《软件分析与验证》

基于控制流自动机的程序验证
谓词抽象

贺飞

清华大学软件学院

2024 年 6 月 7 日

上节课内容：限界模型检验

- 抽象格局、抽象执行、抽象可达图
- 精确抽象可达图
- 可达图和精确抽象可达图的构造
- 限界模型检验算法

目录

抽象

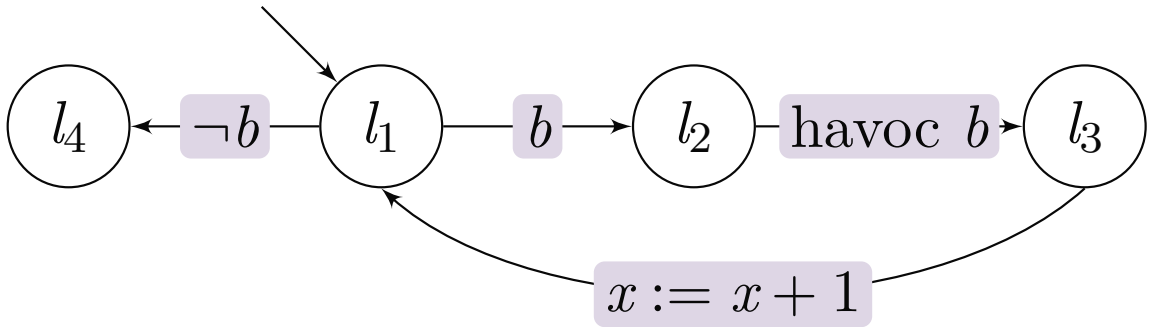
示例

下面的程序（其中 x 是整型变量，b 为布尔变量）是否满足给定的前置-后置条件对?

$\varphi_{pre} : x = 0$

$\varphi_{post} : x \neq -1$

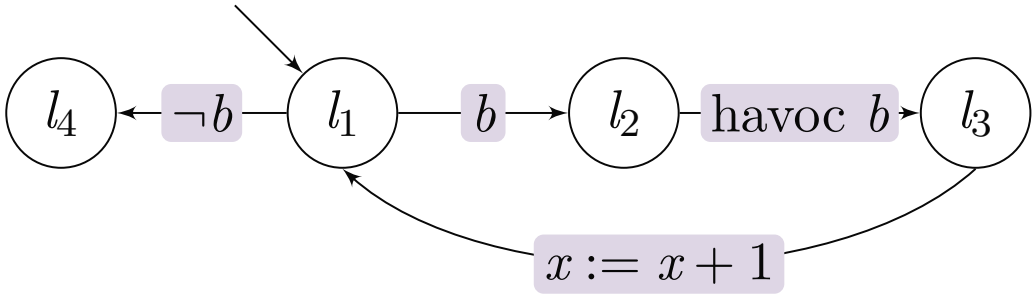
```
while(b){
  havoc b;
  x := x + 1;
}
```



如何验证?

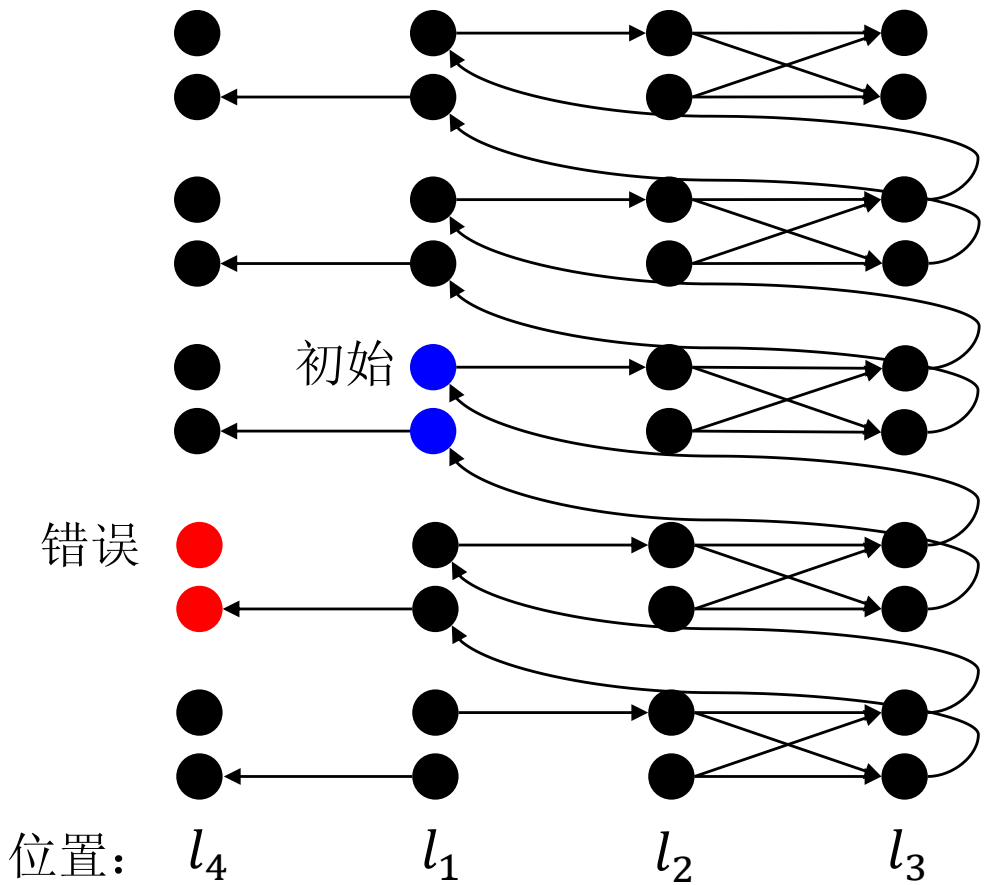
1. 该程序有无穷个格局（x 可以取 0 到无穷大），也有无穷个可达格局，其可达图无法绘制。
2. 该程序的精确抽象可达图也含无穷个格局，所以限界模型检验方法也无法验证该程序的正确性。

观察： x 被初始化为 0，其值只会增大，不会减少。因此 x 永远不等于 -1, 程序满足规约。



$\varphi_{pre} : x = 0$

$\varphi_{post} : x \neq -1$



$\{b \mapsto \textbf{true}, x \mapsto 2\}$
$\{b \mapsto \textbf{false}, x \mapsto 2\}$

$\{b \mapsto \textbf{true}, x \mapsto 1\}$
$\{b \mapsto \textbf{false}, x \mapsto 1\}$

$\{b \mapsto \textbf{true}, x \mapsto 0\}$
$\{b \mapsto \textbf{false}, x \mapsto 0\}$

$\{b \mapsto \textbf{true}, x \mapsto -1\}$
$\{b \mapsto \textbf{false}, x \mapsto -1\}$

$\{b \mapsto \textbf{true}, x \mapsto -2\}$
$\{b \mapsto \textbf{false}, x \mapsto -2\}$

程序状态

等价类划分

将程序格局集合划分成有限个等价类：

- 初始等价类：包含初始格局的等价类
- 错误等价类：包含错误格局的等价类

定义等价类之间的抽象变迁关系 T^{α}：

- 给定两个等价类 C_1, C_2，如果存在格局 $c_1 \in C_1$ 可以变迁到格局 $c_2 \in C_2$，即 $(c_1, c_2) \in T$，则令 $(C_1, C_2) \in T^{\alpha}$。
- 由抽象变迁关系确定的执行称抽象执行。

引理

如果不存在从初始等价类到错误等价类的抽象执行，则必不存在从初始格局到错误格局的具体执行。

注意：上述引理的反命题不成立：如果存在一条从初始等价类到错误等价类的抽象执行，可能不存在与之对应的具体执行，此时称该抽象执行为伪执行（spurious execution）。

抽象是程序验证的核心问题

笼统的讲，抽象就是建立从具体格局空间到抽象格局空间的映射：

- 具体格局空间一般都比较大，甚至包含无穷个格局
- 抽象格局空间则相对小得多，通常都是有限的

等价类划分是一种特殊的抽象：

- 不考虑程序位置，只对程序状态空间进行划分
- **优点：** 能够借助控制流自动机理论进行分析和验证
- **缺点：** 抽象受到限制，不一定能够找到最合适的抽象

回顾上一章中抽象格局的定义（一个有序对 (l, φ)，其中 l 是位置，φ 是逻辑公式）：

- 每一个等价类可以用刻画该等价类的逻辑公式来表示

基于抽象可达图的验证算法

定理

给定程序 P 和规约 $(\varphi_{pre}, \varphi_{post})$，如果程序存在一个不含错误抽象格局的抽象可达图，则程序满足规约。

不含错误抽象格局的抽象可达图也称为程序的正确性论据（safety proof）。

注意：程序的抽象可达图不唯一，我们需要找出可以作为正确性论据的抽象可达图。

示例

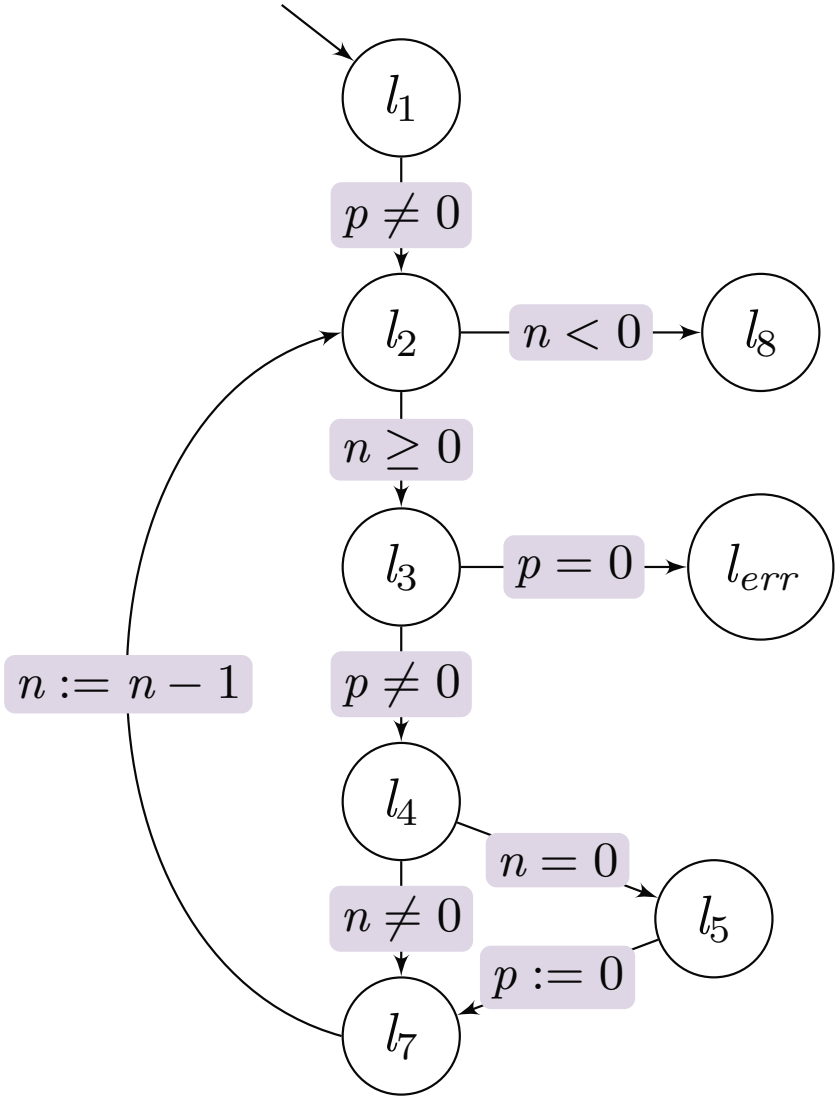
P_{goanna} 程序代码：

```
1 assume p != 0;
2 while (n >= 0){
3   assert p != 0;
4   if(n == 0){
5     p := 0;
6   }
7   n := n - 1;
8 }
```

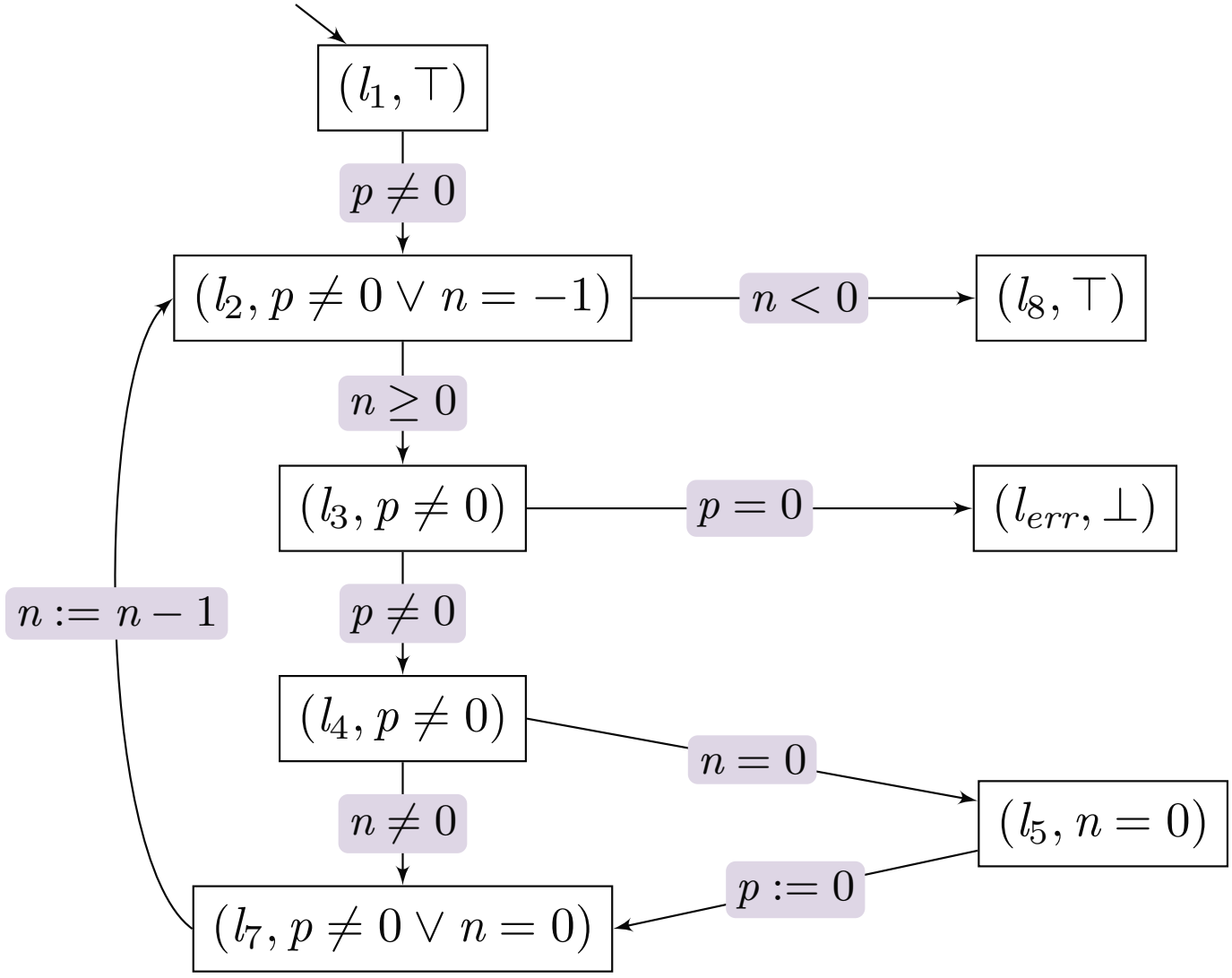
assert p 被转换为

```
if(!p){goto ERR}
```

控制流自动机：



示例：一个正确性证据


$(l_1, \top)$
$p \neq 0$
$(l_2, p \neq 0 \vee n = -1)$
$n < 0$
$(l_8, \top)$
$n \geq 0$
$(l_3, p \neq 0)$
$p = 0$
$(l_{err}, \bot)$
$n := n - 1$
$p \neq 0$
$(l_4, p \neq 0)$
$n = 0$
$(l_5, n = 0)$
$n \neq 0$
$p := 0$
$(l_7, p \neq 0 \vee n = 0)$

如何构造程序的正确性论据？

如何构造程序的正确性证据?

- 程序抽象可达图的构造过程可能不终止!
- $sp(\varphi, st) \Rightarrow \varphi'$ 是关键，如何弱化 $sp(\varphi, st)$ 得到合适的 φ'?
- 前面给出的抽象可达图与程序控制流自动机正好同构。
- 实际情况中，同一位置（例如，循环头）可能被多次经过，得到多个可达抽象格局。

构造正确性证据的关键是确定每个抽象格局的逻辑公式:

- 一般意义下，这是一个不可判定问题
- 能否通过限制逻辑公式的形式，从而获得一个可终止的算法?

谓词抽象

抽象最强后置条件

为获得一个可终止的抽象可达图构造过程，我们限制抽象格局逻辑公式只能由某个事先确定的谓词集合 B 中的谓词构成。

定义

φ 关于语句 st 和谓词集合 B 的抽象最强后置条件（abstract strongest postcondition）是：

$$sp_B^{\#}(\varphi, st) = \bigwedge \{p \in B \mid sp(\varphi, st) \Rightarrow p\}$$

对于 $B = \emptyset$ 的特殊情况，定义 $sp_B^{\#}(\varphi, st) = \bigwedge\{\} = \top$。

定义

对于程序的一个抽象可达图，如果对任意的 $((l, \varphi), st, (l', \varphi')) \in T^{\alpha}$，都满足：

$$sp_B^{\#}(\varphi, st) \Leftrightarrow \varphi'$$

则称其为程序关于谓词集合 B 的精确抽象可达图。

给定控制流自动机 $G = (Loc, \Delta, l_{in}, l_{ex})$ 和前置条件 φ_{pre}

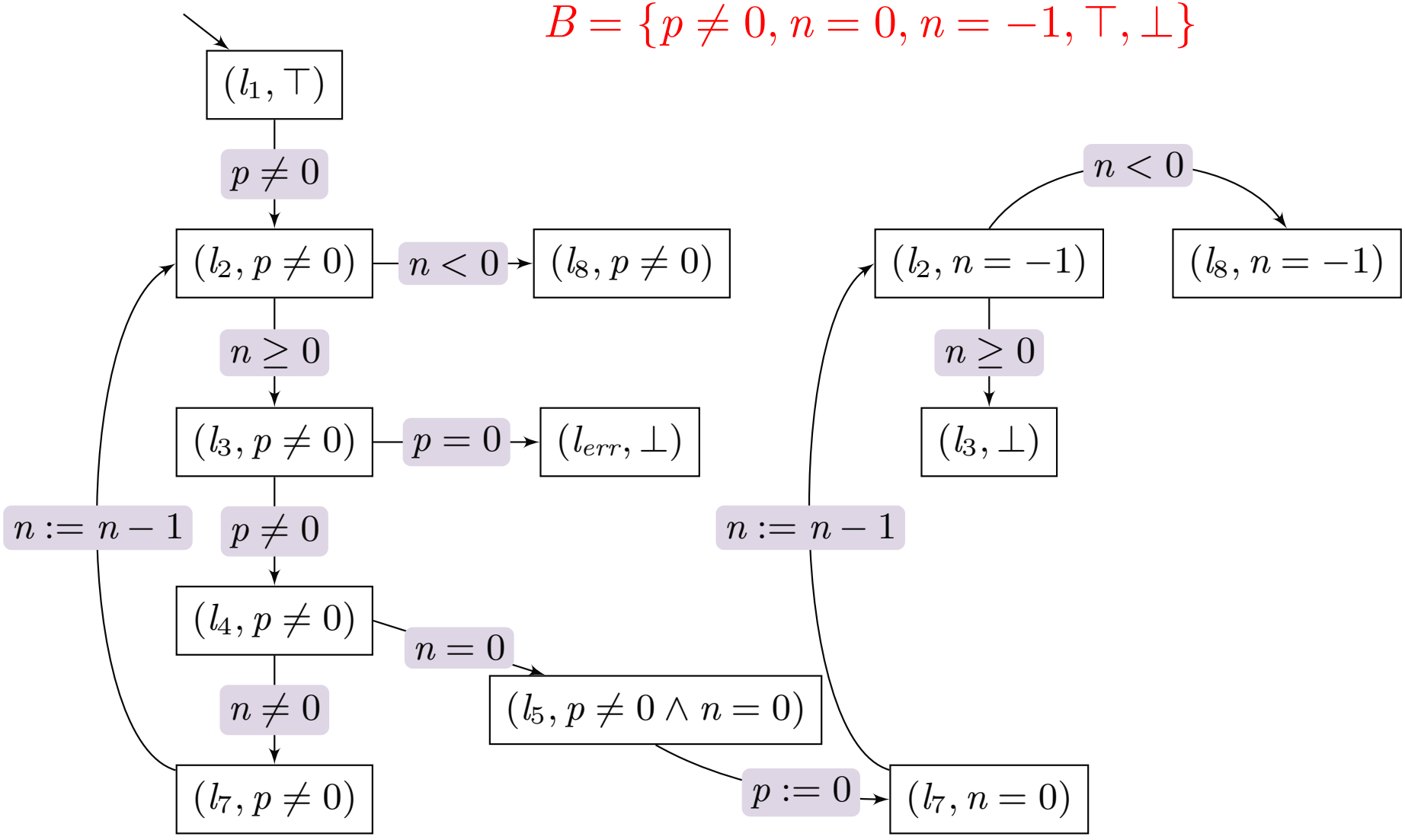
算法 $ConstructARGB(G, \varphi_{pre})$

$C^\alpha \leftarrow \{(l_{in}, \varphi_{pre})\},\ T^\alpha \leftarrow \emptyset,\ wl \leftarrow \{(l_{in}, \varphi_{pre})\}$
while $wl \neq \emptyset$ **do**
  $(l, \varphi) \leftarrow \textsc{RemoveFirst}(wl)$
  for all $(l, st, l') \in \Delta$ **do**
    $\varphi' \leftarrow sp_B^\#(\varphi, st)$
    $T^\alpha \leftarrow T^\alpha \cup \{((l, \varphi), st, (l', \varphi'))\}$
    if $(l', \varphi') \notin C^\alpha$ **then**
      $C^\alpha \leftarrow C^\alpha \cup (l', \varphi')$
      if $\varphi' \neq \bot$ **then**
        $wl \leftarrow wl \cup \{(l', \varphi')\}$
return (C^α, T^α)

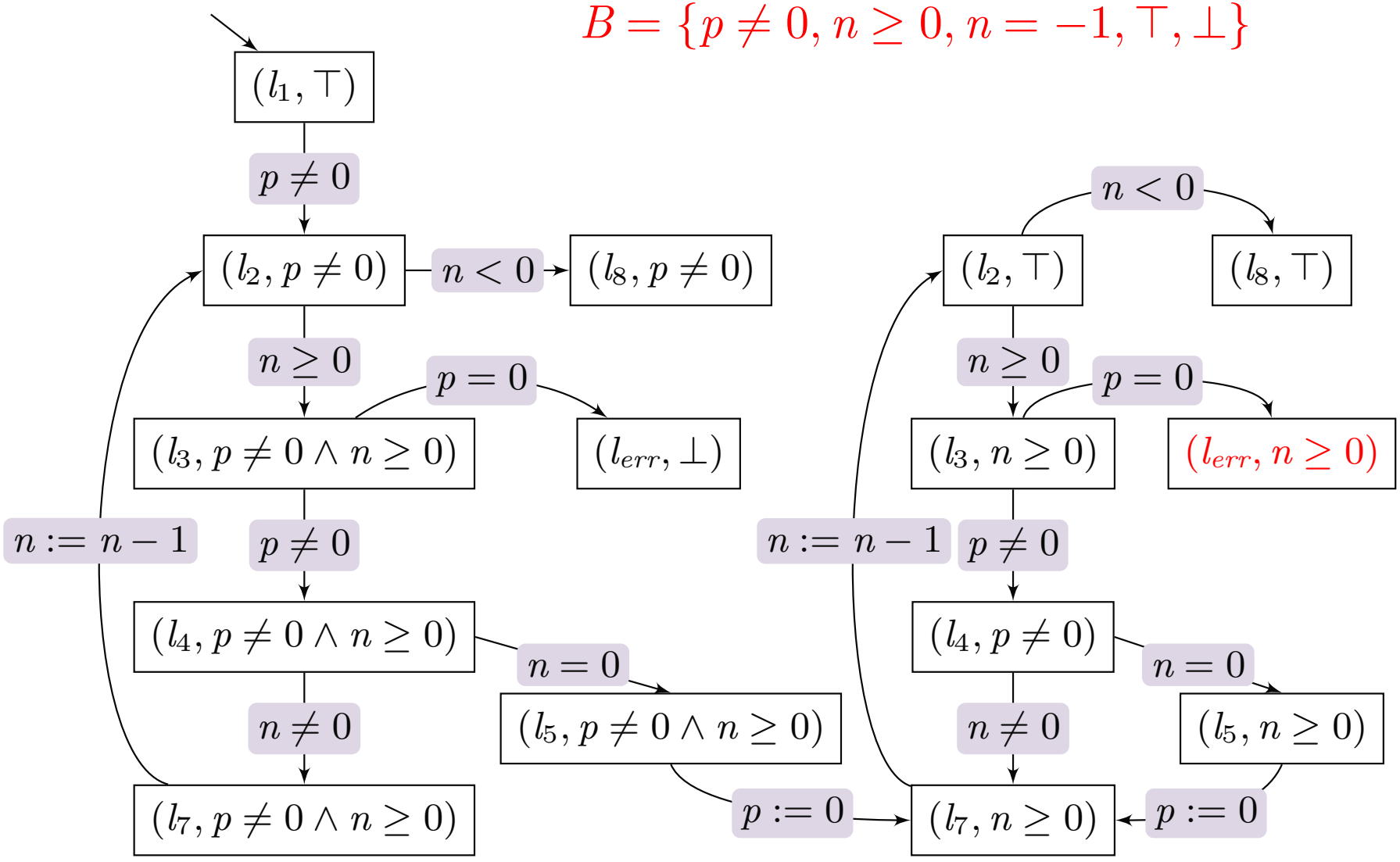
如果 B 是一个有限集合，该算法一定终止；又称为谓词抽象算法

示例程序 P_{goanna} 关于 B 的精确抽象可达图

$B = \{p \neq 0, n = 0, n = -1, \top, \bot\}$



示例程序 P_{goanna} 关于 B 的精确抽象可达图



谓词集合 B

构造程序关于 B 的精确抽象可达图，观察错误抽象格局 $(l_{err}, -)$：

- 如果上面的逻辑公式都是 $\bot$，说明不存在从初始抽象格局到错误抽象格局的抽象执行。
- 否则，存在一条从初始抽象格局到错误抽象格局的抽象执行（但该执行可能是伪执行）。

在不同的谓词集合下构造的精确抽象可达图可能是程序的一个正确性论据，也有可能会产生一条伪错误执行。

如何确定合适的谓词集合

1. 初始化 $B = \emptyset$
2. 构造程序关于 B 的精确抽象可达图 G
   - 如果 G 是程序正确性论据，终止算法并报告“程序满足规约，G 是正确性论据”；
   - 否则，返回 G 中从初始抽象格局到错误抽象格局的一条抽象执行 π。
3. 检查 π 是否对应程序的具体执行
   - 如果是，终止算法并报告“程序不满足规约，π 是反例”
   - 否则，π 是一条伪执行。
4. 分析 π 是伪执行的原因，得到新的谓词，插入到 B 中。
5. 重复上面的步骤 2 至步骤 4。

总结：基于谓词抽象的程序验证

- 程序抽象
- 正确性论据
- 基于抽象可达图的验证算法
- 谓词抽象

下节课内容

- 前沿性内容

谢谢!