## 上下文无关文法与下推自动机实验报告

# 1. 小组信息与分工

小组成员：

孙明皓2023210887：实现CFG的ε产生式、单产生式和无用符号消除算法

郎波 2023211048：负责PDA转换为CFG的算法实现

倪旭东2023211046：负责程序测试、调试和报告撰写

# 2. 实验环境

操作系统：Windows 10/11

开发语言：Python 3.10+

依赖库：python标准库（collections, itertools）

# 3. 程序设计思路与核心算法

本程序分为两个主要模块：

**1. PDA到CFG转换 (pda\_transform.py)**

* **设计思路**:  
  实现下推自动机(PDA)到上下文无关文法(CFG)的转换算法。核心是模拟PDA的栈操作过程，将状态转移映射为CFG的产生式规则。
* **核心算法**:
  + **状态三元组映射**: 将PDA的状态组合 **[q, z, gamma]** 映射为单字母非终结符(A-R)，状态过多时映射为An(A1, A2, …)的形式、
  + **转换规则处理**:
    - ε转移: 直接生成终结符产生式
    - 压栈操作: 使用笛卡尔积生成所有可能的状态序列
  + **CFG生成伪代码**：

for (q, a, z), transitions in self.delta.items():

for next\_q, push in transitions:

if not push: *# ε情况*

cfg.P[var].append([a] if a != 'ε' else ['ε'])

else: *# 压栈情况*

for states in product(self.Q, repeat=len(push)):

*# 构建产生式右侧*

rhs = [a] if a != 'ε' else []

for i in range(len(push)):

rhs.append(var\_i)

cfg.P[source].append(rhs)

**2. CFG简化 (cfg\_translate.py)**

* **设计思路**:  
  实现CFG简化三部曲：消除ε产生式、消除单产生式、消除无用符号。
* **核心算法**:
  1. **消除ε产生式**:
     + 计算可推导ε的非终结符集合
     + 生成新产生式（考虑符号可选性）
  2. **消除单产生式**:
     + 构建非终结符闭包
     + 替换单产生式为复合产生式
  3. **消除无用符号**:
     + 找出并迭代扩展可生成符号集，删除不在集合的产生式
     + 找出并迭代扩展可达符号集，删除不在集合的产生式

# 4. 输入输出格式

PDA输入示例：

Q={q0,q1}

T={a,b}

Z={z0,A}

δ(q0,a,z0)={(q0,Az0)}

δ(q0,a,A)={(q0,AA)}

δ(q0,b,A)={(q1,ε)}

δ(q1,b,A)={(q1,ε)}

δ(q1,ε,A)={(q1,ε)}

δ(q1,ε,z0)={(q1,ε)}

CFG输入示例：

N={A,B,C,D,S}

T={a,b,c,d,e}

P:

S->C|aA|aB

A->bB|cc

B->ε|d

C->c|ddC

D->e|eDe

S=S

输出格式：类似CFG输入格式，如

N={A,B,S,S1}

T={a,b}

P:

A->a|aA|aAS

B->BS|SB|SBS|a|aA|aAS|bb

S->AB|ASB

S1->AB|ASB|ε

S=S1

# 5. 程序测试用例与执行效果

**CFG简化算法：**

测试样例：

**N={S,A,B,C,D}**

**T={a,b,c,d}**

**P:**

**S->a|bA|B|ccD**

**A->abB|ε**

**B->aA**

**C->ddC**

**D->ddd**

**S=S**

测试结果（简化后CFG）：

**N={A,B,D,S}**

**T={a,b,c,d}**

**P:**

**A->abB**

**B->a|aA**

**D->ddd**

**S->a|aA|b|bA|ccD**

**S=S**

**PDA转换算法：**

测试样例：

**Q={q0,q1}**

**T={a,b}**

**Z={z0,B}**

**δ(q0,b,z0)={(q0,Bz0)}**

**δ(q0,b,B)={(q0,BB)}**

**δ(q0,a,B)={(q1,ε)}**

**δ(q1,a,B)={(q1,ε)}**

**δ(q1,ε,B)={(q1,ε)}**

**δ(q1,ε,z0)={(q1,ε)}**

测试结果：

简化前CFG：

**N={A,B,C,D,E,F,G,H,S}**

**T={a,b}**

**P:**

**A->bAA|bCE**

**B->bAB|bCF**

**C->a|bAC|bCG**

**D->bAD|bCH**

**G->a|ε**

**H->ε**

**S->B|D**

**S=S**

简化后CFG：

**N={C,G,S}**

**T={a,b}**

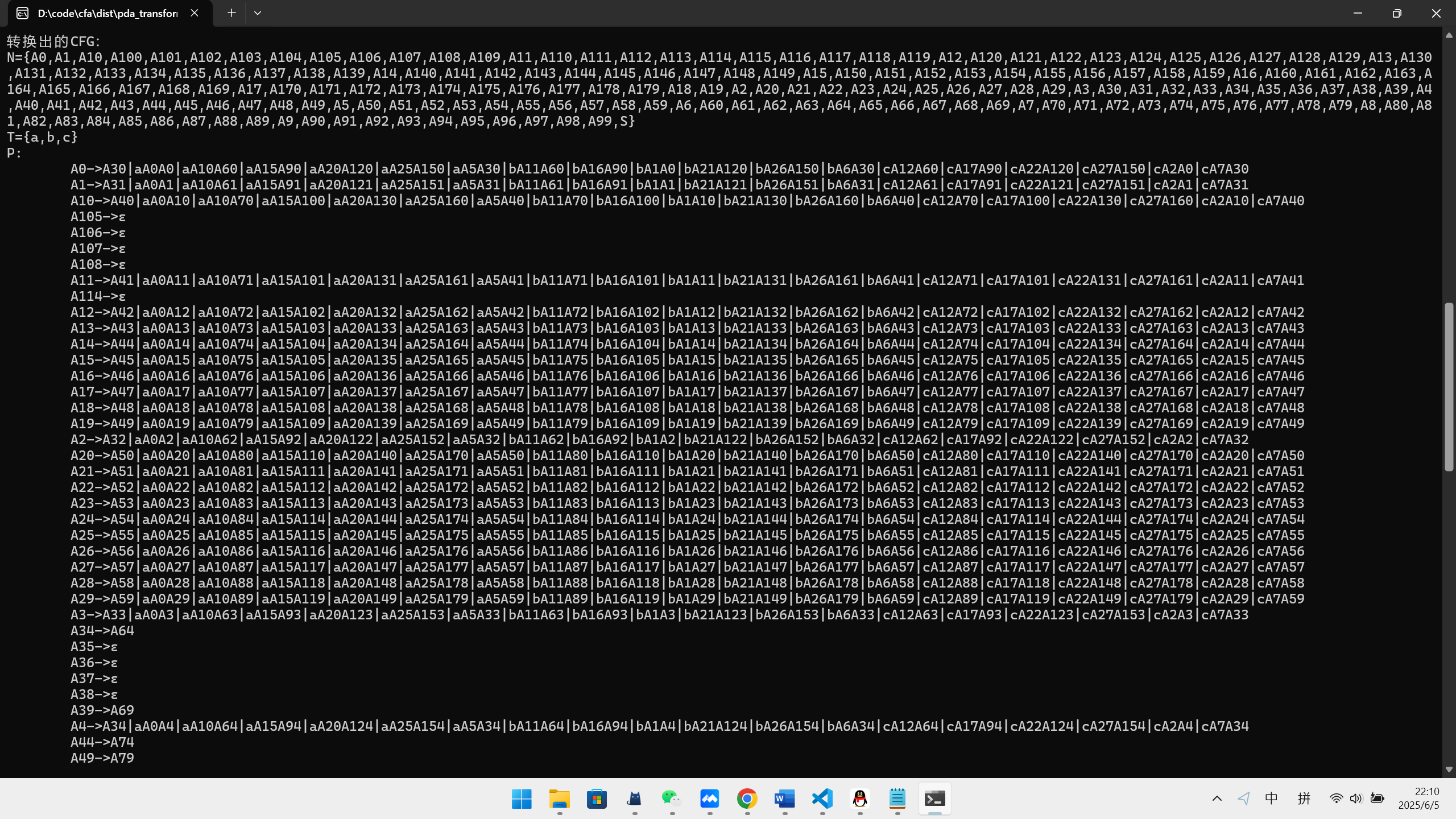
**P:**

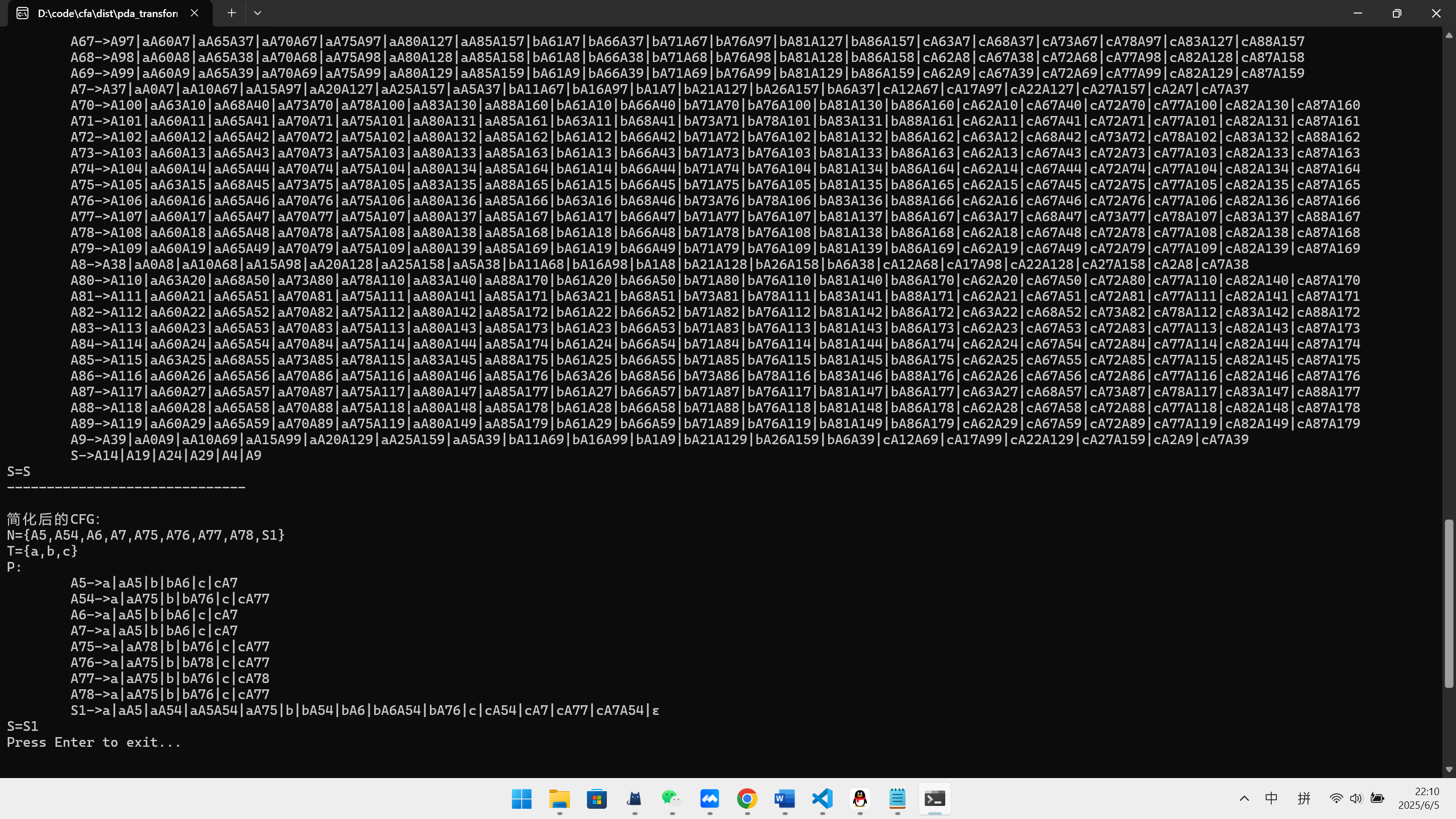
**C->a|bC|bCG**

**G->a**

**S->bC**

**S=S**

另，本程序还进行了复杂下推自动机输入的压力测试，见**测试用例.txt**最后一条，输出如图所示：



**6. 改进思路与建议**

1. **性能优化**:
   * 使用位运算加速ε产生式消除
   * 引入LRU缓存减少重复计算
2. **错误处理增强**:
   * 增加输入格式校验
   * 添加行号定位语法错误
3. **可视化扩展**:
   * 集成Graphviz生成PDA状态图
   * 输出CFG语法树图像
4. **功能扩展**:
   * 支持增广文法
   * 添加LL(1)/LR(1)分析表生成
5. **内存优化**:
   * 使用生成器避免大型笛卡尔积的完全展开