

实验 3: OBDD 生成

二元决策图（简称 BDD）是一个有向无环图，常用于表示布尔函数。每个 BDD 节点都与一个布尔变量关联。每个内部节点有两个输出边：**then** 边和 **else** 边。**then** 边表明布尔变量为“1”，而 **else** 边表明布尔变量为“0”。通过从根节点到叶节点对变量进行真/假决策。给定一个固定的 BDD 变量排序，我们能够为每一个不同的布尔函数生成一个独特的有序的 BDD（Ordered Binary Decision Diagram，简称 OBDD）。在这个项目中，你将处理一个单输出布尔函数，以字符串格式输入，并以 DOT 语言[1]输出你的 ROBDD。

程序接收的输入：使用标准的逻辑运算符（如 **and**, **or**, **not**）描述的布尔函数。表达式中的变量可以是任何小写字母。

示例：

```
a and c or b and c
```

【请注意，在本次实验中，只要求你的程序能处理 `input.txt` 中的特定的布尔表达式即可】

程序的输出：OBDD 的 DOT 格式表示，能够直接用图形工具如 Graphviz[2]进行可视化。

示例：

```
digraph BDD {
node0 [label="a"];
node1 [label="b"];
node2 [label="b"];
node3 [label="c"];
node4 [label="c"];
node5 [label="c"];
node6 [label="c"];
node7 [label="1"];
node8 [label="0"];
node9 [label="1"];
node10 [label="0"];
node11 [label="1"];
node12 [label="0"];
node13 [label="0"];
node14 [label="0"];
node0 -> node1 [label="1"];
node0 -> node2 [label="0"];
node1 -> node3 [label="1"];
node1 -> node4 [label="0"];
node2 -> node5 [label="1"];
node2 -> node6 [label="0"];
node3 -> node7 [label="1"];
node3 -> node8 [label="0"];
node4 -> node9 [label="1"];
node4 -> node10 [label="0"];
node5 -> node11 [label="1"];
node5 -> node12 [label="0"];
node6 -> node13 [label="1"];
node6 -> node14 [label="0"];
}
```

请根据以下规则提交你的报告和程序：

1. 上传你的报告你的报告，报告文件名应为你的学生 ID（例如，12345678.pdf），报告格式为 PDF 格式，字体大小为 12，报告中包含 OBDD 生成的代码实现、实现描述、`input.txt` 中特定的布尔表达式对应的 OBDD 的截图以及渲染该图的 DOT 语言的描述。
2. 上传一个压缩包（例如，B12345678.tgz），包括你的程序源代码、PLA 文件和一个 ReadMe 文件，该文件描述了您的程序如何在 window 系统中编译和执行你的程序。

[1] <https://graphviz.org/doc/info/lang.html>

[2] <https://dreampuf.github.io/GraphvizOnline>