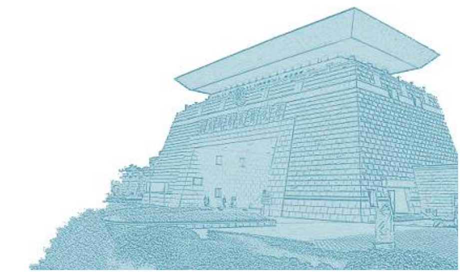
抽象解释
及其在静态分析中的应用

陈立前
国防科技大学



目录

抽象语义

- 从具体语义到抽象语义

具体语义

$$\mathcal{X} : L \rightarrow \mathcal{D} \text{ least solution of } \begin{cases} \mathcal{X}_e \text{ given} \\ \mathcal{X}_{\ell \neq e} = \bigcup_{(\ell', c, \ell) \in A} \mathrm{C}[\![c]\!] \, \mathcal{X}_{\ell'} \end{cases}$$

抽象语义

$$\color{blue}{\mathcal{X}^\sharp} \color{black}{: L \rightarrow \mathcal{D}^\sharp \text{ any solution of }} \begin{cases} \mathcal{X}^\sharp_e \text{ such that } \mathcal{X}_e \subseteq \gamma(\mathcal{X}^\sharp_e) \\ \underline{\mathcal{X}^\sharp_{\ell \neq e}} \supseteq^\sharp \bigcup^\sharp_{(\ell', c, \ell) \in A} \underline{\color{red}{\mathrm{C}^\sharp[\![c]\!]} \color{black}{\, \mathcal{X}^\sharp_{\ell'}}} \end{cases}$$

抽象状态：抽象域上的域元素

抽象操作：抽象域上的域操作

抽象语义

- 从具体语义到抽象语义

具体语义

$$\mathcal{X} : L \to \mathcal{D} \text{ least solution of } \begin{cases} \mathcal{X}_e \text{ given} \\ \mathcal{X}_{\ell \neq e} = \bigcup_{(\ell', c, \ell) \in A} \mathrm{C}[\![c]\!] \mathcal{X}_{\ell'} \end{cases}$$

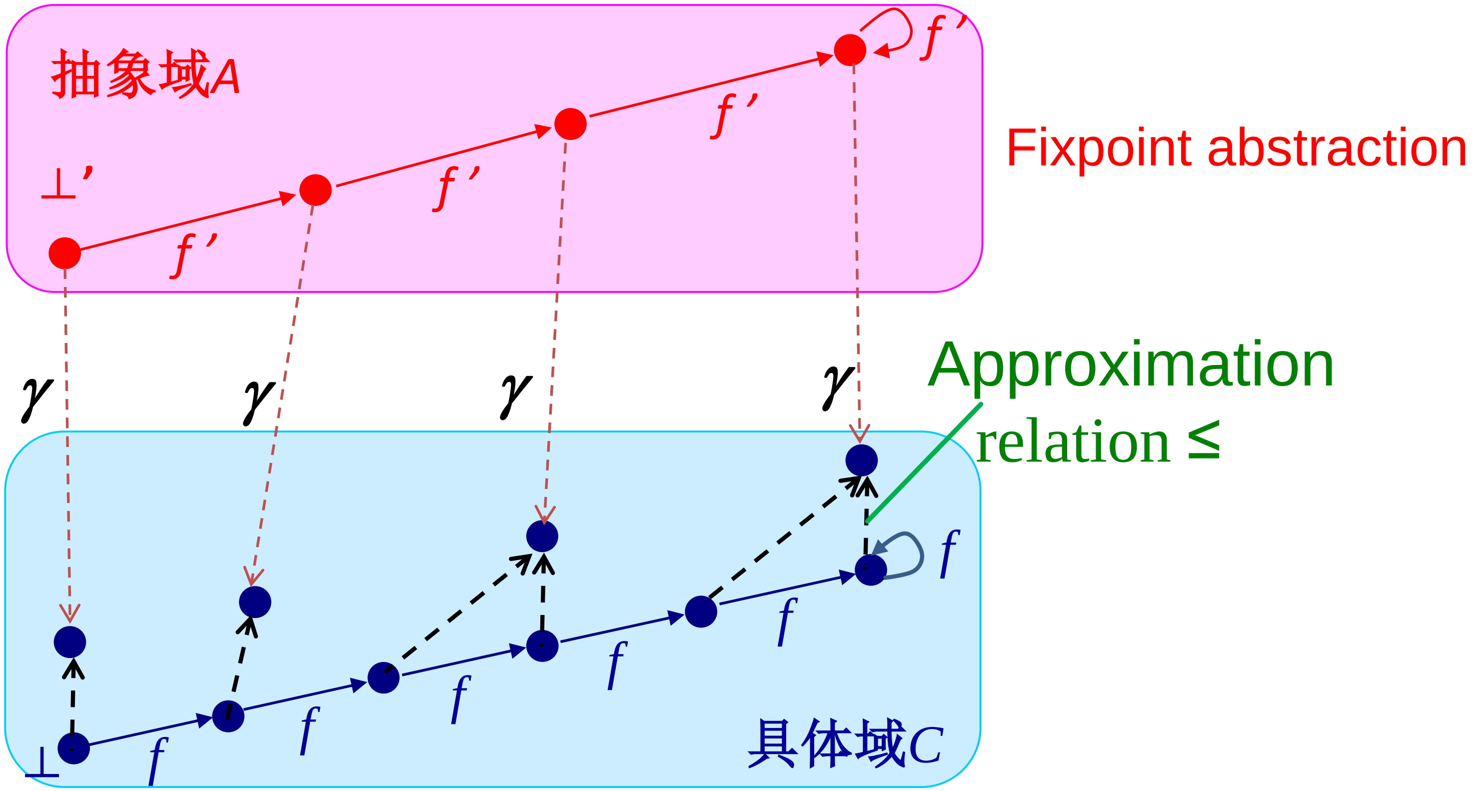
抽象语义

$$\mathcal{X}^\sharp : L \to \mathcal{D}^\sharp \text{ any solution of } \begin{cases} \mathcal{X}_e^\sharp \text{ such that } \mathcal{X}_e \subseteq \gamma(\mathcal{X}_e^\sharp) \\ \underline{\mathcal{X}_{\ell \neq e}^\sharp \supseteq^\sharp} \bigcup^\sharp_{(\ell', c, \ell) \in A} \underline{\mathrm{C}^\sharp[\![c]\!] \mathcal{X}_{\ell'}^\sharp} \end{cases}$$

$$\boxed{\text{可靠性保证 } \forall \ell \in L: \gamma(\mathcal{X}_\ell^\sharp) \supseteq \mathcal{X}_\ell}$$

抽象分析

- 本质：在抽象域*A*中计算函数 *f'* 的最小不动点



$$f \circ \gamma \le \gamma \circ f' \Rightarrow \mathrm{lfp}_{\perp} f \le \gamma(\mathrm{lfp}_{\perp'} f')$$

抽象分析

- 抽象域上的迭代策略

$$\mathcal{X}_e^{\sharp 0} \stackrel{\text{def}}{=} \mathcal{X}_e^{\sharp} \quad \text{such that } \mathcal{X}_e \subseteq \gamma(\mathcal{X}_e^{\sharp})$$

$$\mathcal{X}_{\ell \neq e}^{\sharp 0} \stackrel{\text{def}}{=} \perp^{\sharp}$$

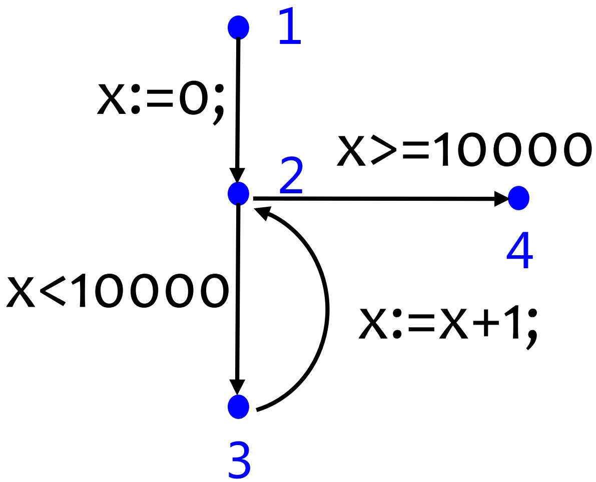
$$\mathcal{X}_\ell^{\sharp n+1} \stackrel{\text{def}}{=} \begin{cases} \mathcal{X}_e^{\sharp} & \text{if } \ell = e \\ \bigcup^{\sharp}_{(\ell', c, \ell) \in A} \mathsf{C}^{\sharp}[\![c]\!] \mathcal{X}_{\ell'}^{\sharp n} & \text{if } \ell \notin \mathcal{W}, \ell \neq e \\ \boxed{\mathcal{X}_\ell^{\sharp n} \triangledown \bigcup^{\sharp}_{(\ell', c, \ell) \in A} \mathsf{C}^{\sharp}[\![c]\!] \mathcal{X}_{\ell'}^{\sharp n} \quad \text{if } \ell \in \underline{\mathcal{W}}, \ell \neq e} \end{cases}$$

终止性保证

加宽点集合：一般是循环头

抽象分析—示例

- 基于区间域的抽象分析




```
1 x:=0;
2 while ( x<1000) do
3    x:=x+1;
done; 4
```

$X_1 = [-\infty,+\infty]$
$X_2 = (C[[x:=0]]X_1 \cup C[[x:=x+1]]X_3)$
$X_3 = C[[x<10000]]X_2$
$X_4 = C[[x>=10000]]X_2$

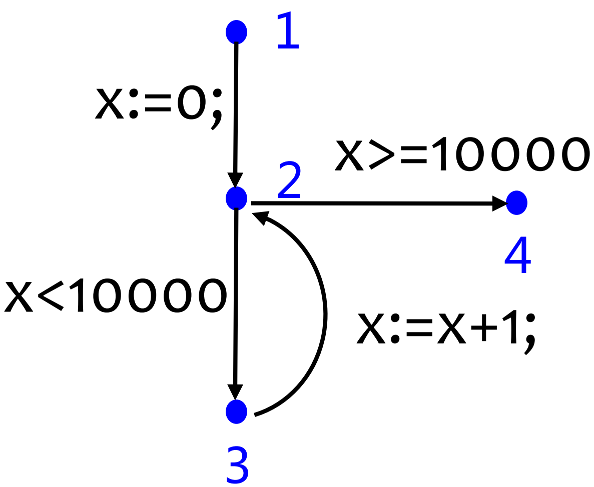
⇒

$X_1 = [-\infty,+\infty]$
$X_2 = [0,0] \cup (X_3+[1,1])$
$X_3 = [0,9999] \cap X_2$
$X_4 = [10000,+\infty] \cap X_2$

抽象分析—示例

- 基于区间域的抽象分析

$$X_1 = [-\infty,+\infty]$$
$$X_2 = [0,0] \cup (X_3+[1,1])$$
$$X_3 = [0,9999] \cap X_2$$
$$X_4 = [10000,+\infty] \cap X_2$$

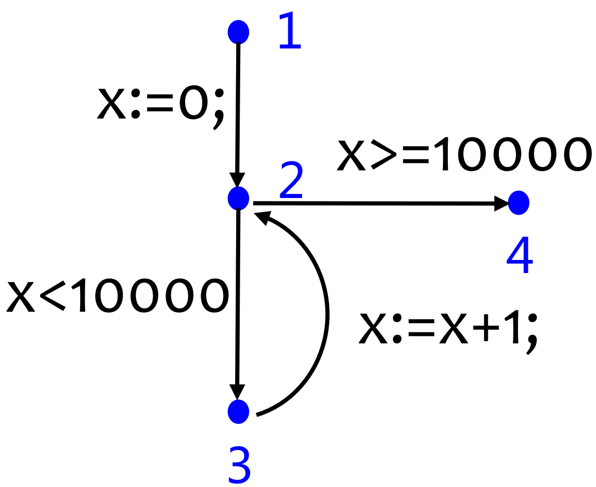


ℓ	$X_\ell^{\#0}$
1	$\top^{\#}$
2 ∇	$\bot^{\#}$
3	$\bot^{\#}$
4	$\bot^{\#}$

抽象分析—示例

- 基于区间域的抽象分析

$X_1 = [-\infty,+\infty]$
$X_2 = [0,0] \cup (X_3+[1,1])$
$X_3 = [0,9999] \cap X_2$
$X_4 = [10000,+\infty] \cap X_2$



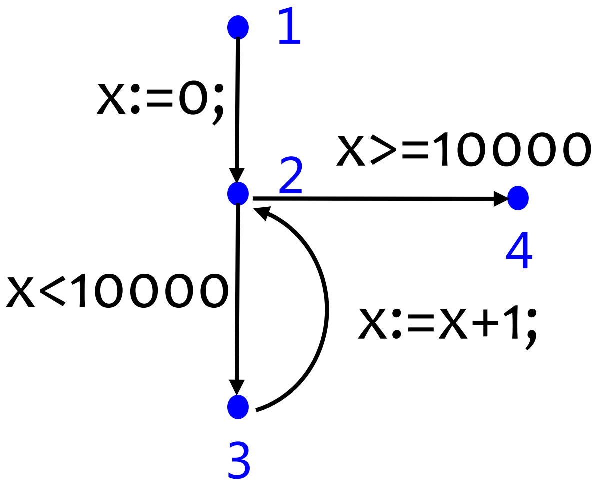
ℓ	$X_\ell^{\#0}$	$X_\ell^{\#1}$	
1	$\top^\#$	$\top^\#$	
2 ∇	$\bot^\#$	[0,0]	
3	$\bot^\#$	$\bot^\#$	
4	$\bot^\#$	$\bot^\#$	

$$X_2^{\#1} = \bot^\# \nabla ([0,0] \cup \bot^\#) = \bot^\# \nabla [0,0] = [0,0]$$

抽象分析—示例

- 基于区间域的抽象分析

$X_1 = [-\infty,+\infty]$
$X_2 = [0,0] \cup (X_3+[1,1])$
$X_3 = [0,9999] \cap X_2$
$X_4 = [10000,+\infty] \cap X_2$



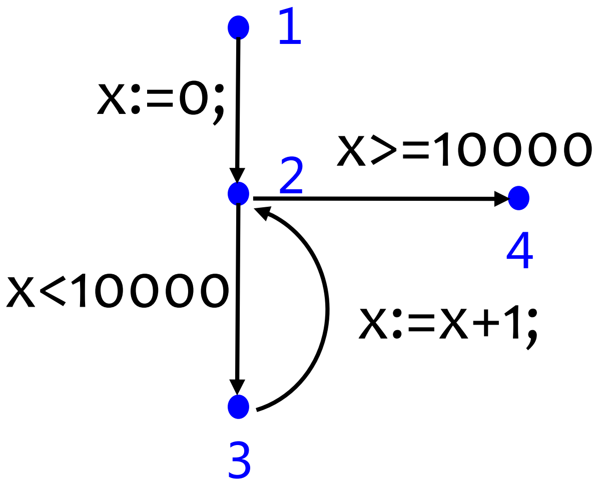
ℓ	$X_\ell^{\#0}$	$X_\ell^{\#1}$	$X_\ell^{\#2}$
1	$\top^\#$	$\top^\#$	$\top^\#$
2 ∇	$\bot^\#$	[0,0]	[0,0]
3	$\bot^\#$	$\bot^\#$	[0,0]
4	$\bot^\#$	$\bot^\#$	$\bot^\#$

$$X_2^{\#1} = \bot^\# \nabla ([0,0] \cup \bot^\#) = \bot^\# \nabla [0,0] = [0,0]$$
$$X_2^{\#2} = [0,0] \nabla ([0,0] \cup \bot^\#) = [0,0] \nabla [0,0] = [0,0]$$

抽象分析—示例

- 基于区间域的抽象分析

$$
\begin{aligned}
X_1 &= [-oo,+oo] \\
X_2 &= [0,0] \cup (X_3+[1,1]) \\
X_3 &= [0,9999] \cap X_2 \\
X_4 &= [10000,+oo] \cap X_2
\end{aligned}
$$



ℓ	$X_\ell^{\#0}$	$X_\ell^{\#1}$	$X_\ell^{\#2}$	$X_\ell^{\#3}$	
1	$\top^\#$	$\top^\#$	$\top^\#$	$\top^\#$[0,+	
2 ∇	$\bot^\#$	[0,0]	[0,0]	oo]	
3	$\bot^\#$	$\bot^\#$	[0,0]	[0,0]	
4	$\bot^\#$	$\bot^\#$	$\bot^\#$	$\bot^\#$	

$$X_2^{\#1} = \bot^\# \nabla ([0,0] \cup \bot^\#) = \bot^\# \nabla [0,0] = [0,0]$$

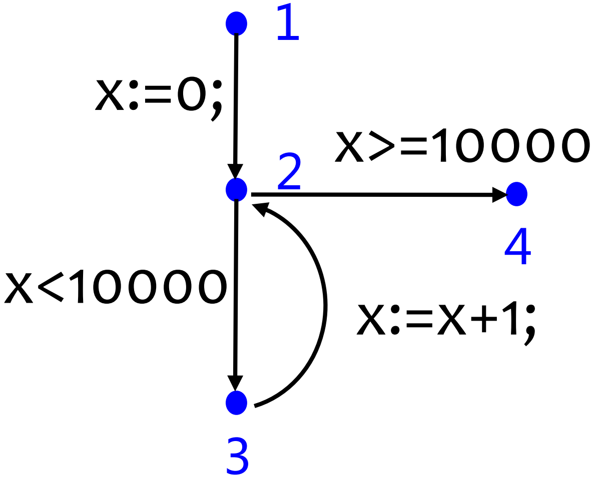
$$X_2^{\#2} = [0,0] \nabla ([0,0] \cup \bot^\#) = [0,0] \nabla [0,0] = [0,0]$$

$$X_2^{\#3} = [0,0] \nabla ([0,0] \cup [1,1]) = [0,0] \nabla [0,1] = [0,+oo]$$

抽象分析—示例

- 基于区间域的抽象分析

X1 = [-oo,+oo]
X2 = [0,0] ∪ (X3+[1,1])
X3 = [0,9999] ∩ X2
X4 = [10000,+oo] ∩ X2



ℓ	$\mathcal{X}_\ell^{\#0}$	$\mathcal{X}_\ell^{\#1}$	$\mathcal{X}_\ell^{\#2}$	$\mathcal{X}_\ell^{\#3}$	$\mathcal{X}_\ell^{\#4}$
1	$\top^\#$	$\top^\#$	$\top^\#$	$\top^\#$[0,+	$\top^\#$
2 ∇	$\bot^\#$	[0,0]	[0,0]	oo]	[0,+oo]
3	$\bot^\#$	$\bot^\#$	[0,0]	[0,0]	[0,9999]
4	$\bot^\#$	$\bot^\#$	$\bot^\#$	$\bot^\#$	[10000,+oo]

$$\mathcal{X}_2^{\#1} = \bot^\# \nabla ([0,0] \cup \bot^\#) = \bot^\# \nabla [0,0] = [0,0]$$

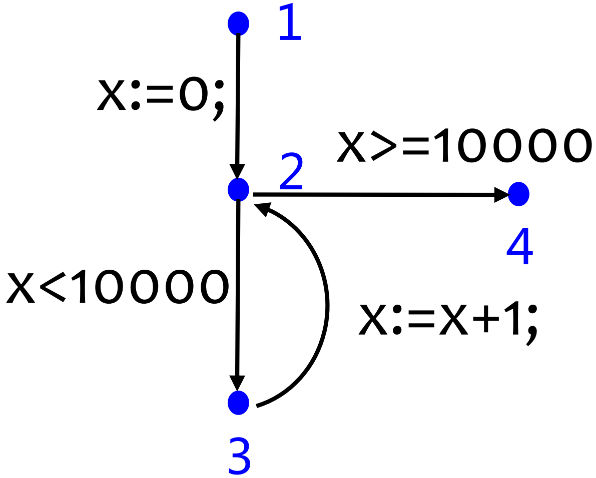
$$\mathcal{X}_2^{\#2} = [0,0] \nabla ([0,0] \cup \bot^\#) = [0,0] \nabla [0,0] = [0,0]$$

$$\mathcal{X}_2^{\#3} = [0,0] \nabla ([0,0] \cup [1,1]) = [0,0] \nabla [0,1] = [0,+oo]$$

$$\mathcal{X}_2^{\#4} = [0,+oo[\nabla ([0,0] \cup [1,10000]) = [0,+oo] \nabla [0,10000] = [0,+oo]$$

抽象分析—示例

- 基于区间域的抽象分析



ℓ	$X_\ell^{\#4}$
1	$\top^{\#}$
2 ∇	[0,+oo]
3	[0,9999]
4	[10000,+oo]

不变式信息

```
1 x:=0;
{x∈ [0,+oo]}
2 while ( x<10000) do
    {x∈ [0,9999]}
3     x:=x+1;
      {x∈ [1,10000]}
done; 4
{x∈ [10000,+oo]}
```

抽象分析的结果

- 抽象分析的结果：各程序点处可靠的不变式

具体语义	$\mathcal{X} : L \to \mathcal{D}$ least solution of $\begin{cases} \mathcal{X}_e \text{ given} \\ \mathcal{X}_{\ell \neq e} = \bigcup_{(\ell', c, \ell) \in A} \mathrm{C}[\![c]\!] \mathcal{X}_{\ell'} \end{cases}$
抽象语义	$\mathcal{X}^\sharp : L \to \mathcal{D}^\sharp$ any solution of $\begin{cases} \mathcal{X}_e^\sharp \text{ such that } \mathcal{X}_e \subseteq \gamma(\mathcal{X}_e^\sharp) \\ \mathcal{X}_{\ell \neq e}^\sharp \supseteq^\sharp \bigcup^\sharp_{(\ell', c, \ell) \in A} \mathrm{C}^\sharp[\![c]\!] \mathcal{X}_{\ell'}^\sharp \end{cases}$

$$\boxed{\text{可靠性保证 } \forall \ell \in L: \gamma(\mathcal{X}_\ell^\sharp) \supseteq \mathcal{X}_\ell}$$

程序点 $\ell \in L$ 处**可靠**的不变式

程序点 $\ell \in L$ 处**最精确**的不变式

抽象分析的结果

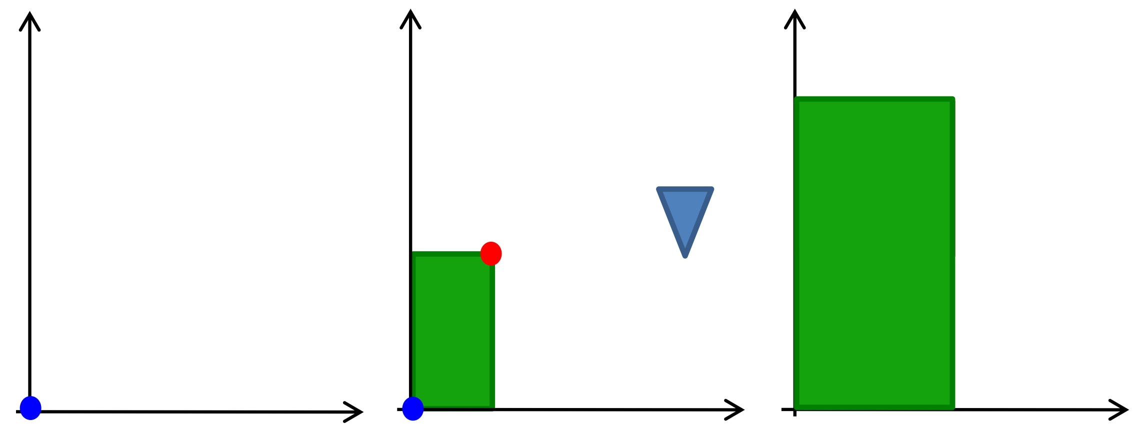
- 抽象分析的结果：各程序点处可靠的不变式

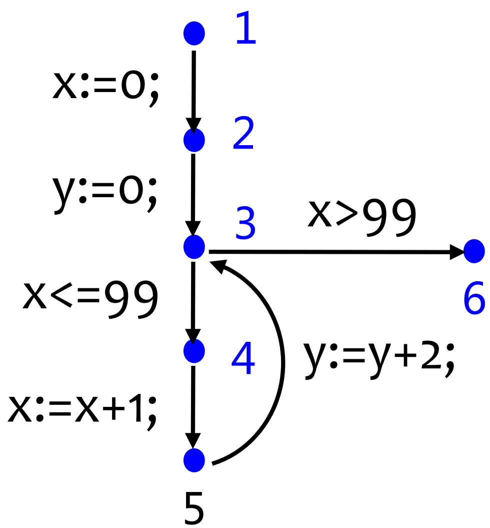
根据不变式，来报警
(检查程序中的错误)

抽象分析的应用—检查程序错误

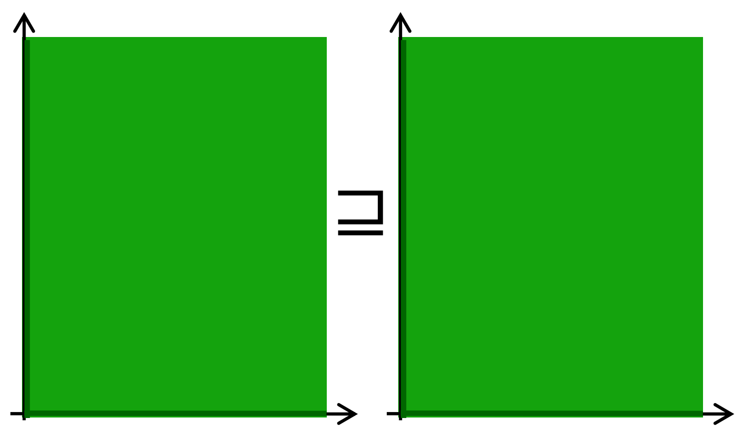
- 示例：检查数值相关错误（区间抽象）

```
1 x:=0;
2 y:=0;
3 while ( x<=99) do
4    x:=x+1;
5    y:=y+2;
done; 6
```





程序点3处



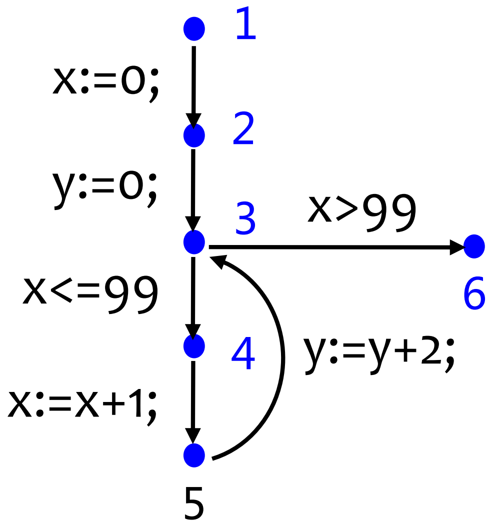
$x \in [0,+oo]$

$y \in [0,+oo]$

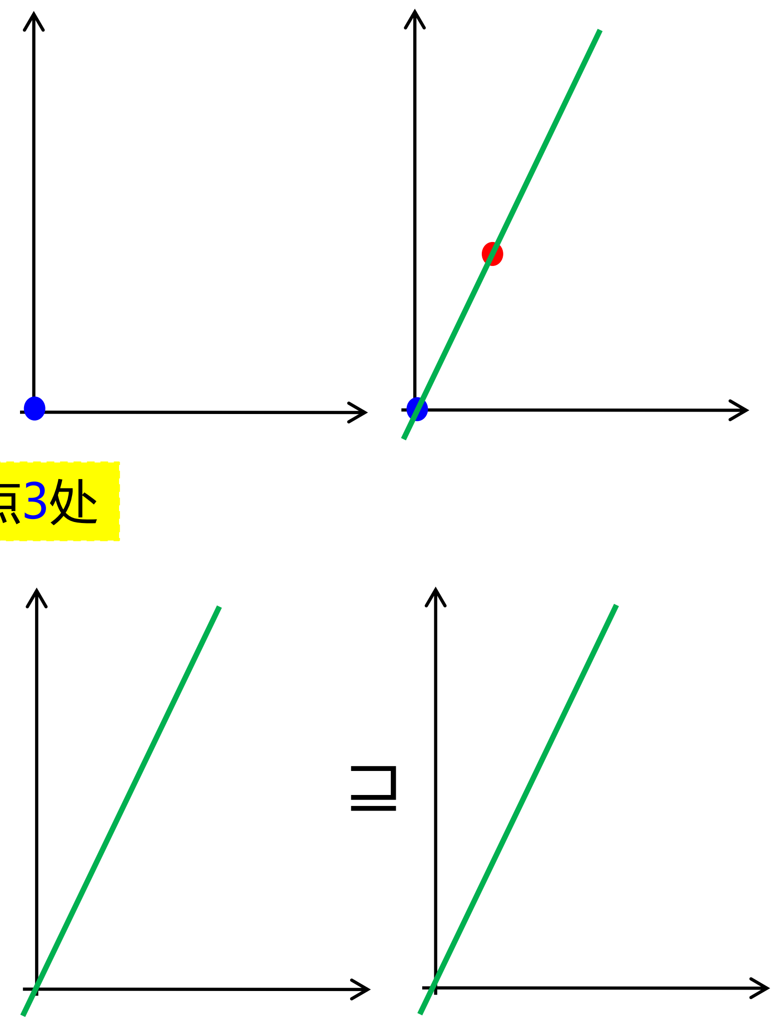
抽象分析的应用—检查程序错误

- 示例：检查数值相关错误（线性等式抽象）

```
1 x:=0;
2 y:=0;
3 while ( x<=99) do
4    x:=x+1;
5    y:=y+2;
done; 6
```



程序点3处



$y = 2x$

抽象分析的应用—检查程序错误

- 示例：检查数值相关错误

程序点4处不变式：$\{x \in [0,99],\ \ y=2x\}$

```
int A[198];
1 x:=0;
2 y:=0;
3 while ( x<=99) do
4    A[y]:=0;
5    x:=x+1;
6    y:=y+2;
done; 7
```

```
int A[199];
1 x:=0;
2 y:=0;
3 while ( x<=99) do
4    A[y]:=0;
5    x:=x+1;
6    y:=y+2;
done; 7
```

抽象分析的应用—检查程序错误

- 示例：检查数值相关错误

程序点4处不变式：$\{x \in [0,99],\quad y=2x\}$

```
int A[198];
1 x:=0;
2 y:=0;
3 while ( x<=99) do
4    A[y]:=0;
5    x:=x+1;
6    y:=y+2;
done; 7
```

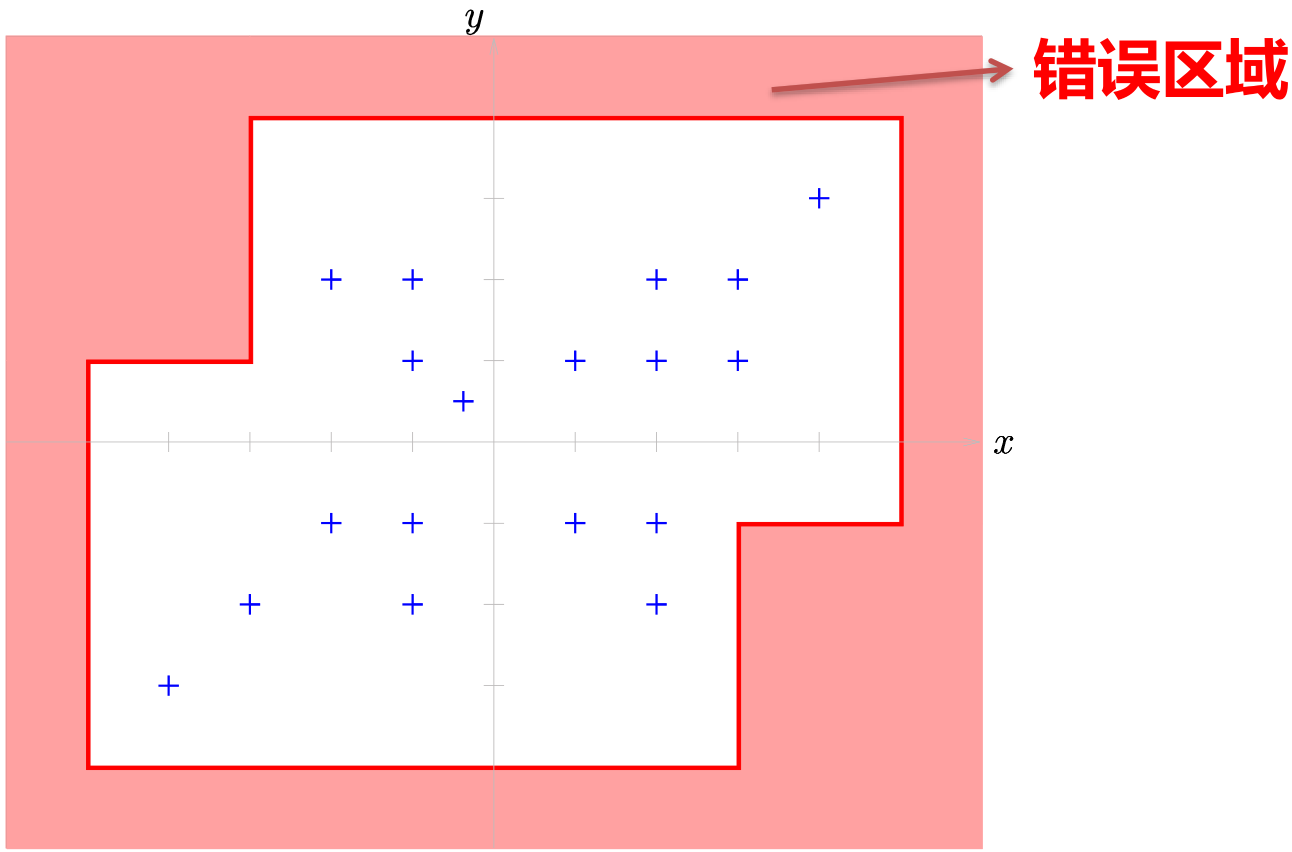
数组越界！

```
int A[199];
1 x:=0;
2 y:=0;
3 while ( x<=99) do
4    A[y]:=0;
5    x:=x+1;
6    y:=y+2;
done; 7
```

安全！

抽象分析的应用—检查程序错误

- 示例：根据不变式检查数值相关错误

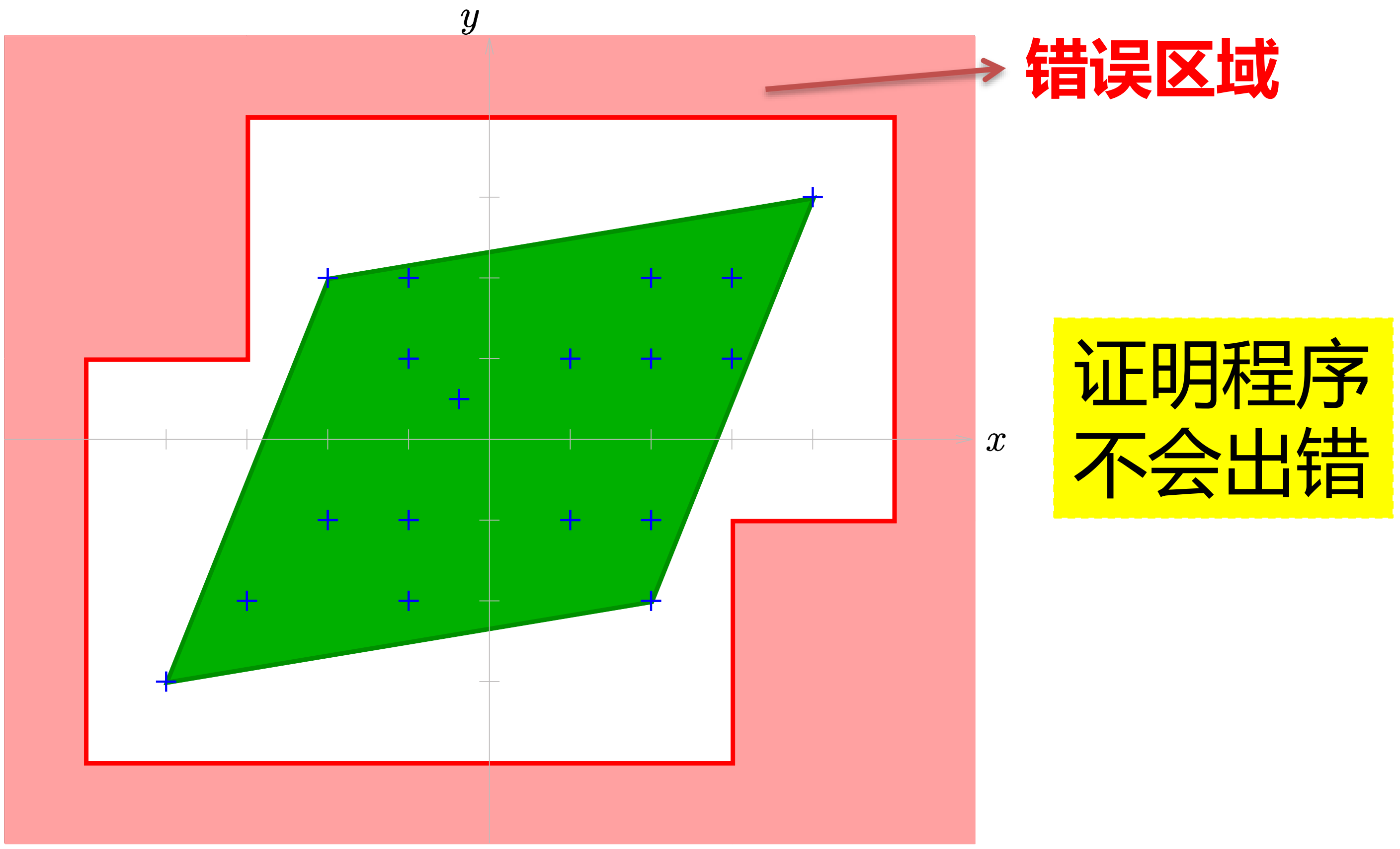


点的集合：每个点表示一个可能的程序状态

（程序状态指各程序变量的取值情况）

抽象分析的应用—检查程序错误

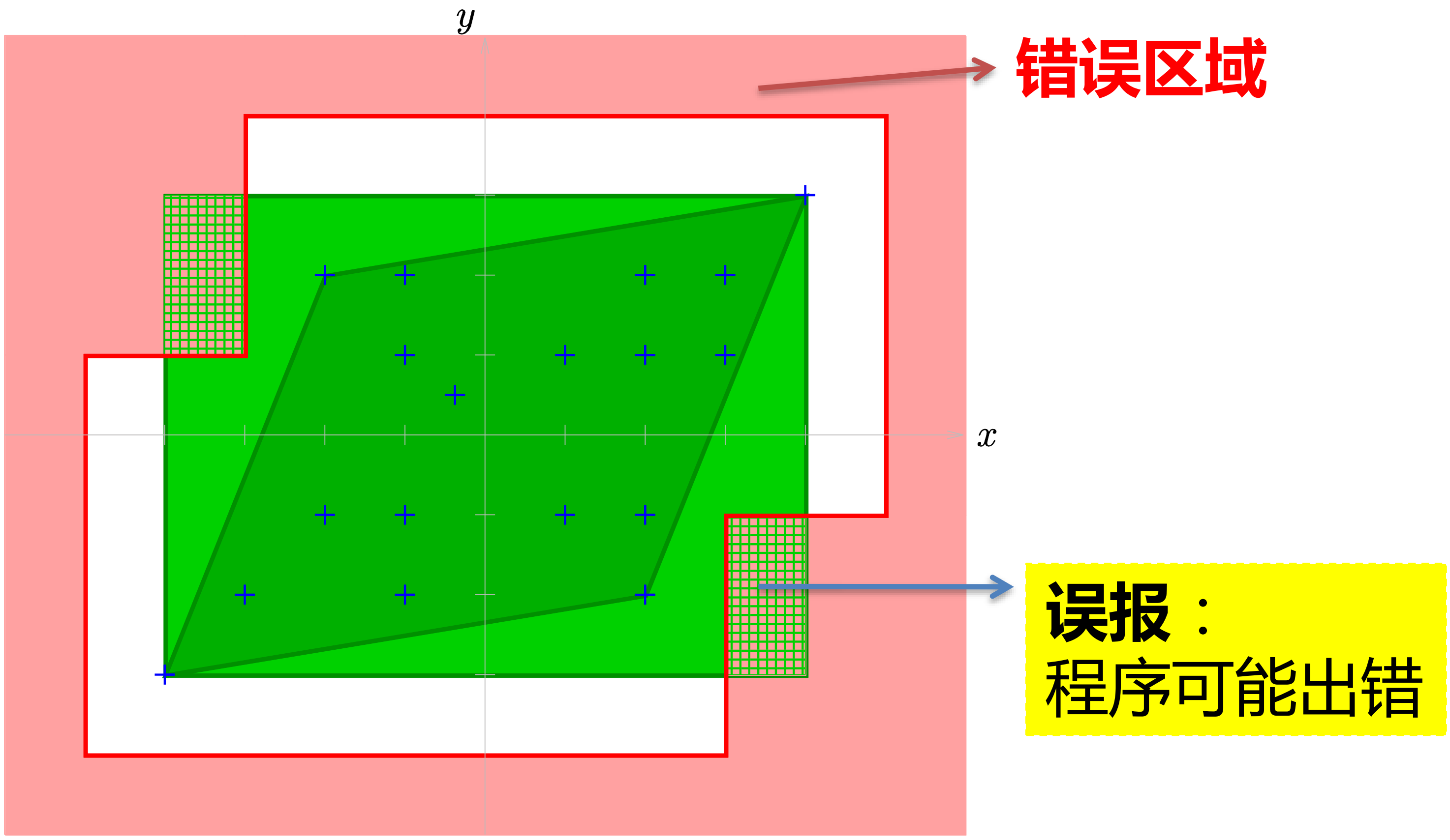
- 示例：根据不变式检查数值相关错误



多面体抽象域（$\Sigma_k a_k x_k \leq b$）

抽象分析的应用—检查程序错误

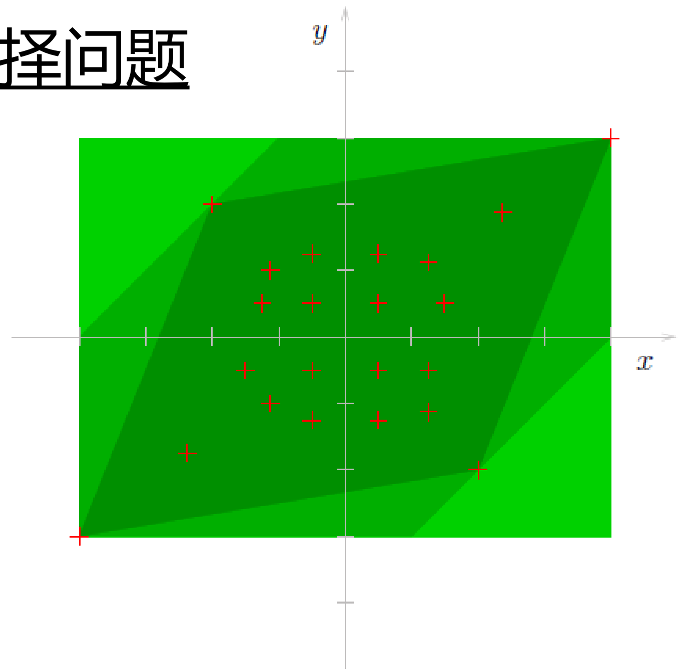
- 示例：根据不变式检查数值相关错误



区间抽象域（$a<=x<=b$）

抽象分析的应用—检查程序错误

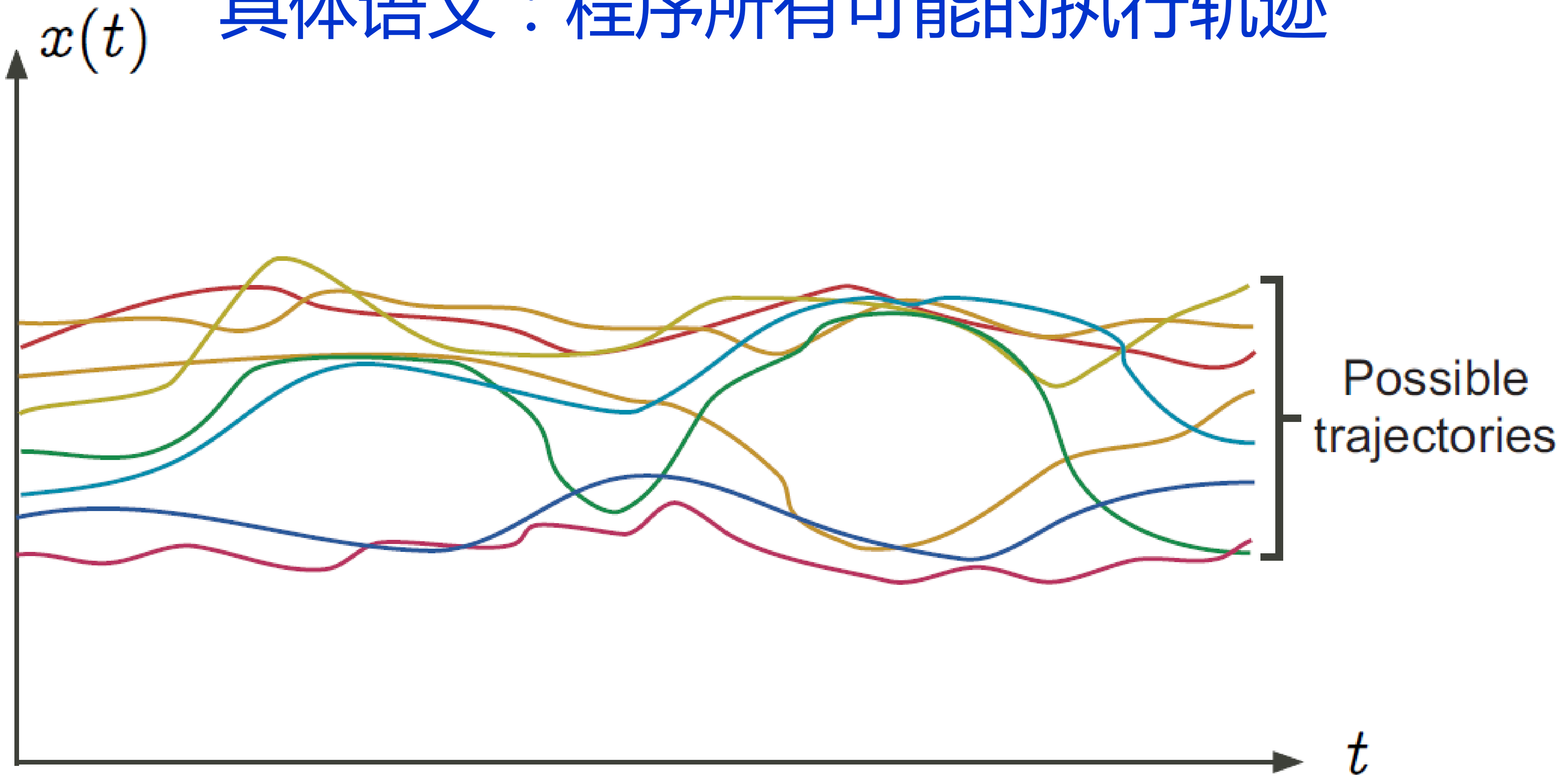
- 示例：根据不变式检查数值相关错误
  - <u>抽象域的选择问题</u>



包罗的点更少
→越精确
→但计算代价也越高

基于抽象解释的错误检测—直观解释

具体语义：程序所有可能的执行轨迹

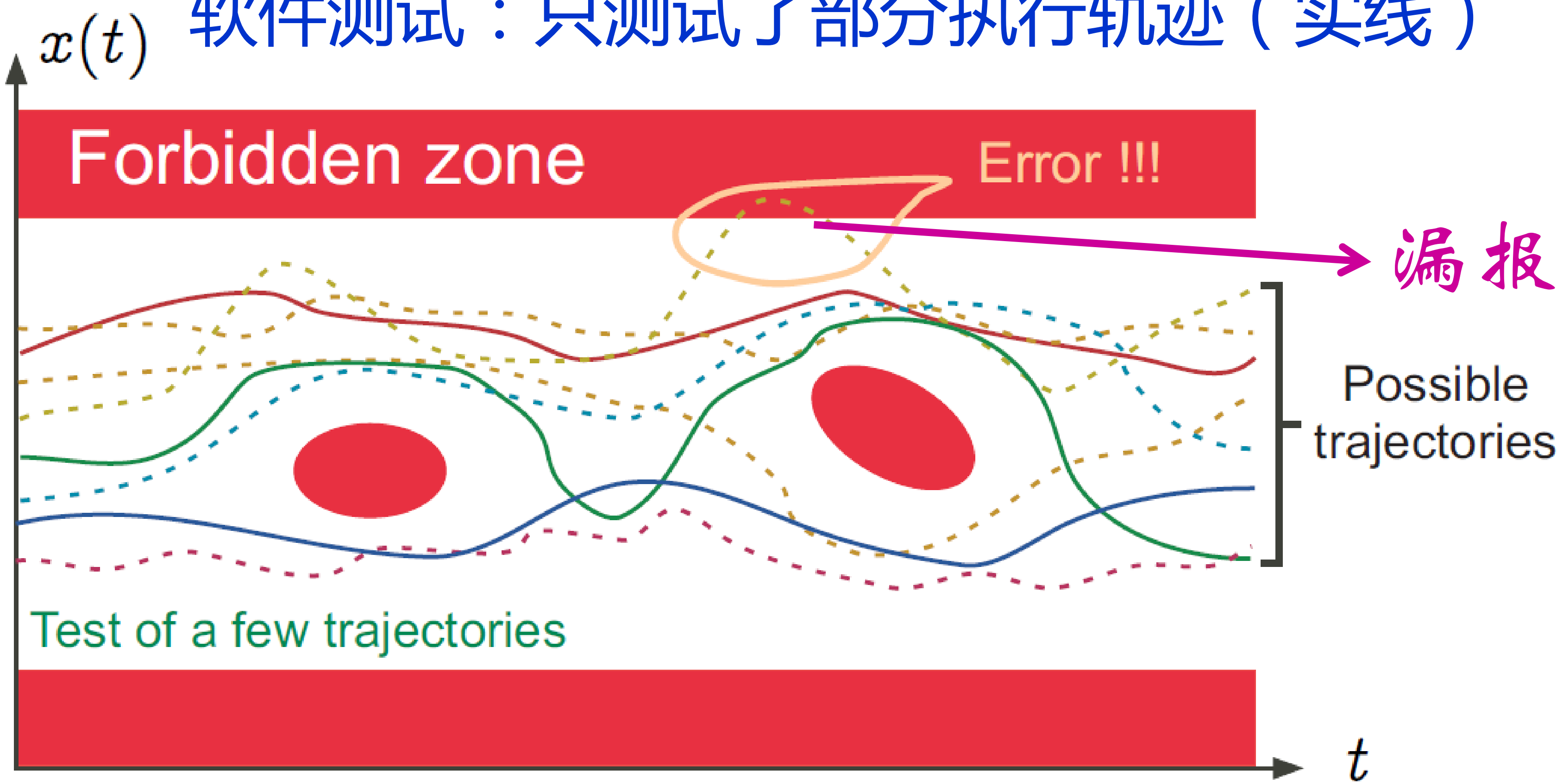


基于抽象解释的错误检测—直观解释

程序Safety性质：执行轨迹不进入红色区域

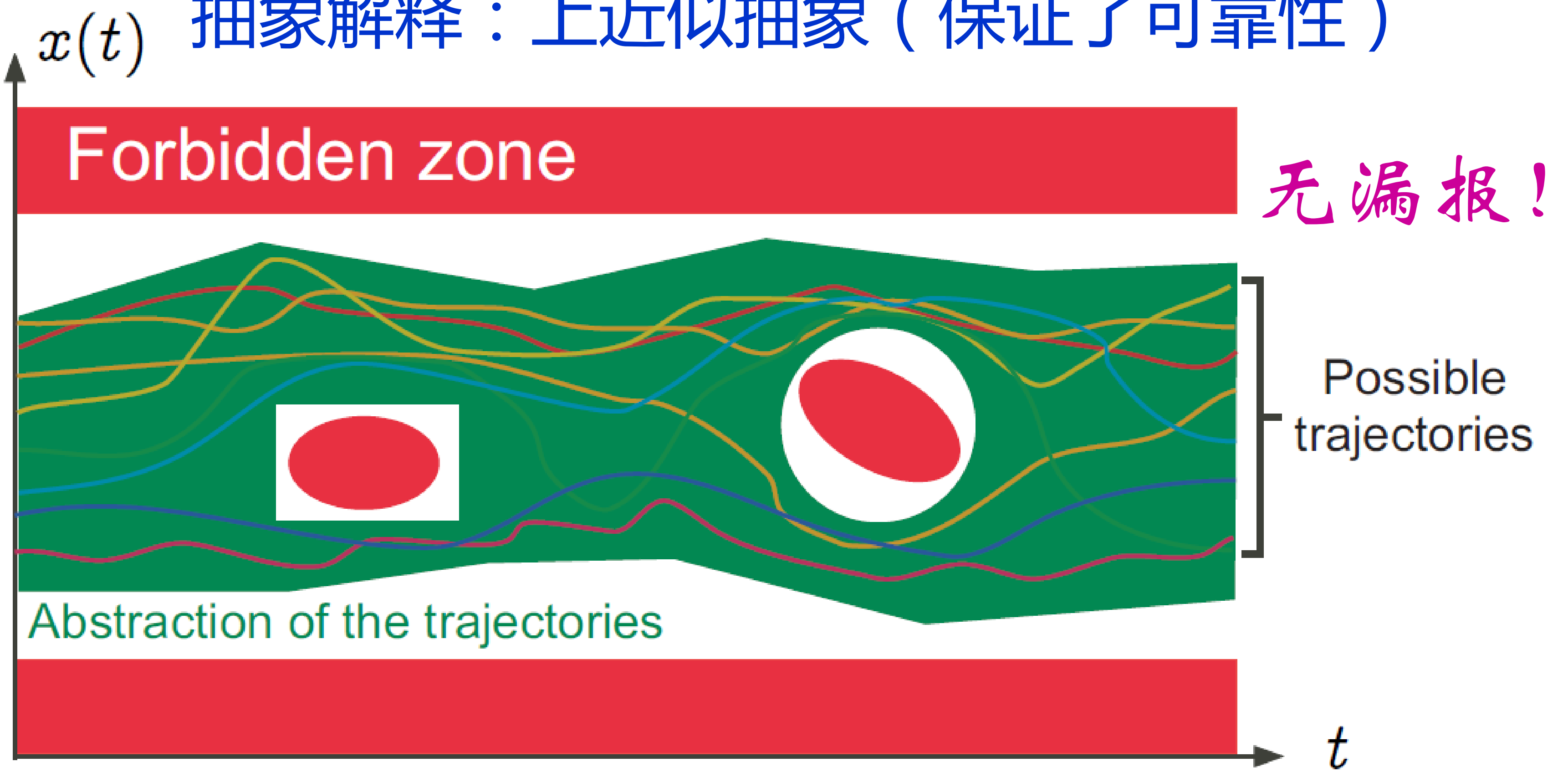
基于抽象解释的错误检测—直观解释

软件测试：只测试了部分执行轨迹（实线）



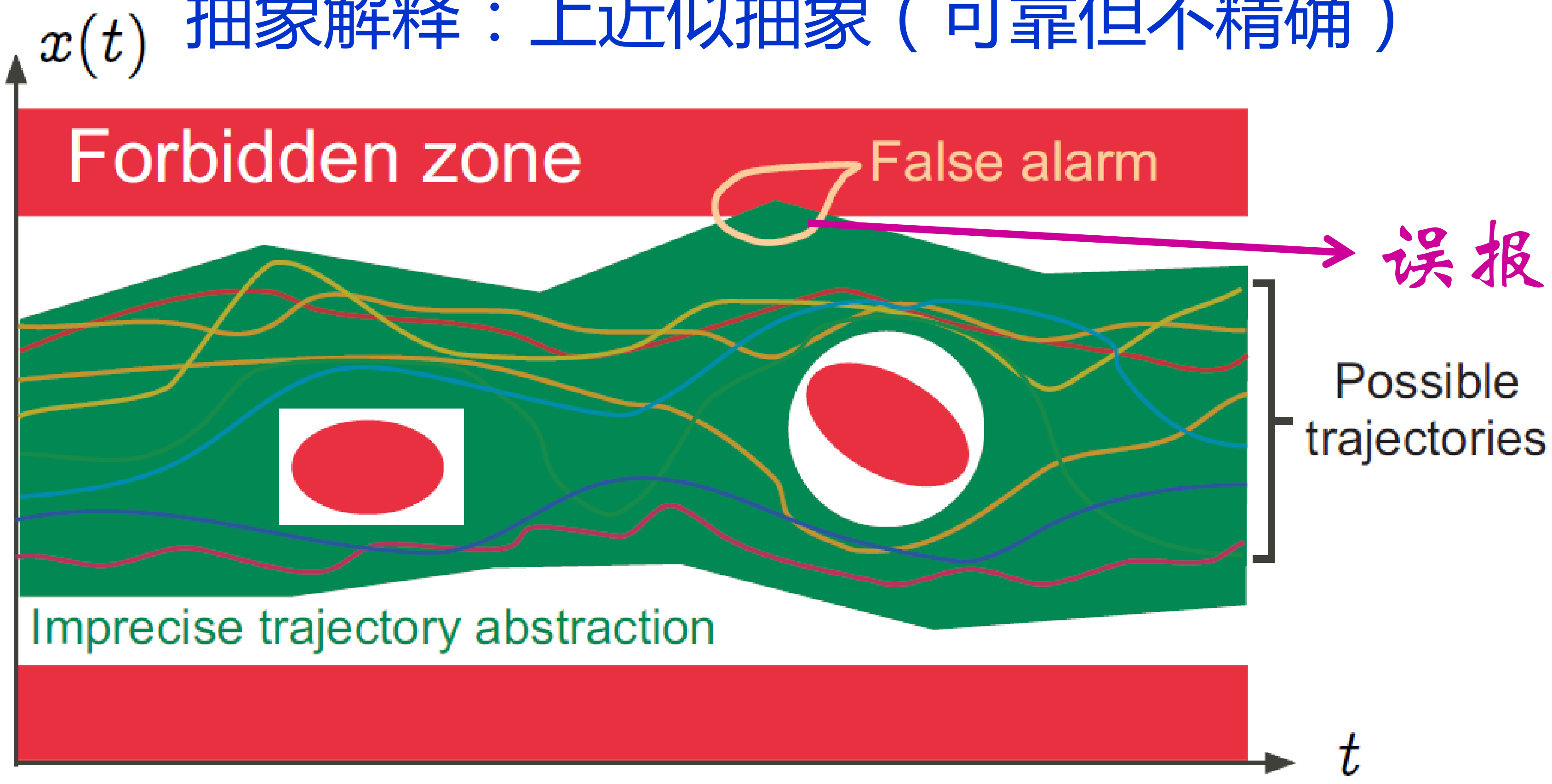
基于抽象解释的错误检测—直观解释

抽象解释：上近似抽象（保证了可靠性）



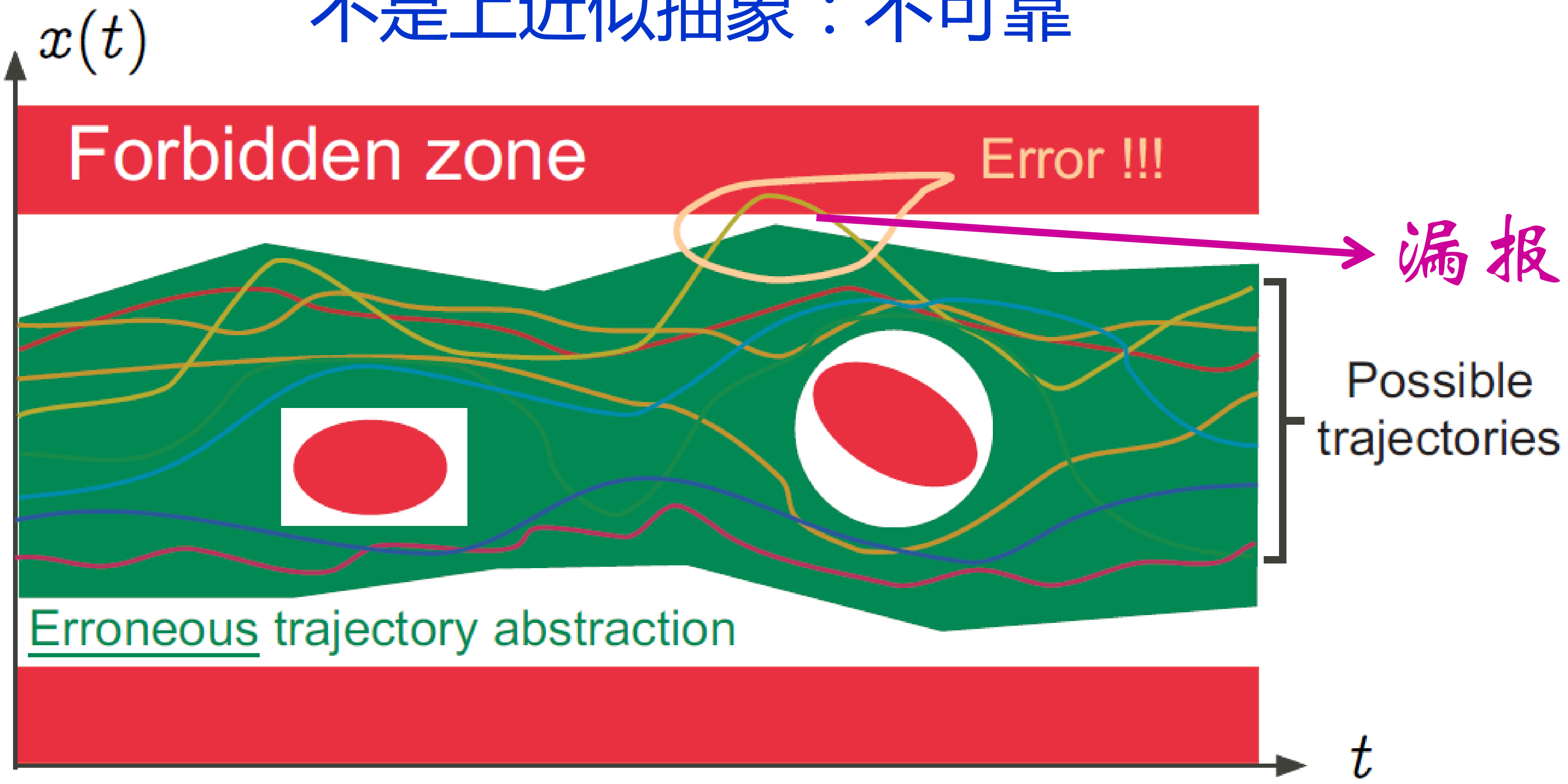
基于抽象解释的错误检测—直观解释

抽象解释：上近似抽象（可靠但不精确）

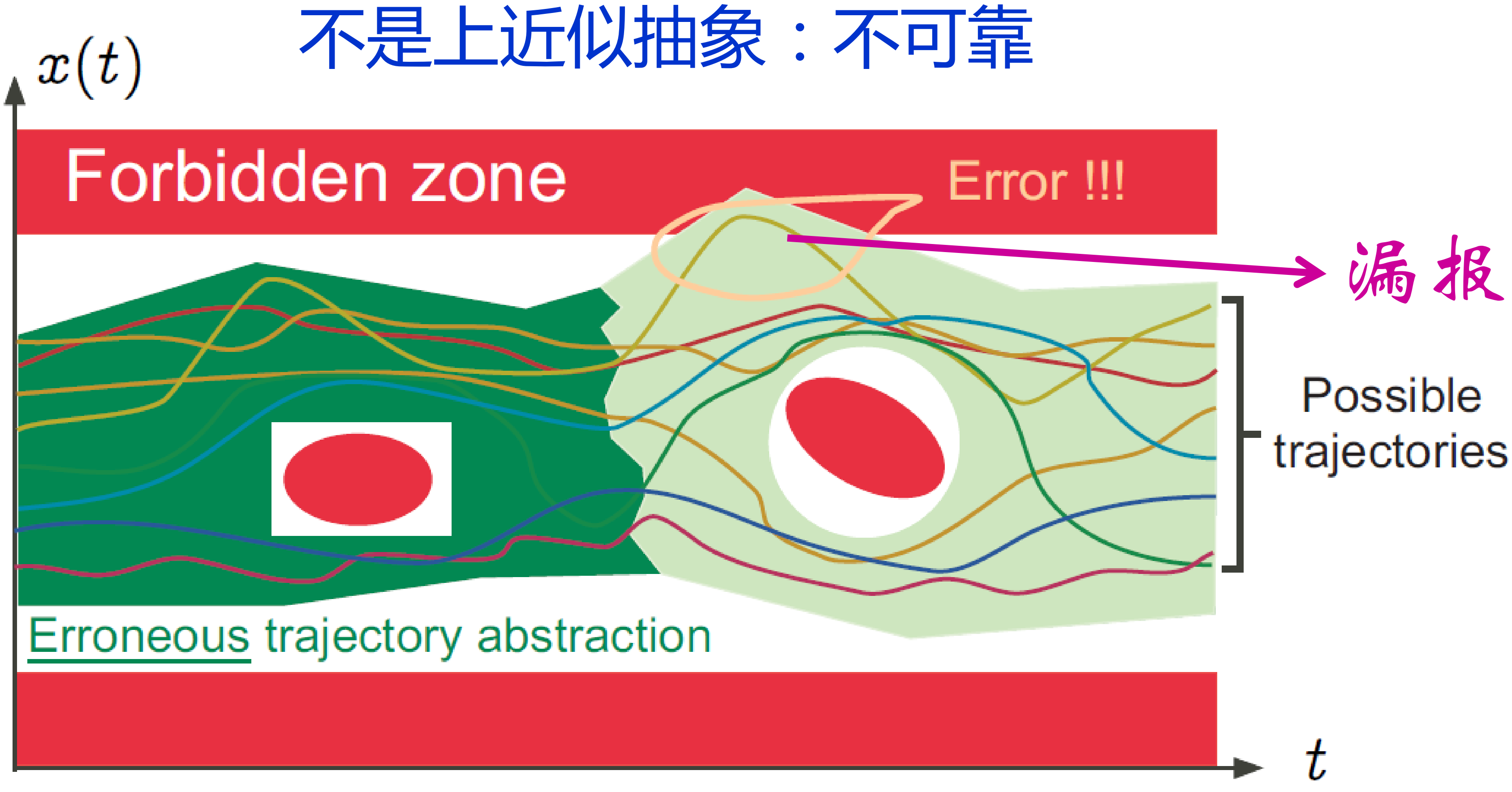


基于抽象解释的错误检测—直观解释

不是上近似抽象：不可靠



基于抽象解释的错误检测—直观解释



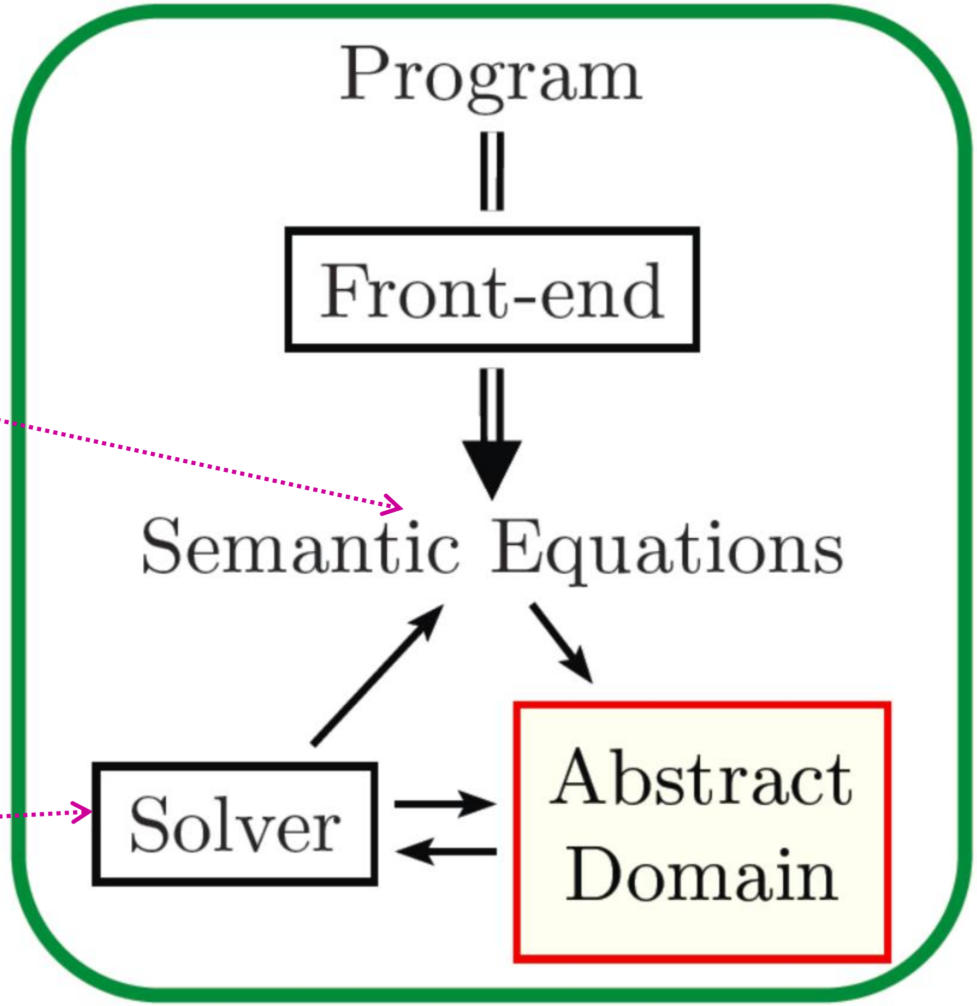
目录

抽象解释器

- 一般框架

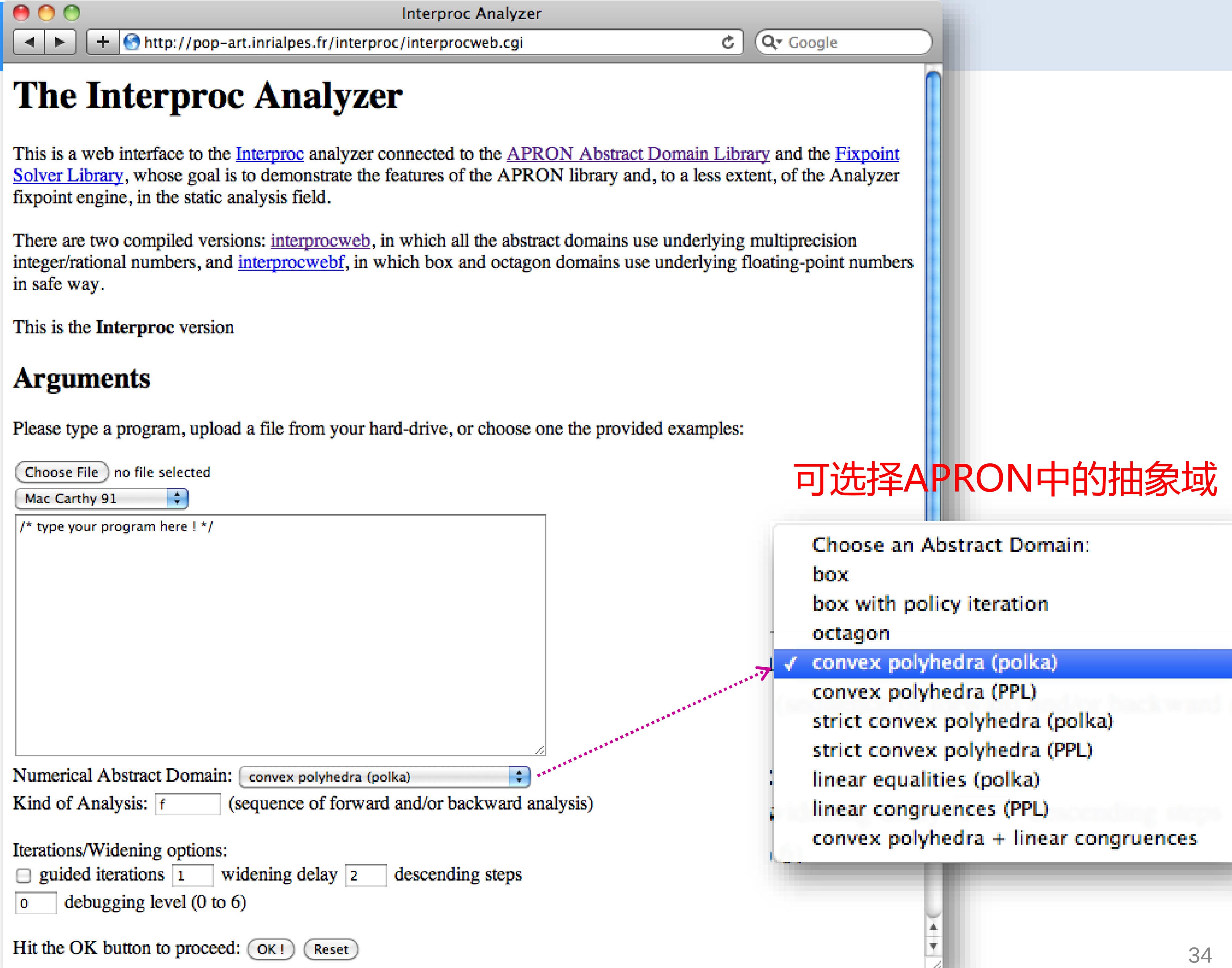
控制流图
（迁移系统）

基于迭代的
不动点求解



抽象解释器剖析

- 示例：Interproc
  - http://pop-art.inrialpes.fr/interproc/interprocweb.cgi
  - 开源工具
    - 用于展示开源抽象域库APRON的静态分析工具
    - 支持整型、浮点型等运算的分析
    - 能自动发现变量间的数值不变式
    - 支持过程间分析（包括递归函数）
    - 不支持数组、结构体等复杂数据结构、也不支持动态内存分配等



可选择APRON中的抽象域

Interproc Analyzer

http://pop-art.inrialpes.fr/inte

Source

```
/* exact semantics:
   if (n>=101) then n-10 else 91 */
proc MC(n:int) returns (r:int)
var t1:int, t2:int;
begin
  if (n>100) then
     r = n-10;
  else
     t1 = n + 11;
     t2 = MC(t1);
     r = MC(t2);
  endif;
end

var
a:int, b:int;
begin
  b = MC(a);
end
```

Interproc Analyzer

http://pop-art.inrialpes.fr/interproc/interprocweb.cgi

Google

Analysis Result

Run interprocweb or interprocwebf ?

Result

Annotated program after forward analysis

```
proc MC (n : int) returns (r : int) var t1 : int, t2 : int;
begin
  /* (L6 C5) top */
  if n > 100 then
     /* (L7 C17) [|n-101>=0|] */
     r = n - 10; /* (L8 C14)
                 [|-n+r+10=0; n-101>=0|] */
  else
    /* (L9 C6) [|-n+100>=0|] */
    t1 = n + 11; /* (L10 C17)
                 [|-n+t1-11=0; -n+100>=0|] */
    t2 = MC(t1); /* (L11 C17)
                 [|-n+t1-11=0; -n+100>=0; -n+t2-1>=0; t2-91>=0|] */
    r = MC(t2); /* (L12 C16)
                [|-n+t1-11=0; -n+100>=0; -n+t2-1>=0; t2-91>=0; r-t2+10>=0;
                  r-91>=0|] */
  endif; /* (L13 C8) [|-n+r+10>=0; r-91>=0|] */
end

var a : int, b : int;
begin
  /* (L18 C5) top */
  b = MC(a); /* (L19 C12)
             [|-a+b+10>=0; b-91>=0|] */
end
```

基于抽象解释的静态分析工具

- Interproc 实践
  - http://pop-art.inrialpes.fr/interproc/interprocweb.cgi

基于抽象解释的静态分析工具

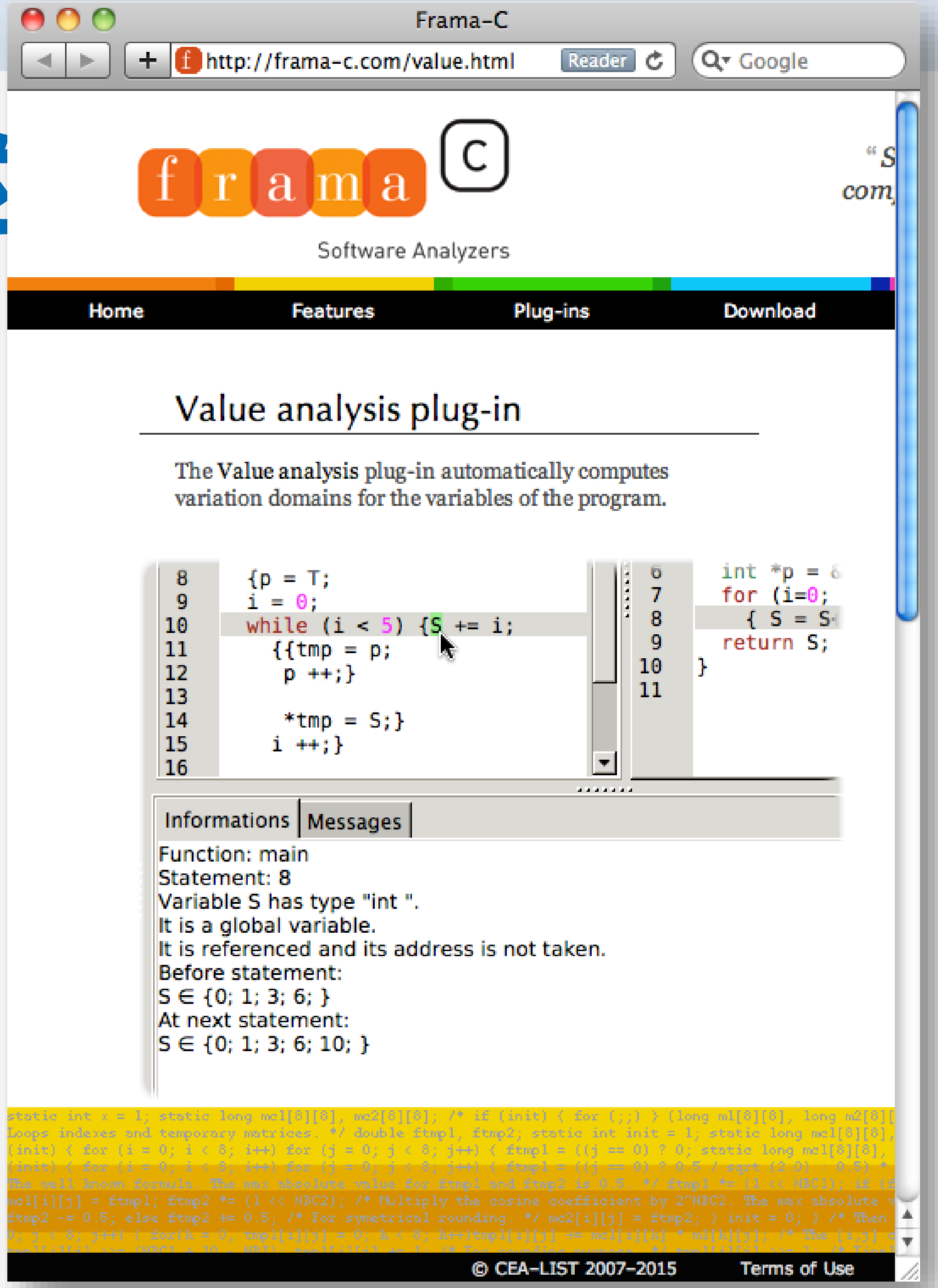
- 商业化工具
  - PolySpace (MathWorks)
  - ASTREE (AbsInt)
  - …
- 开源工具
  - Frama-C
  - …

基于抽象解释的静态分析工具

- PolySpace（MathWorks)
  - 商业化工具
  - <u>视频</u>

基于抽象解释的

- FramaC
  - http://frama-c.com/value.html
  - 开源工具
  - Value analysis插件
    - 基于值集合、区间集合的值范围分析
    - 支持标准C语法（包括数组、结构体、动态内存分配等）
    - 能够检查程序错误



小结

- 一、抽象解释概述
- 二、抽象解释理论的数学基础
- 三、具体语义下的静态分析
- 四、抽象语义下的静态分析
- 五、基于抽象解释的静态分析工具

参考资料

- Patrick Cousot. Abstract Interpretation in a Nutshell. http://www.di.ens.fr/~cousot/AI/IntroAbsInt.html
- Patrick Cousot. Abstract Interpretation. http://www.di.ens.fr/~cousot/AI/
- Michael I. Schwartzbach. Lecture Notes on Static Analysis. http://www.itu.dk/people/brabrand/UFPE/Data-Flow -Analysis/static.pdf
- Patrick Cousot. A very informal introduction to the principles of abstract interpretation. http://web.mit.edu/16.399/www/lecture_01-intro/Cousot_MIT_2005_Course_01_4-1.pdf
- Patrick Cousot, Radhia Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In ACM POPL‘77, 1977.

谢谢！