

# 实验报告（上下文无关文法的变换算法）

课程编号：3132112040 实践课程名称：形式语言与自动机 学年：2022-2023 学期：4

学生姓名	张梓良	学号	2021212484
指导教师姓名	杨正球	起止时间	2023-05-31 至 2023-06-18
项目名称	设计上下文无关文法的变换算法		
项目内容	<p>负责内容：</p> <ol style="list-style-type: none"><li>实验问题分析。</li><li>报告编写。</li></ol> <p>实验问题分析和函数设计：</p> <ol style="list-style-type: none"><li>输入： 输入是通过 <code>read_input</code> 函数实现的。输入内容为一个名为 <code>input.txt</code> 的文件。该文件中包含非终结符集合、终结符集合、产生式集合和起始符号。 在该函数中，首先通过 <code>getline</code> 函数从输入中读取第一行，即非终结符集合，然后解析出其中的每个非终结符，并将其加入集合 <code>N</code> 中。 接着读取第二行，即终结符集合，解析出其中的每个终结符，并将其加入集合 <code>T</code> 中。然后跳过第三行，开始解析产生式集合。通过 <code>while</code> 循环逐行读取输入，对于每一行，首先解析出产生式左部的非终结符，然后解析出右部的每个可能性，并将它们作为 <code>vector&lt;char&gt;</code> 类型的元素加入到一个 <code>vector&lt;vector&lt;char&gt;&gt;</code> 中，最后将该产生式加入到 <code>P</code> 中。 最后解析出起始符号 <code>S</code>，并返回读取到的所有信息。</li><li>对上下文无关文法进行变换： 消除 <math>\epsilon</math> 产生式的过程如下：<ol style="list-style-type: none"><li>找到所有可以直接推出 <math>\epsilon</math> 的非终结符；</li><li>不断找到新的可以推出 <math>\epsilon</math> 的非终结符，直到集合不再变化；</li><li>根据新的产生式集合生成新的产生式集合 <code>P1</code>。</li></ol>消除单产生式的过程如下：<ol style="list-style-type: none"><li>遍历原始产生式集合 <code>P</code> 中的每个非终结符 <code>A</code>；</li><li>找出所有可以直接推出单个非终结符的产生式，并将这些非终结符加入 <code>new_nonterminals</code> 中；</li><li>不断扩展 <code>new_nonterminals</code>，直到 <code>new_nonterminals</code> 不再变化为止；</li><li>构建新的产生式列表；</li><li>去除重复的产生式，如果新的产生式列表为空，则添加一个产生式 <code>A <math>\rightarrow</math> N</code>。</li></ol>消除无用符号的过程如下：<ol style="list-style-type: none"><li>找出所有能直接推导出终结符的非终结符，并将其加入 <code>N1</code> 集合中；</li><li>不断扩展 <code>N1</code> 集合，直到 <code>N1</code> 不再发生变化；</li><li>从原始产生式集合 <code>P</code> 中删除所有左部非终结符不在 <code>N1</code> 集合中的产生</li></ol></li></ol>		

	<p>式；</p> <p>4. 反向遍历所有的“可达性”非终结符，并将它们加入到新的非终结符集合 <code>new_nonterminals</code> 中；</p> <p>5. 同样从原始产生式集合 <code>P</code> 中删除所有左部非终结符不在 <code>new_nonterminals</code> 集合中的产生式。</p> <p>3. 输出：</p> <p>输出是通过 <code>print_grammar</code> 函数实现的。</p> <p>首先输出非终结符集合 <code>N</code> 和终结符集合 <code>T</code>，并通过循环遍历输出各个元素，其中使用了迭代器并判断是否为最后一个元素来避免末尾多余的逗号。然后按照 <code>key</code> 对产生式集合 <code>P</code> 中的元素进行排序，并依次输出左部符号和右部符号，其中右部符号按照字典序排序，同样使用了迭代器并判断是否为最后一个元素来避免末尾多余的竖线。最后输出开始符号 <code>S</code> 即可。</p> <p>需要注意的是，这里使用了一些 C++11 中新增的语法特性，比如 <code>auto</code> 用于获取迭代器类型和范围 <code>for</code> 循环，还有 <code>lambda</code> 表达式和 <code>map</code> 排序等。</p>
结 论	<p>通过本次实验，我们学习了上下文无关文法的相关定义和性质，掌握了消除 <math>\epsilon</math> 产生式、单产生式和无用符号的算法，并通过编程实现了这些算法来消除文法中不必要的符号和产生式，得到了与原文法等价的新文法。此外，在输出新文法时，我们还需要按照规定的格式进行排序和格式化处理。本次实验使我们更好地理解上下文无关文法的应用和优化方法，并提高了自己的编程能力。</p>