$D \rightarrow e \mid eDe$

输入	输出		
	消除 ϵ 产生式后的文法	消除单一产生式后的文法	消除无用符号后的文法
$S \rightarrow C aA aB$ $A \rightarrow bB cc$ $B \rightarrow \epsilon d$ $C \rightarrow c ddC$ $D \rightarrow e eDe$ $\#$	消除 ϵ 产生式后的文法: 非终结符: $[A, B, C, S, D]$ 终结符: $[a, b, c, d, e]$ 开始符号: S 产生式: $A \rightarrow bB cc b$ $B \rightarrow d$ $C \rightarrow ddC c$ $D \rightarrow e eDe$ $S \rightarrow C aA aB a$	消除单一产生式后的文法: 非终结符: $[A, B, C, S, D]$ 终结符: $[a, b, c, d, e]$ 开始符号: S 产生式: $A \rightarrow bB cc b$ $B \rightarrow d$ $C \rightarrow ddC c$ $D \rightarrow e eDe$ $S \rightarrow aA aB a ddC c$	消除无用符号后的文法: 非终结符: $[A, B, C, S]$ 终结符: $[a, b, c, d]$ 开始符号: S 产生式: $A \rightarrow bB cc b$ $B \rightarrow d$ $C \rightarrow ddC c$ $S \rightarrow ddC c aA aB a$

答案为:

$A \rightarrow b|bB|cc$

$B \rightarrow d$

$C \rightarrow c|ddC$

$S \rightarrow a|aA|aB|c|ddC$

程序输出与答案一致。