《软件分析与验证》

最强后置条件

贺飞

清华大学软件学院

2024 年 5 月 31 日

目录

示例

下面的程序（其中 x, y, z 为整型变量）是否满足给定的前置-后置条件对?

```
z = y + x;
assume x <= 25;
x = z * z - 2;
assume x < 0;
havoc x;
assume z * (y % 23) < -20;
```

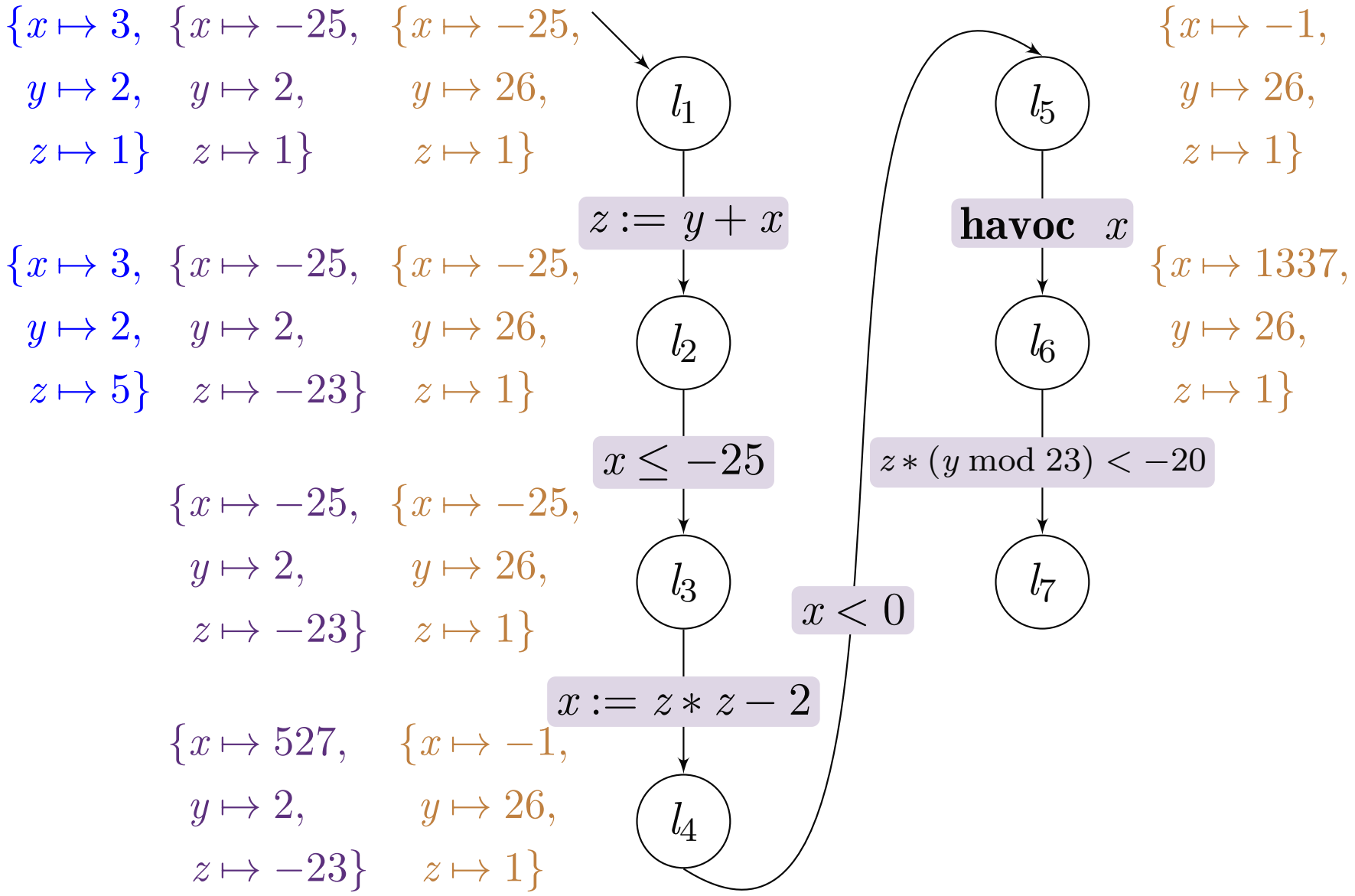
$$\varphi_{\text{pre}}: \ y \geq 1$$

$$\varphi_{\text{post}}: \ y \geq 45$$

根据基于执行的证明方法，任何一条从初始格局到错误格局的执行都可以作为该程序不满足规约的证据。

需要注意的是，此程序有不止一个初始格局。

示例


$\{x \mapsto 3, y \mapsto 2, z \mapsto 1\}$
$\{x \mapsto -25, y \mapsto 2, z \mapsto 1\}$
$\{x \mapsto -25, y \mapsto 26, z \mapsto 1\}$
l_1
$z := y + x$
$\{x \mapsto 3, y \mapsto 2, z \mapsto 5\}$
$\{x \mapsto -25, y \mapsto 2, z \mapsto -23\}$
$\{x \mapsto -25, y \mapsto 26, z \mapsto 1\}$
l_2
$x \leq -25$
$\{x \mapsto -25, y \mapsto 2, z \mapsto -23\}$
$\{x \mapsto -25, y \mapsto 26, z \mapsto 1\}$
l_3
$x := z * z - 2$
$\{x \mapsto 527, y \mapsto 2, z \mapsto -23\}$
$\{x \mapsto -1, y \mapsto 26, z \mapsto 1\}$
l_4
$x < 0$
l_5
havoc x
$\{x \mapsto -1, y \mapsto 26, z \mapsto 1\}$
$\{x \mapsto 1337, y \mapsto 26, z \mapsto 1\}$
l_6
$z * (y \bmod 23) < -20$
l_7

示例

在上面的方法中，每次只计算程序的一个格局

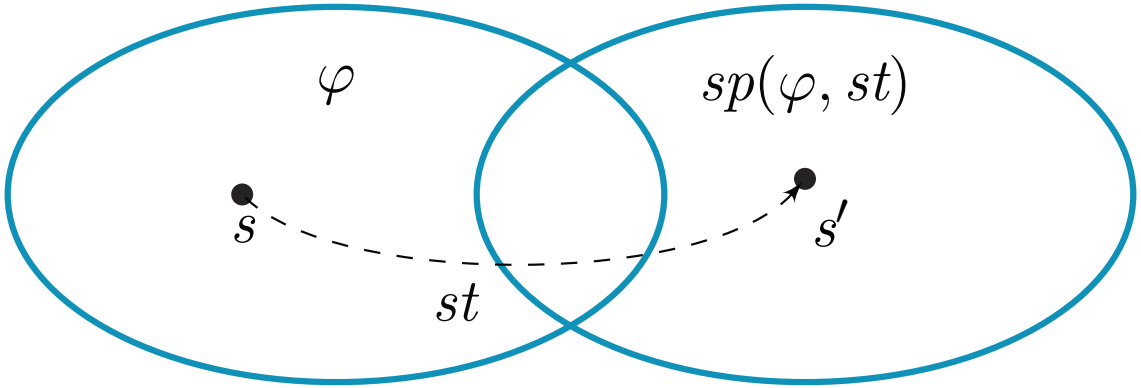
- 对于每一条执行，都需要从初始格局出发，依次计算经过每一条语句得到的可达格局。
- 为了找到一条出错的执行，在最坏情况下我们可能需要考虑程序的所有执行，计算出程序所有可达格局的集合。
- 这显然是一种极其低效的方法，并且在程序包含无数条执行的情况下，可能不终止。

能否一次计算程序的多个格局?

- 以逻辑公式代表在特定位置上的程序状态集合
- 计算经过每一条语句后的所有可达格局集合

最强后置条件定义

最强后置条件



给定一阶逻辑公式 φ 和程序语句 st，最强后置条件 $sp(\varphi, st)$ 是满足以下条件的一阶逻辑公式：

- $\forall s \models \varphi$，从 s 出发执行语句 st 终止，则执行后到达的状态 s' 满足 $sp(\varphi, st)$；
- $\forall s \models sp(\varphi, st)$，一定存在一个满足 φ 的前状态 s_0，使得从 s_0 出发执行语句 st 得到 s。

最强后置条件

定义 (复习：后像)

设 R 是定义在集合 X 上的一个二元关系，$Y \subseteq X$ 是 X 的一个子集，Y 关于 R 的后像（post-image）定义为：

$$post(Y, R) ::= \{x \in X \mid \text{存在 } y \in Y \text{ 使得 } (y, x) \in R\}$$

定义

给定一阶逻辑公式 φ 和程序语句 st，最强后置条件 $sp(\varphi, st)$ 是程序状态集合 $\{\varphi\}$ 关于二元关系 $[\![st]\!]$ 的后像，即：

$$\{sp(\varphi, st)\} = post(\{\varphi\}, [\![st]\!])$$

注意：

- 从满足 φ 的状态出发执行 st，如果执行不终止，没有后像；只有在执行终止时，才有后像，并且后状态一定满足 $sp(\varphi, st)$。
- 如果 st 对所有满足 φ 的状态都不终止，即 $post(\{\varphi\}, [\![st]\!]) = \emptyset$，则对应的最强后置条件也为空。

定义 (语义蕴涵)

给定两个一阶逻辑公式 φ 和 ψ，如果对任意解释 $\mathcal{M}$ 和任意赋值 ρ，只要 $[\![\varphi]\!]_{\mathcal{M},\rho}$ 为真，$[\![\psi]\!]_{\mathcal{M},\rho}$ 就必为真，就称 φ 语义蕴涵（*implies*）ψ，或称 ψ 是 φ 的有效推论（*consequence*），记为 $\varphi \Rightarrow \psi$。

定理

$\varphi \Rightarrow \psi$ 当且仅当 $\{\varphi\} \subseteq \{\psi\}$，即

$$\{s \mid s \models \varphi\} \subseteq \{s \mid s \models \psi\}$$

定义 (有效的霍尔三元组)

给定程序 st 及其前置条件 φ 和后置条件 ψ，如果

$$post(\{\varphi\}, [\![st]\!]) \subseteq \{\psi\}$$

成立，则称程序 st 满足规约 (φ, ψ)，记作 $st \models (\varphi, \psi)$；也称霍尔三元组 $\{\varphi\}\ st\ \{\psi\}$ 是有效式，记作 $\models \{\varphi\}\ st\ \{\psi\}$。

定理

程序 st 满足规约 (φ, ψ) 的充要条件是：$sp(\varphi, st) \Rightarrow \psi$

最强后置条件计算

定理 (空语句的最强后置条件)

$$sp(\varphi, \textbf{skip}) \Leftrightarrow \varphi$$

定理 (赋值语句的最强后置条件)

$$sp(\varphi, x := e) \Leftrightarrow \exists x_0.\ x = e[x \mapsto x_0] \wedge \varphi[x \mapsto x_0]$$

注意：$sp(\varphi, x := e)$ 在执行语句之后的后状态上考查，设前状态上 x 的值为 x_0，则显然 $x = e[x \mapsto x_0]$ 和 $\varphi[x \mapsto x_0]$ 都成立。

定理 (havoc 语句的最强后置条件)

$$sp(\varphi, \textbf{havoc}\ x) \Leftrightarrow \exists x.\ \varphi$$

定理

如果项 t 中不含变量 x，则下面两个公式语义等价：

$$(\exists x.\ \varphi \wedge x = t) \Leftrightarrow \varphi[x \mapsto t]$$

例

$$\begin{aligned} & \exists x_0.\ x_0 \bmod 2 = 0 \wedge x = x_0 + 1 \\ \Leftrightarrow\ & \exists x_0.\ x_0 \bmod 2 = 0 \wedge x_0 = x - 1 \\ \Leftrightarrow\ & (x - 1) \bmod 2 = 0 \end{aligned}$$

例

令 $x, \hat{x}$ 为两个整型变量

$$
\begin{aligned}
& \exists \hat{x}.\ a[\hat{x}] = 23 \land x = 2 \cdot \hat{x} \\
\Leftrightarrow\ & \exists \hat{x}.\ a[\hat{x}] = 23 \land \hat{x} = x \operatorname{div} 2 \land x \bmod 2 = 0 \\
\Leftrightarrow\ & a[x \operatorname{div} 2] = 23 \land x \bmod 2 = 0
\end{aligned}
$$

例

令 $x, \hat{x}, y$ 为三个实数类型变量

$$
\begin{aligned}
& \exists \hat{x}.\ a[\hat{x}] = 23 \land x = y \cdot \hat{x} \\
\Leftrightarrow\ & \exists \hat{x}.\ a[\hat{x}] = 23 \land x = y \cdot \hat{x} \land y \neq 0 \\
& \quad \lor a[\hat{x}] = 23 \land x = y \cdot \hat{x} \land y = 0 \\
\Leftrightarrow\ & a\left[\frac{x}{y}\right] = 23 \land y \neq 0 \\
& \quad \lor (\exists \hat{x}.\ a[\hat{x}] = 23) \land x = 0 \land y = 0
\end{aligned}
$$

定理

如果项 t 中不含变量 x，则下列两个公式语义等价：

$$(\forall x.\, \varphi \vee x \neq t) \Leftrightarrow \varphi[x \mapsto t]$$

最强后置条件：假设语句

定理 (假设语句的最强后置条件)

$$sp(\varphi, \textbf{assume } p) \Leftrightarrow \varphi \wedge p$$

证明.

$$\begin{aligned}
&\{sp(\varphi, \textbf{assume } p)\} \\
&= post(\{\varphi\}, [\![\textbf{assume } p]\!]) \\
&= \{s' \mid \exists s \in \{\varphi\} \wedge (s, s') \in [\![\textbf{assume } p]\!]\} \\
&= \{s' \mid \exists s \in \{\varphi\} \wedge s = s' \wedge s' \in \{p\}\} \\
&= \{\varphi \wedge p\}
\end{aligned}$$

□

最强后置条件：顺序组合语句

定理 (顺序组合语句的最强后置条件)

$$sp(\varphi, st_1; st_2) \Leftrightarrow sp(sp(\varphi, st_1), st_2)$$

证明.

$$\begin{aligned}
&sp(\varphi, st_1; st_2) \\
&= post(\{\varphi\}, [\![st_1; st_2]\!]) \\
&= \{s'' \mid \exists s \in \{\varphi\} \wedge (s, s'') \in [\![st_1; st_2]\!]\} \\
&= \{s'' \mid \exists s \exists s'.\ s \in \{\varphi\} \wedge (s, s') \in [\![st_1]\!] \wedge (s', s'') \in [\![st_2]\!]\} \\
&= \{s'' \mid \exists s' \in sp(\{\varphi\}, st_1) \wedge (s', s'') \in [\![st_2]\!]\} \\
&= sp(sp(\varphi, st_1), st_2)
\end{aligned}$$

□

例 (计算 $sp(i \geq n,\ i := i + k)$)

$$
\begin{aligned}
& sp(i \geq n,\ i := i + k) \\
\Leftrightarrow\ & \exists i_0.\ i = i_0 + k \wedge i_0 \geq n \\
\Leftrightarrow\ & i - k \geq n
\end{aligned}
$$

例 (计算 $sp(i \geq n,\ \textbf{assume}\ \ k \geq 0; i := i + k)$)

$$
\begin{aligned}
& sp(i \geq n,\ \textbf{assume}\ \ k \geq 0; i := i + k) \\
\Leftrightarrow\ & sp(sp(i \geq n,\ \textbf{assume}\ \ k \geq 0),\ i := i + k) \\
\Leftrightarrow\ & sp(k \geq 0 \wedge i \geq n,\ i := i + k) \\
\Leftrightarrow\ & \exists i_0.\ i = i_0 + k \wedge k \geq 0 \wedge i_0 \geq n \\
\Leftrightarrow\ & k \geq 0 \wedge i - k \geq n
\end{aligned}
$$

最强后置条件：分支语句

定理 (分支语句的最强后置条件)

$$sp(\varphi, \textbf{if}\ (p)\ \{st_1\}\ \textbf{else}\ \{st_2\}) \Leftrightarrow sp(\varphi \wedge p, st_1) \vee sp(\varphi \wedge \neg p, st_2)$$

引理

$$sp(\varphi_1 \vee \varphi_2, st) = sp(\varphi_1, st) \vee sp(\varphi_2, st)$$

证明.

基于上面的引理，根据 $\varphi \Leftrightarrow (\varphi \wedge p) \vee (\varphi \wedge \neg p)$ 证明分支语句的最强后置条件定理。 □

最强后置条件：循环语句

注意：

$$\textbf{while } (p)\ \{st\} \equiv (\textbf{assume } p; st)^*; \quad \textbf{assume } \neg p$$

定理 (循环语句的最强后置条件)

$$sp(\varphi, \textbf{while } (p)\ \{st\}) \Leftrightarrow \bigvee_{i \in \mathbb{N}} sp\left(sp^i(\varphi, \textbf{assume } p; st),\ \textbf{assume } \neg p\right)$$

证明.

$$\begin{aligned}
\{sp(\varphi, (\textbf{assume } p; st)^*)\} &= \bigvee_{i \in \mathbb{N}} sp(\varphi, (\textbf{assume } p; st)^i) \\
&= \bigvee_{i \in \mathbb{N}} sp^i(\varphi, (\textbf{assume } p; st)) \\
\{sp(\varphi, \textbf{while } (p)\ \{st\})\} &= sp(sp(\varphi, (\textbf{assume } p; st)^*), \textbf{assume } \neg p) \\
&= \bigvee_{i \in \mathbb{N}} sp(sp^i(\varphi, \textbf{assume } p; st),\ \textbf{assume } \neg p)
\end{aligned}$$

循环语句的最强后置条件

循环语句的最强后置条件一般无法直接计算。但对于一些特殊循环，有可能利用前面的定理计算其最强后置条件。

例

$$
\begin{aligned}
& sp(i = 0,\ \textbf{while}\ (x)\{i := i + 1; \textbf{havoc}\ x\}) \\
\Leftrightarrow\ & \bigvee_{k \in \mathbb{N}} sp(sp^k(i = 0, \textbf{assume}\ x; i := i + 1; \textbf{havoc}\ x),\ \textbf{assume}\ \neg x) \\
\Leftrightarrow\ & sp(i = 0, \textbf{assume}\ \neg x) \vee sp(i = 1, \textbf{assume}\ \neg x) \vee \ldots \\
\Leftrightarrow\ & \neg x \wedge i \geq 0
\end{aligned}
$$

验证条件

$$\{\varphi\}\ st_1; \ldots; st_n\ \{\psi\}$$

- 用 wp 表示的验证条件：

$$\varphi \Rightarrow wp(st_1; \ldots; st_n,\ \psi)$$

- 用 sp 表示的验证条件：

$$sp(\varphi,\ st_1; \ldots; st_n) \Rightarrow \psi$$

示例

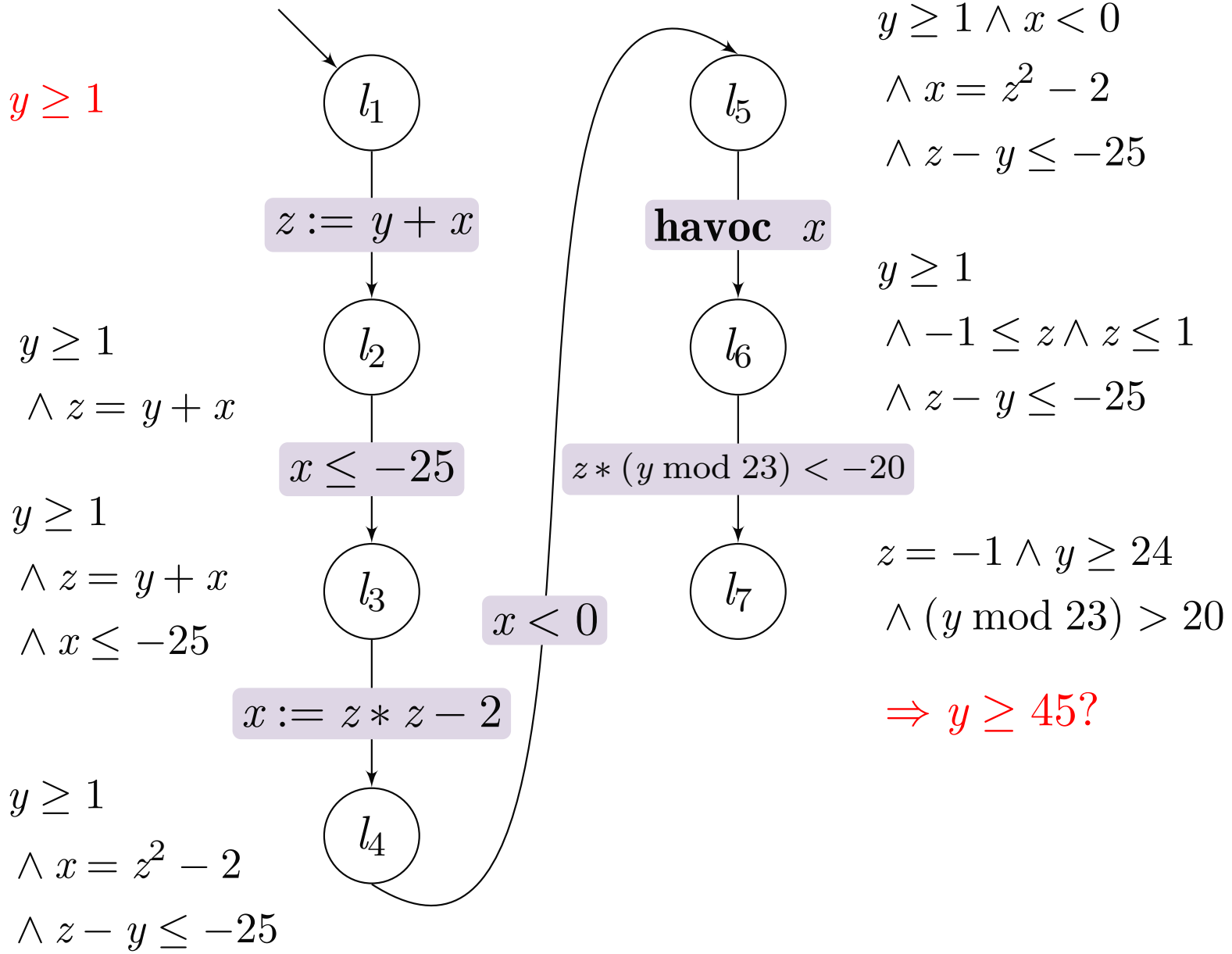
下面的程序（其中 x, y, z 为整型变量）是否满足给定的前置-后置条件对?

```
z = y + x;
assume x <= 25;
x = z * z - 2;
assume x < 0;
havoc x;
assume z * (y % 23) < -20;
```

$$\varphi_{\text{pre}}: \ y \geq 1$$

$$\varphi_{\text{post}}: \ y \geq 45$$

示例


$y \geq 1$
l_1
$z := y + x$
l_2
$y \geq 1$
$\wedge\, z = y + x$
$x \leq -25$
l_3
$y \geq 1$
$\wedge\, z = y + x$
$\wedge\, x \leq -25$
$x := z * z - 2$
l_4
$y \geq 1$
$\wedge\, x = z^2 - 2$
$\wedge\, z - y \leq -25$
$x < 0$
l_5
havoc x
l_6
$y \geq 1$
$\wedge\, -1 \leq z \wedge z \leq 1$
$\wedge\, z - y \leq -25$
$z * (y \bmod 23) < -20$
l_7
$z = -1 \wedge y \geq 24$
$\wedge\, (y \bmod 23) > 20$
$y \geq 1 \wedge x < 0$
$\wedge\, x = z^2 - 2$
$\wedge\, z - y \leq -25$
$\Rightarrow y \geq 45?$

总结：最强后置条件

- 最强后置条件定义
- 量词消去
- 最强后置条件计算
- 基于最强后置条件的程序验证

下节课内容

- 基于循环展开的自动程序验证
- 基于抽象的自动程序验证

谢谢!