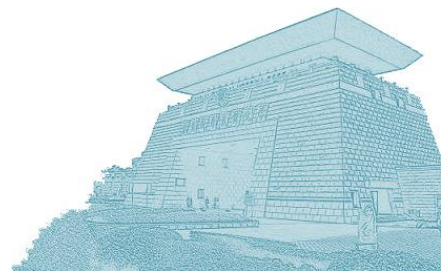


# 抽象解释 及其在静态分析中的应用

陈立前  
国防科技大学



# 思考—Pretest

- 如下程序有没有错误？

```
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
```

# 思考—Pretest

- 抽象解释可以用来自动检查程序错误

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;
```



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;
```

安全！

# 本讲内容介绍

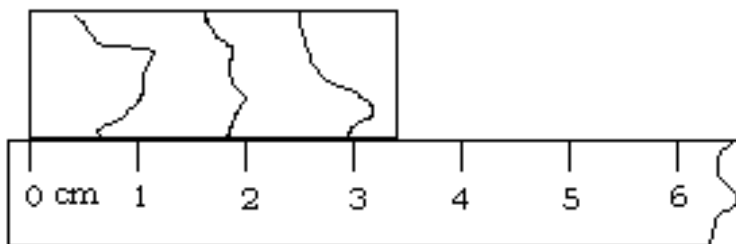
- 一、抽象解释概述
- 二、抽象解释理论的数学基础
- 三、具体语义下的静态分析
- 四、抽象语义下的静态分析
- 五、基于抽象解释的静态分析工具

# 本讲内容介绍

- 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

# 抽象解释

- 抽象解释: 对程序语义进行抽象(或近似)的通用理论



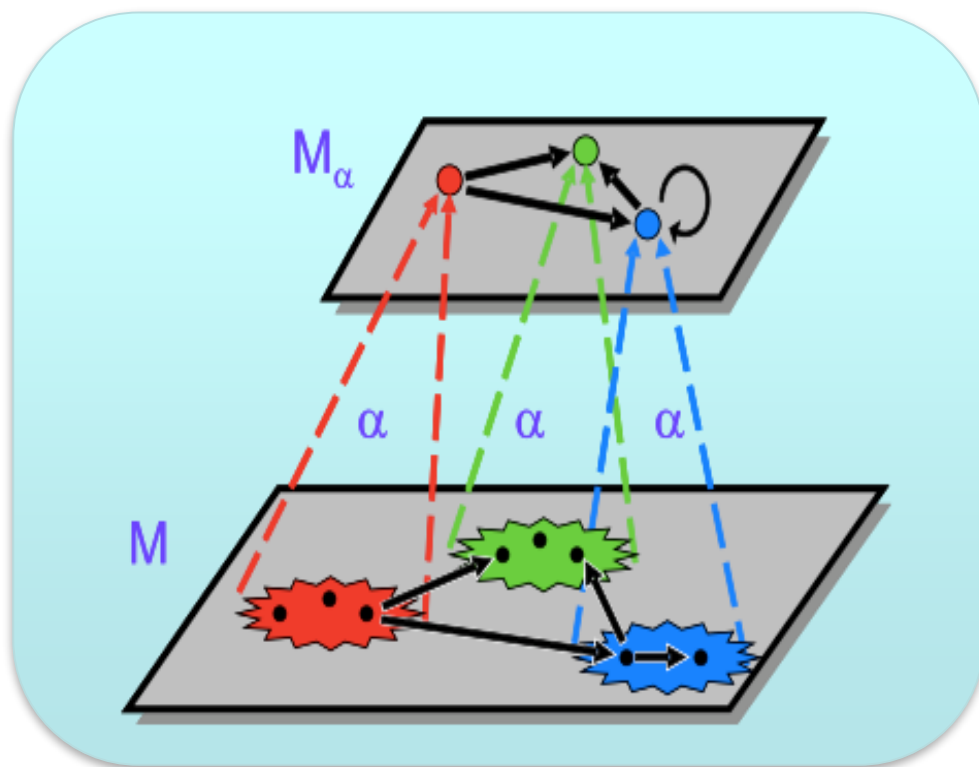
近似

$(12345678 + 123456) * 1234$   
的结果是正数还是负数？

抽象  
(忽略与当前问题无关的方面)

# 抽象解释

- 核心思想：形式化地描述 抽象（或近似）
  - 为静态分析提供了统一框架



抽象空间

具体空间

# 抽象解释

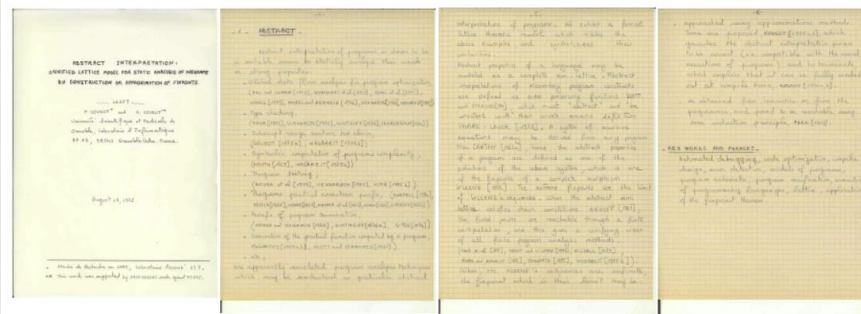
## ● 发展历史—理论

ABSTRACT INTERPRETATION : A UNIFIED LATTICE MODEL FOR STATIC ANALYSIS  
OF PROGRAMS BY CONSTRUCTION OR APPROXIMATION OF FIXPOINTS

Patrick Cousot\* and Radhia Cousot\*\* @ POPL 1977

### On submitting to POPL

- For POPL'77, we submit (on Aug. 12, 1976) copies of a two-hands written manuscript of 100 pages. The paper is accepted !

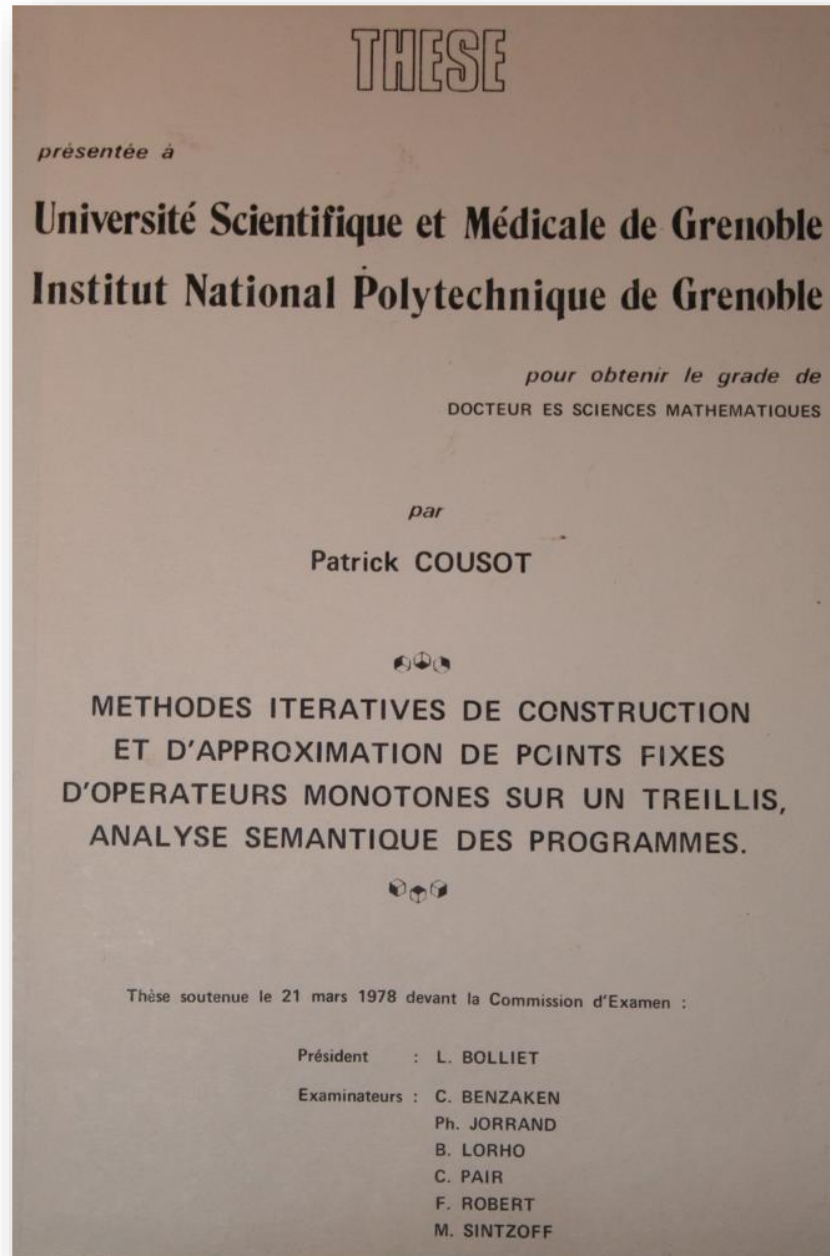




# 抽象解释

- 发展历史—理论

Patrick Cousot  
博士论文  
@ 1978



# 抽象解释

## ● 发展历史—工具

- 1996~：涌现了多个基于抽象解释的程序分析工具
  - SLAM (Microsoft)
  - Coverity Prevent (Coverity)
  - PolySpace (MathWorks)
  - ASTREE (AbsInt)
  - Clousot (Microsoft)
  - Zoncolan (Facebook)
  - CGS (NASA)
  - Sparrow (Fasoo)
  - ...

### Products [\[edit\]](#)

*Coverity Static Analysis Verification Engine (Coverity SAVE)* is a [static code analysis](#) tool for [C](#), [C++](#), [C#](#) and [Java](#). Coverity commercialized a research tool for finding bugs through static analysis,<sup>[1]</sup> the Stanford Checker, which used [abstract interpretation](#) to identify [defects](#) in [source code](#).<sup>[2]</sup>

### Polyspace

From Wikipedia, the free encyclopedia

*For the computational complexity class, see [PSPACE](#).*

**Polyspace** is a static code analysis tool for large-scale analysis by [abstract interpretation](#) to detect, c

**Astrée** is a [static analyzer](#) based on [abstract interpretation](#). It analyzes programs written in [the C programming language](#) and outputs an exhaustive list of possible runtime errors and [assertion](#) violations.

# 抽象解释

## ● 发展历史—工业应用

### ➤ Mathworks的Polyspace

- 96年欧洲阿里亚娜5号火箭升空后爆炸，法国INRIA与欧洲宇航局共同组织力量研制PolySpace，07年被Mathworks收购

### ➤ 美国宇航局的CGS(由美国KT公司CodeHawk商业化)

- 美国宇航局的火星探路者、火星探测器、深空1号等

### ➤ 巴黎高师的ASTREE（由德国AbsInt公司商业化）

- 空客A340/A380飞机飞控软件、欧空局自动货运飞船



空中客车A340主飞控软件  
(13.2万行C 代码)



空中客车A380主飞控软件  
(约35万行C 代码)



欧空局自动货运飞船ATV“儒勒·凡尔纳”号(约19万行C 代码)

# 抽象解释

## ● 发展历史

30 years of AI @ POPL 2008

### 30 years of Abstract Interpretation

San Francisco USA, January 9, 2008  
affiliated with POPL 2008



Thomas Ball

Patrick Cousot

Chris Hankin

Manuel Hermenegildo

Chris Hote

Neil Jones

Ganesan Ramalingam

Famantanantsoa Randimbivolona

Francesco Ranzato

Mooly Sagiv

David Sands

Andreas Podelski

Kwangkuen Yi



30 years ago, in March 1978, Patrick Cousot defended his PhD thesis (Docteur es Sciences Mathématiques), which started the era of abstract interpretation. 30YAI celebrates this event by inviting some of the most representative scientists in the field, showing the relevance,

perspectives and challenges of abstract interpretation in programming languages and systems.



Abstract interpretation is a theory of sound approximation of mathematical structures, in particular those involved in the behavior of computer systems. It allows the systematic derivation of sound methods and algorithms for approximating undecidable or highly complex problems in various areas of computer science like for instance in static program analysis, system verification, model checking, program transformation, process calculi, security, software watermarking, type inference, theorem proving, constraint solving, parsing and comparative semantics, systems biology.

Organizers:

Roberto Giacobazzi Dave Schmidt





# 抽象解释

- 发展历史

Next 40 years of Abstract Interpretation @ POPL 2017



"Next 40 years of Abstract Interpretation"

**Abstract Interpretation – 40  
years back + some years ahead**

N40AI 2017  
January 21st, 2017  
Paris, France

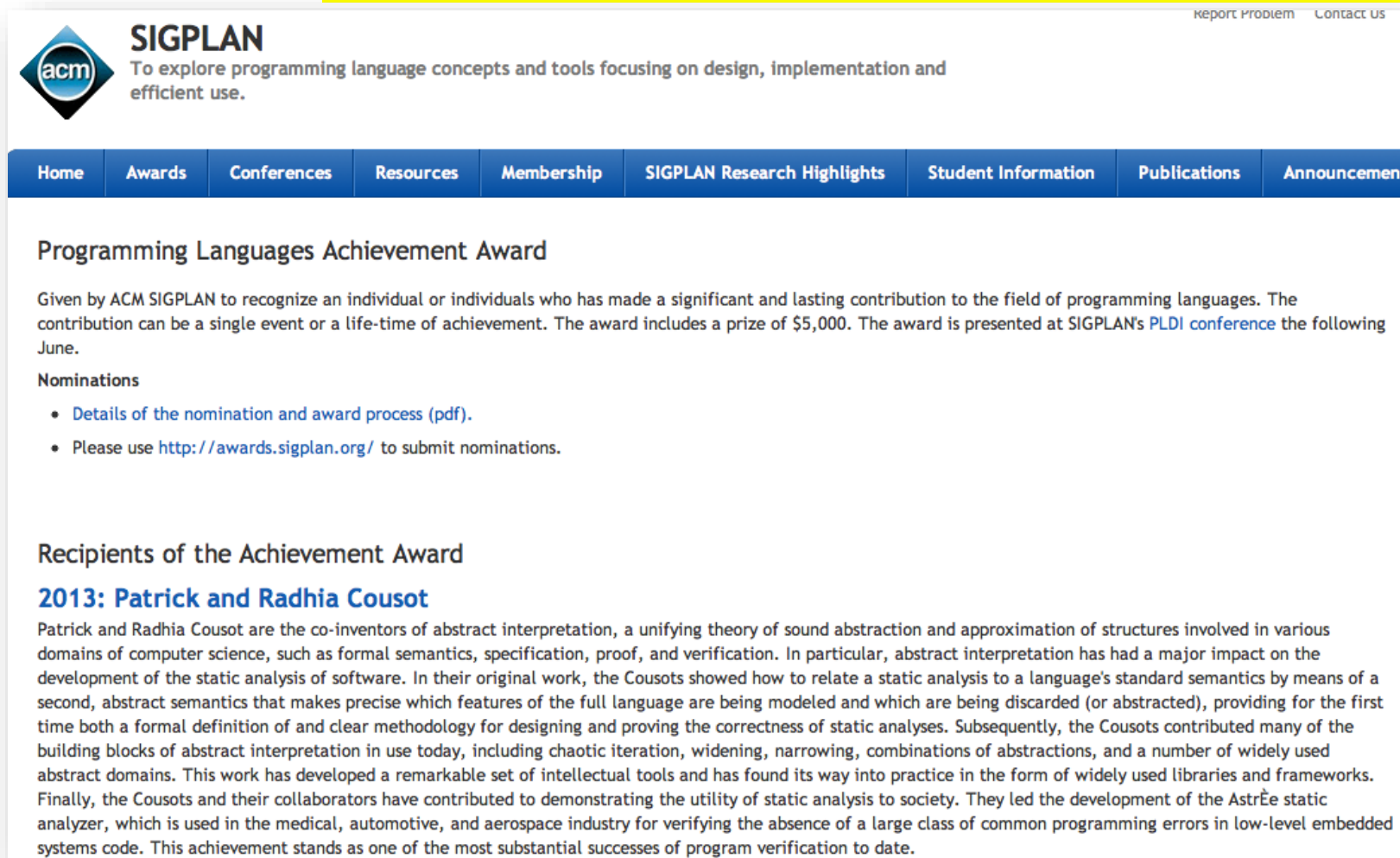
**Patrick Cousot**

[pcousot@cs.nyu.edu](mailto:pcousot@cs.nyu.edu) [cs.nyu.edu/~pcousot](http://cs.nyu.edu/~pcousot)

# 抽象解释

## ● 发展历史

## 2013年ACM SIGPLAN 程序语言成就奖



The screenshot shows the ACM SIGPLAN website. At the top left is the ACM SIGPLAN logo with the text "SIGPLAN To explore programming language concepts and tools focusing on design, implementation and efficient use." To the right are links for "Report Problem" and "Contact Us". Below this is a navigation bar with links: Home, Awards, Conferences, Resources, Membership, SIGPLAN Research Highlights, Student Information, Publications, and Announcements. The main content area is titled "Programming Languages Achievement Award". It contains a paragraph describing the award, given by ACM SIGPLAN to recognize significant contributions to programming languages, with a prize of \$5,000 presented at the SIGPLAN PLDI conference. Below this is a "Nominations" section with two bullet points: "Details of the nomination and award process (pdf)." and "Please use <http://awards.sigplan.org/> to submit nominations." The next section is "Recipients of the Achievement Award", featuring the 2013 recipients, Patrick and Radhia Cousot. A detailed paragraph follows, describing their work on abstract interpretation, a unifying theory of sound abstraction and approximation of structures, and its impact on static analysis of software. It mentions their original work on relating static analysis to standard semantics, the development of the Astrée static analyzer, and its application in the medical, automotive, and aerospace industries for verifying the absence of common programming errors in low-level embedded systems code.

**SIGPLAN**  
To explore programming language concepts and tools focusing on design, implementation and efficient use.

Report Problem Contact Us

Home Awards Conferences Resources Membership SIGPLAN Research Highlights Student Information Publications Announcements

### Programming Languages Achievement Award

Given by ACM SIGPLAN to recognize an individual or individuals who has made a significant and lasting contribution to the field of programming languages. The contribution can be a single event or a life-time of achievement. The award includes a prize of \$5,000. The award is presented at SIGPLAN's [PLDI conference](#) the following June.

**Nominations**

- [Details of the nomination and award process \(pdf\).](#)
- Please use <http://awards.sigplan.org/> to submit nominations.

### Recipients of the Achievement Award

#### 2013: Patrick and Radhia Cousot

Patrick and Radhia Cousot are the co-inventors of abstract interpretation, a unifying theory of sound abstraction and approximation of structures involved in various domains of computer science, such as formal semantics, specification, proof, and verification. In particular, abstract interpretation has had a major impact on the development of the static analysis of software. In their original work, the Cousots showed how to relate a static analysis to a language's standard semantics by means of a second, abstract semantics that makes precise which features of the full language are being modeled and which are being discarded (or abstracted), providing for the first time both a formal definition of and clear methodology for designing and proving the correctness of static analyses. Subsequently, the Cousots contributed many of the building blocks of abstract interpretation in use today, including chaotic iteration, widening, narrowing, combinations of abstractions, and a number of widely used abstract domains. This work has developed a remarkable set of intellectual tools and has found its way into practice in the form of widely used libraries and frameworks. Finally, the Cousots and their collaborators have contributed to demonstrating the utility of static analysis to society. They led the development of the Astrée static analyzer, which is used in the medical, automotive, and aerospace industry for verifying the absence of a large class of common programming errors in low-level embedded systems code. This achievement stands as one of the most substantial successes of program verification to date.

# 抽象解释

- 发展历史      2018年 约翰·冯诺依曼奖



Patrick Cousot awarded  
John von Neumann  
Medal

Patrick Cousot is the recipient of the  
IEEE John von Neumann medal, given  
"for outstanding achievements in  
computer-related science and  
technology".

[Read More](#)



## IEEE JOHN VON NEUMANN MEDAL RECIPIENTS

2018    PATRICK COUSOT  
Professor, New York University,  
New York, New York, USA

"For introducing abstract interpretation, a  
powerful framework for automatically  
calculating program properties with broad  
application to verification and optimization."

# 本讲内容介绍

- 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



# 抽象解释的数学基础

- 序理论：（完全）偏序、格
- 不动点理论
- Galois连接
- 近似 (approximation)

# 偏序

- 定义

- 若集合  $D$  上的二元关系  $\sqsubseteq$  是自反的、传递的和反对称的，则称  $\sqsubseteq$  是  $D$  上的偏序关系(Partial order)，称  $(D, \sqsubseteq)$  为偏序集(Poset)

- 示例

- $(\mathbb{Z}, \leq)$
- $(\wp(X), \sqsubseteq)$
- $(\mathbb{Z}^2, \sqsubseteq)$ , where  $(a, b) \sqsubseteq (a', b') \Leftrightarrow a \geq a' \wedge b \leq b'$

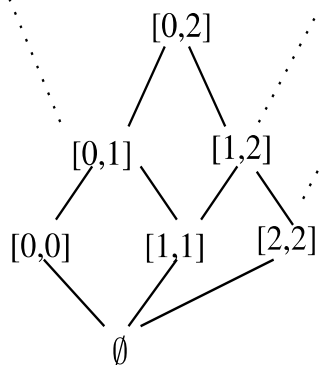
# 格

- 定义

- 偏序集  $(D, \sqsubseteq)$  称为格, 如果  $D$  中任何两个元素  $a, b$  都有最小上界(即上确界, 记作  $a \sqcup b$ ) 和最大下界(即下确界, 记作  $a \sqcap b$ )。该格记作  $(D, \sqsubseteq, \sqcup, \sqcap)$ 。

- 示例

- 整数  $(\mathbb{Z}, \leq, \max, \min)$
- 整数区间  $(\{ [a, b] \mid a, b \in \mathbb{Z}, a \leq b \} \cup \{ \emptyset \}, \subseteq, \sqcup, \sqcap)$



$$\begin{aligned} [a, b] \sqcup [a', b'] &\stackrel{\text{def}}{=} [\min(a, a'), \max(b, b')] \\ [a, b] \sqcap [a', b'] &\stackrel{\text{def}}{=} [\max(a, a'), \min(b, b')], \text{ or } \emptyset \text{ if } \max(a, a') > \min(b, b') \end{aligned}$$

# 格

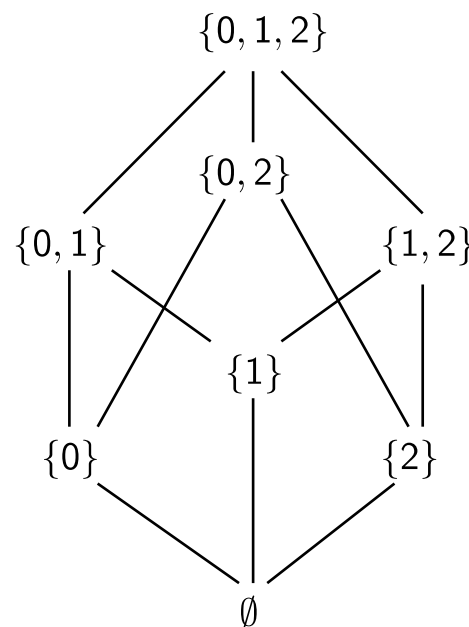
- 定义

- 偏序集  $(D, \sqsubseteq)$  称为完全格, 如果  $D$  的所有子集  $X$  都有最小上界  $\sqcup X$  和最大下界  $\sqcap X$ 。特别地,  $D$  总存在最小元  $\perp = \sqcup \emptyset$  和最大元  $\top = \sqcap D$ 。该完全格记作  $(D, \sqsubseteq, \sqcup, \sqcap, \perp, \top)$ 。

- 示例

- 幂集  $(\wp(S), \subseteq, \cup, \cap, \emptyset, S)$

$(\wp(\{0,1,2\}), \subseteq, \cup, \cap, \emptyset, \{0,1,2\})$



# 格

- 定义

- 偏序集  $(D, \sqsubseteq)$  称为完全格, 如果  $D$  的所有子集  $X$  都有最小上界  $\sqcup X$  和最大下界  $\sqcap X$ 。特别地,  $D$  总存在最小元  $\perp = \sqcup \emptyset$  和最大元  $\top = \sqcap D$ 。该完全格记作  $(D, \sqsubseteq, \sqcup, \sqcap, \perp, \top)$ 。

- 示例

- 幂集  $(\wp(S), \subseteq, \cup, \cap, \emptyset, S)$
- 带无穷界的整数区间

$$(\mathbb{I}, \subseteq, \sqcup, \sqcap, \emptyset, [-\infty, +\infty])$$

$$\mathbb{I} \stackrel{\text{def}}{=} \{ [a, b] \mid a \in \mathbb{Z} \cup \{-\infty\}, b \in \mathbb{Z} \cup \{+\infty\}, a \leq b \},$$

$$\sqcup_{i \in I} [a_i, b_i] \stackrel{\text{def}}{=} [\min_{i \in I} a_i, \max_{i \in I} b_i],$$

$$\sqcap_{i \in I} [a_i, b_i] \stackrel{\text{def}}{=} [\max_{i \in I} a_i, \min_{i \in I} b_i], \text{ or } \emptyset \text{ if } \max > \min.$$

# 完全偏序

## ● 链

- 设  $(D, \sqsubseteq)$  为偏序集, 称  $D$  的非空子集  $S$  为链, 如果  $S$  中任意两个元素关于  $\sqsubseteq$  皆可比。

## ● 完全偏序

- 设  $(D, \sqsubseteq)$  为偏序集, 若
  - 1)  $D$  有最小元, 记为  $\perp_D$ ;
  - 2) 对  $D$  中每个链  $S$ , 其最小上界  $\sqcup S$  存在。

则称  $(D, \sqsubseteq)$  为完全偏序(CPO)。该 CPO 记作  $(D, \sqsubseteq, \sqcup, \perp)$

## ● 示例



$(\mathbb{N}, \leq)$

⋮  
3  
|  
2  
|  
1  
|  
0

# 完全偏序

## ● 链

- 设  $(D, \sqsubseteq)$  为偏序集, 称  $D$  的非空子集  $S$  为链, 如果  $S$  中任意两个元素关于  $\sqsubseteq$  皆可比。

## ● 完全偏序

- 设  $(D, \sqsubseteq)$  为偏序集, 若
  - 1)  $D$  有最小元, 记为  $\perp_D$ ;
  - 2) 对  $D$  中每个链  $S$ , 其最小上界  $\sqcup S$  存在。

则称  $(D, \sqsubseteq)$  为完全偏序(CPO)。该 CPO 记作  $(D, \sqsubseteq, \sqcup, \perp)$

## ● 示例



$(\mathbb{N}, \leq)$

0  
1  
2  
3  
⋮

不是CPO

$(\mathbb{N} \cup \{\infty\}, \leq)$

0  
1  
2  
3  
⋮  
∞

# 完全偏序

## ● 链

- 设  $(D, \sqsubseteq)$  为偏序集, 称  $D$  的非空子集  $S$  为链, 如果  $S$  中任意两个元素关于  $\sqsubseteq$  皆可比。

## ● 完全偏序

- 设  $(D, \sqsubseteq)$  为偏序集, 若
  - 1)  $D$  有最小元, 记为  $\perp_D$ ;
  - 2) 对  $D$  中每个链  $S$ , 其最小上界  $\sqcup S$  存在。

则称  $(D, \sqsubseteq)$  为完全偏序(CPO)。该 CPO 记作  $(D, \sqsubseteq, \sqcup, \perp)$

## ● 示例



$(\mathbb{N}, \leq)$

0  
1  
2  
3  
⋮

不是CPO

$(\mathbb{N} \cup \{\infty\}, \leq)$

0  
1  
2  
3  
⋮  
∞

是CPO



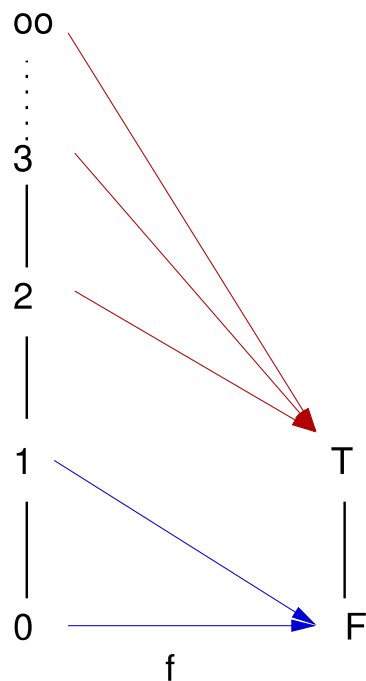
# 函数

- 偏序集 $(D, \sqsubseteq)$ 和 $(D', \sqsubseteq')$ 上的函数 $f : D \rightarrow D'$  是**单调**的, 如果只要 $x \sqsubseteq y$ 则  $f(x) \sqsubseteq' f(y)$
- 设  $(D, \sqsubseteq)$ 和  $(D', \sqsubseteq')$  是 CPO, 函数  $f : D \rightarrow D'$ 。若对于  $D$  中的每个链  $S$ ,  $\sqcup f(S)$  存在且有  $f(\sqcup S) = \sqcup f(S)$ , 则称  $f$  是  **$\sqcup$ -连续的**
- $f$  是  $\sqcup$ -连续的, 当且仅当  $f$  是单调的且对于  $D$  中每个链  $S$ , 有  $f(\sqcup S) \sqsubseteq \sqcup f(S)$ 
  - 若  $D$  只含有穷链, 则  $f$  是  $\sqcup$ -连续的当且仅当  $f$  是单调的。

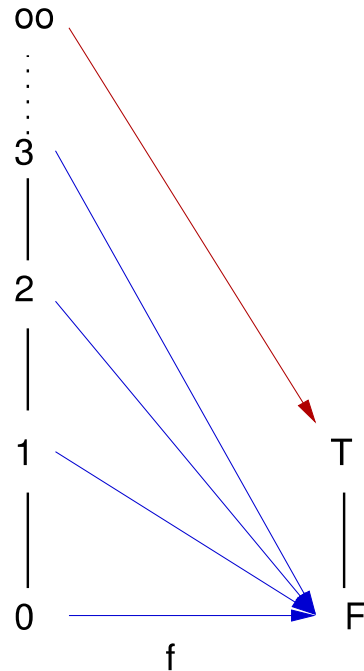
# 函数

- 示例

➤ 考虑如下从  $(\mathbb{N} \cup \{\infty\}, \leq)$  到  $(\{F, T\}, \sqsubseteq)$  的函数，其中  $F \sqsubseteq T$



$f(x)$ : if  $x \geq 2$  then  $T$  else  $F$



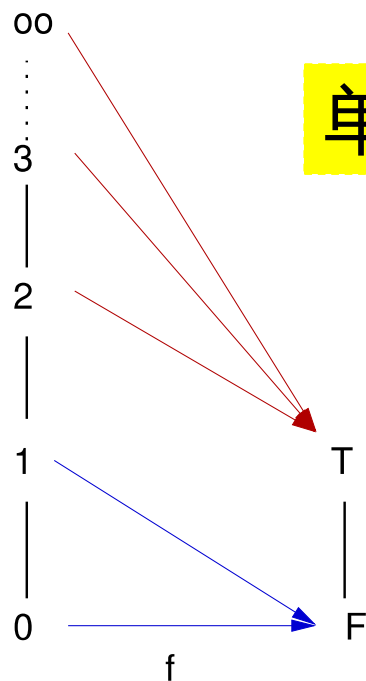
$f(x)$ : if  $x = \infty$  then  $T$  else  $F$



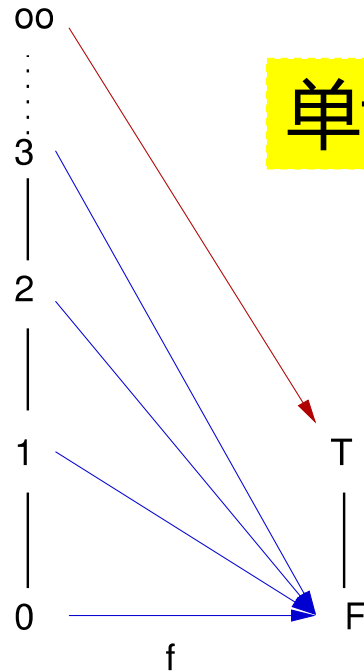
# 函数

- 示例

- 考虑如下从  $(\mathbb{N} \cup \{\infty\}, \leq)$  到  $(\{F, T\}, \sqsubseteq)$  的函数，其中  $F \sqsubseteq T$



$f(x)$ : if  $x \geq 2$  then  $T$  else  $F$



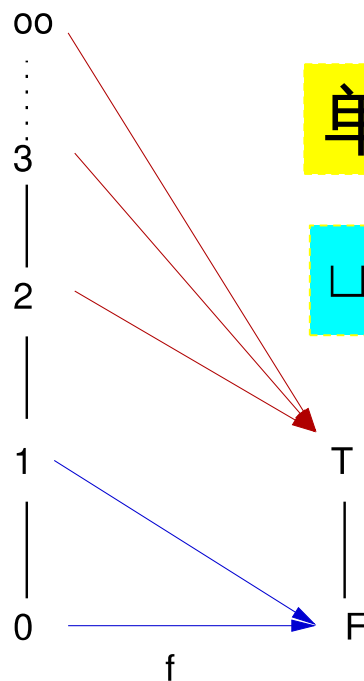
$f(x)$ : if  $x = \infty$  then  $T$  else  $F$



# 函数

- 示例

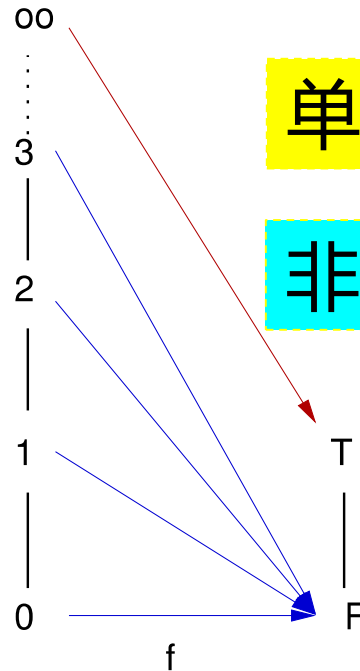
- 考虑如下从  $(\mathbb{N} \cup \{\infty\}, \leq)$  到  $(\{F, T\}, \sqsubseteq)$  的函数，其中  $F \sqsubseteq T$



单调的

$\sqsubseteq$ -连续的

$f(x)$ : if  $x \geq 2$  then  $T$  else  $F$



单调的

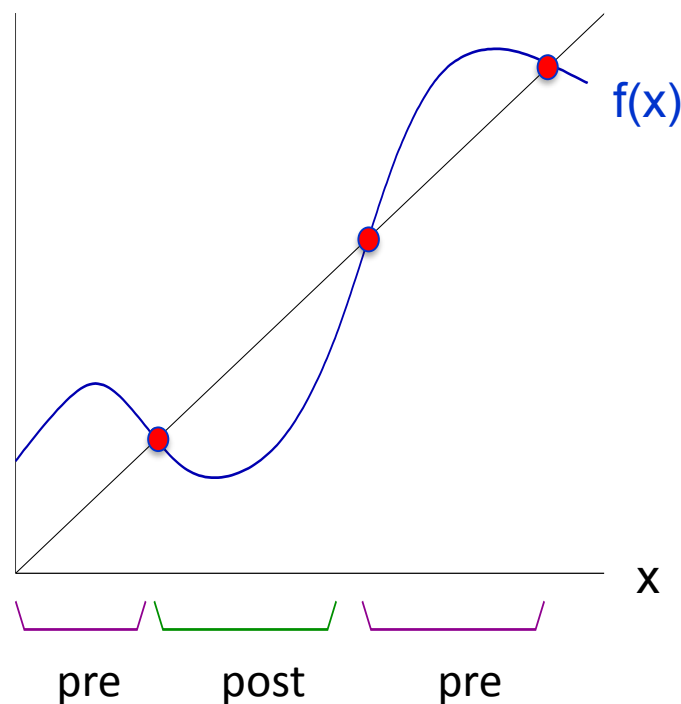
非 $\sqsubseteq$ -连续的

$f(x)$ : if  $x = \infty$  then  $T$  else  $F$



# 不动点理论

- 设 $f$  是偏序集 $(D, \sqsubseteq)$ 上的函数, 则 $D$ 中元素  $x$  称为函数 $f$  的
  - 不动点(fixpoint), 如果  $f(x)=x$
  - 前不动点(pre-fixpoint), 如果  $x \sqsubseteq f(x)$
  - 后不动点(post-fixpoint), 如果  $f(x) \sqsubseteq x$



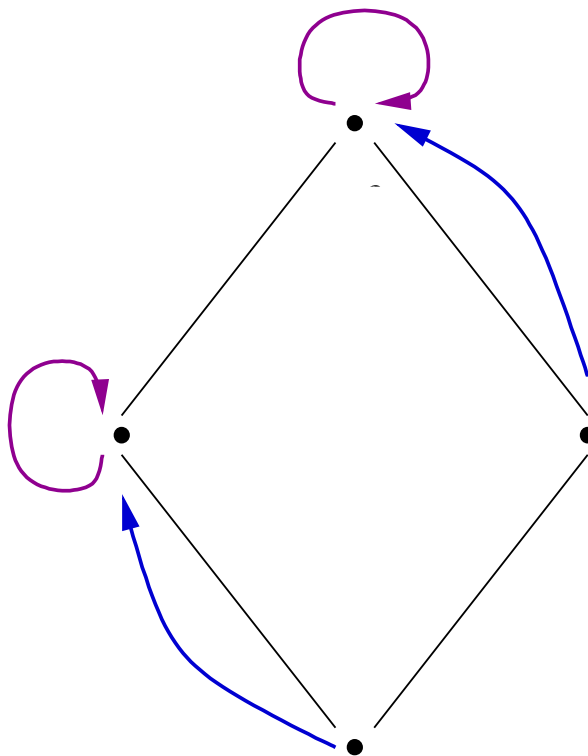
# 不动点理论

- 不动点集合  $\text{fp } f = \{x \in D : f(x) = x\}$
- $x \in \text{fp } f$  称为函数  $f$  的**最小不动点**, 记为  $\text{lfp } f$ , 若对于任意  $y \in \text{fp } f$  皆有  $x \sqsubseteq y$
- $x \in \text{fp } f$  称为函数  $f$  的**最大不动点**, 记为  $\text{gfp } f$ , 若对于任意  $y \in \text{fp } f$  皆有  $y \sqsubseteq x$
- 记  $\text{lfp}_d f$  为关于偏序  $\sqsubseteq$  比  $d$  大的最小不动点
- 记  $\text{gfp}_d f$  为关于偏序  $\sqsubseteq$  比  $d$  小的最大不动点

# 不动点理论

- 示例

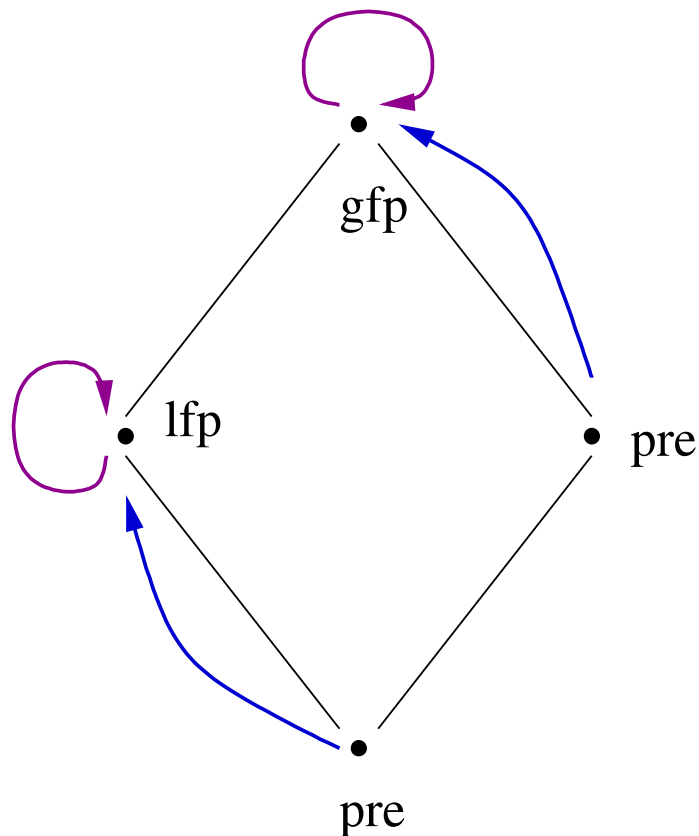
- 函数是否单调?
- 有几个不动点?



# 不动点理论

## ● 示例

- 函数是否单调?
- 有几个不动点?



某单调函数的2个不动点

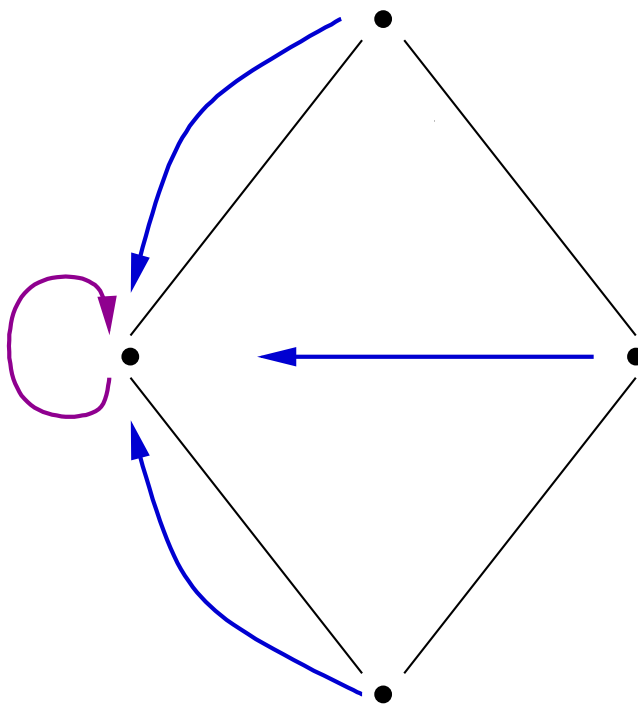




# 不动点理论

- 示例

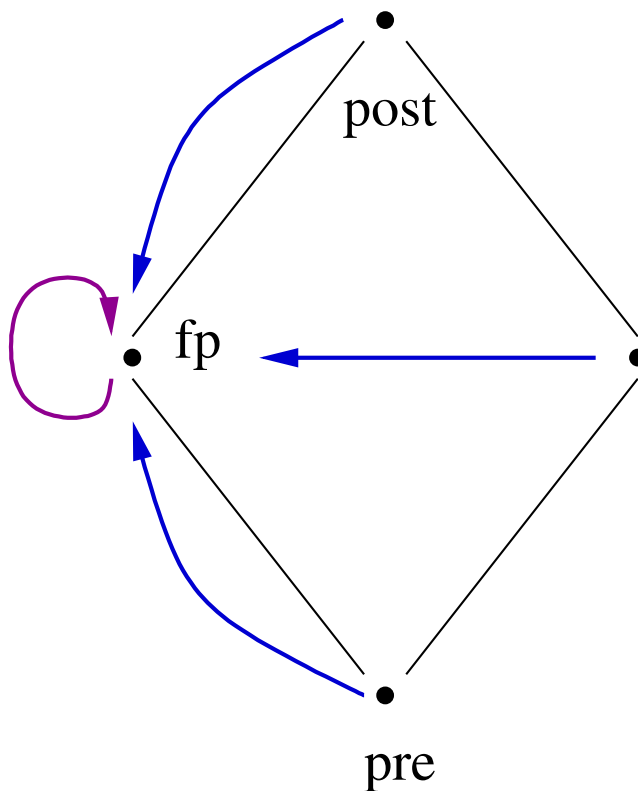
- 函数是否单调?
- 有几个不动点?



# 不动点理论

## ● 示例

- 函数是否单调?
- 有几个不动点?



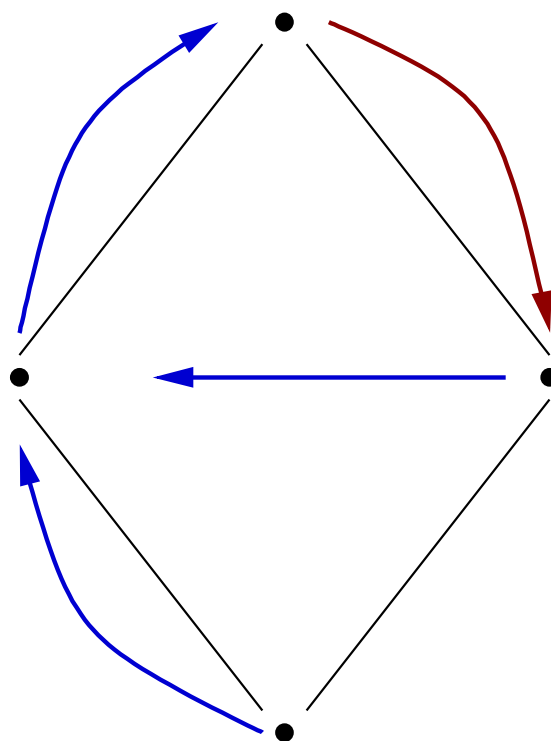
某单调函数的唯一不动点



# 不动点理论

- 示例

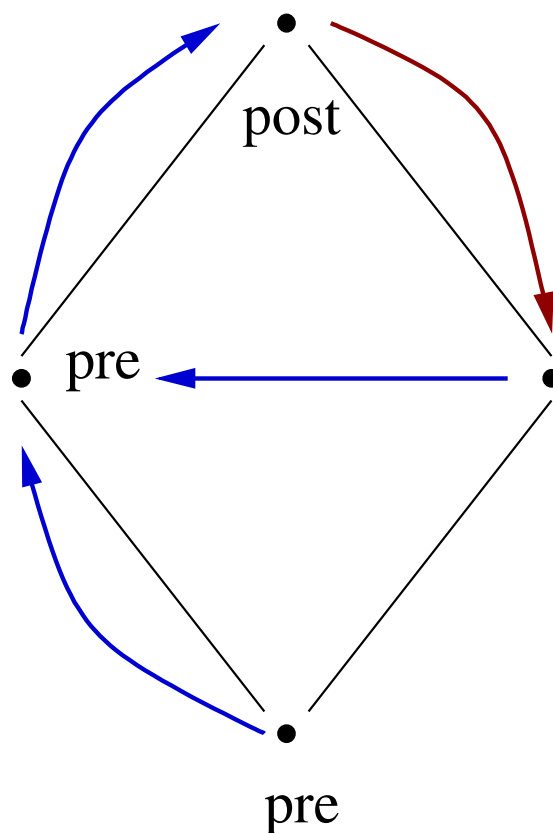
- 函数是否单调?
- 有几个不动点?



# 不动点理论

## ● 示例

- 函数是否单调?
- 有几个不动点?



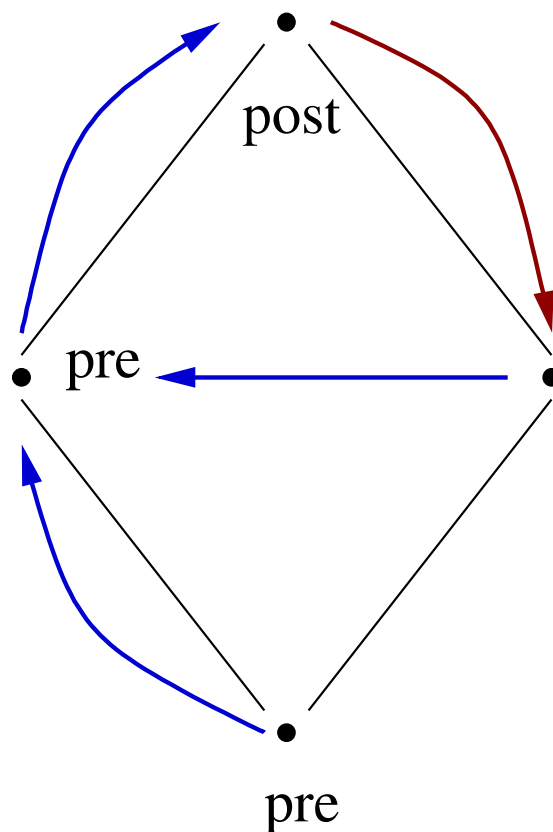
某非单调函数没有不动点



# 不动点理论

## ● 示例

- 函数是否单调?
- 有几个不动点?



某非单调函数没有不动点



# 不动点理论—应用示例

- 表达联立递归方程的解

- 如

$$\begin{cases} x_1 = f(x_1, x_2) \\ x_2 = g(x_1, x_2) \end{cases} \text{ 其中 } x_1, x_2 \text{ 是格 } X \text{ 中的元素}$$

该方程的解可以用格  $X \times X$  上函数  $F$  的不动点来刻画

$$F(x_1, x_2) = (f(x_1, x_2), g(x_1, x_2))$$

最小解即为  $\text{lfp } F$

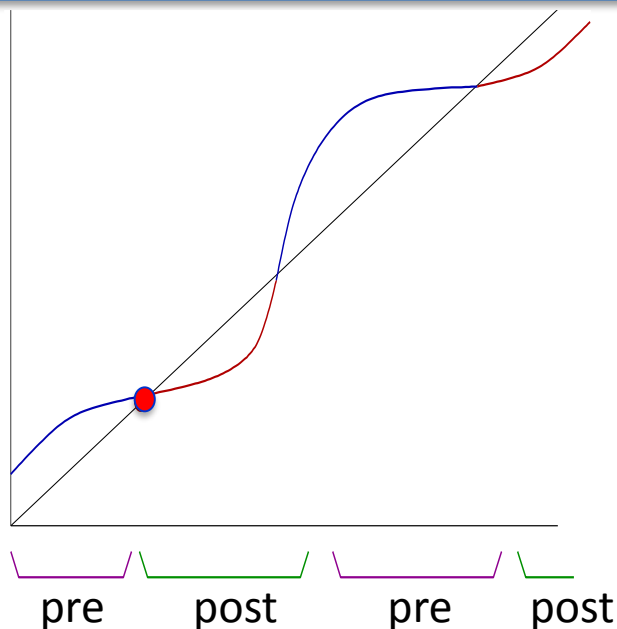
# 不动点理论

## Tarski 不动点定理

完全格  $(D, \sqsubseteq, \sqcup, \sqcap, \perp, \top)$  上单调函数  $f: D \rightarrow D$  的不动点集合构成一个非空完全格, 且

$$\text{lfp } f = \sqcap \text{postfp } f = \sqcap \{x \in D: f(x) \sqsubseteq x\}$$

$$\text{gfp } f = \sqcup \text{prefp } f = \sqcup \{x \in D: f(x) \sqsupseteq x\}$$



# 不动点理论

## Tarski 不动点定理

完全格  $(D, \sqsubseteq, \sqcup, \sqcap, \perp, \top)$  上单调函数  $f : D \rightarrow D$  的不动点集合构成一个非空完全格, 且

$$\text{lfp } f = \sqcap \text{postfp } f = \sqcap \{x \in D : f(x) \sqsubseteq x\}$$

$$\text{gfp } f = \sqcup \text{prefp } f = \sqcup \{x \in D : f(x) \sqsupseteq x\}$$

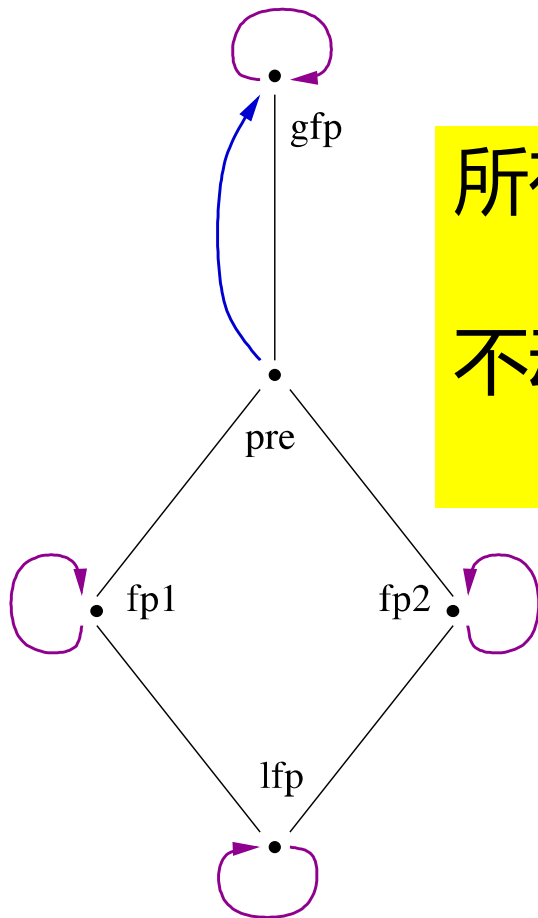
## 不动点构成的完全格

$$(\text{fp } f, \sqsubseteq, \lambda S. \text{lfp}_{\sqcup S} f, \lambda S. \text{gfp}_{\sqcap S} f, \text{lfp } f, \text{gfp } f)$$



# 不动点理论

- Tarski 不动点定理：示例



所有元素构成的格:

$(\{lfp, fp1, fp2, pre, gfp\}, \sqcup, \sqcap, lfp, gfp)$

不动点格:

$(\{lfp, fp1, fp2, gfp\}, \sqcup', \sqcap', lfp, gfp)$

# 不动点理论

## Kleene 不动点定理

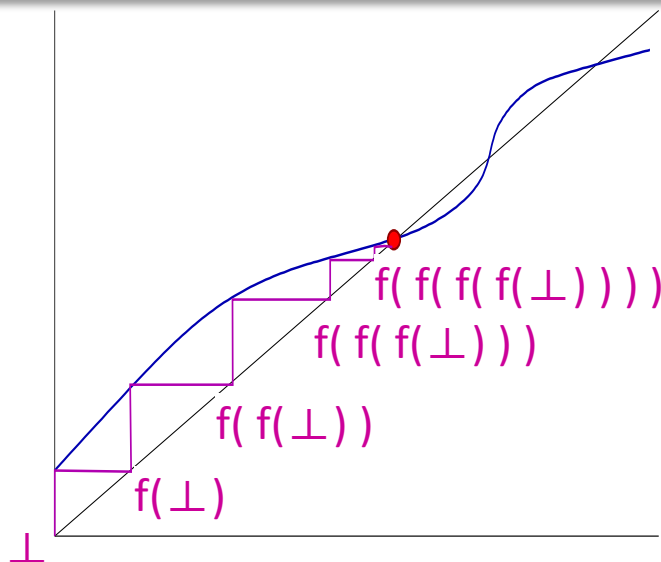
设 $(D, \sqsubseteq)$ 为CPO

- 若函数 $f : D \rightarrow D$  是  $\sqcup$ -连续的, 则有

$$\text{lfp } f = \sqcup \{f^i(\perp) \mid i \in \mathbb{N}\}$$

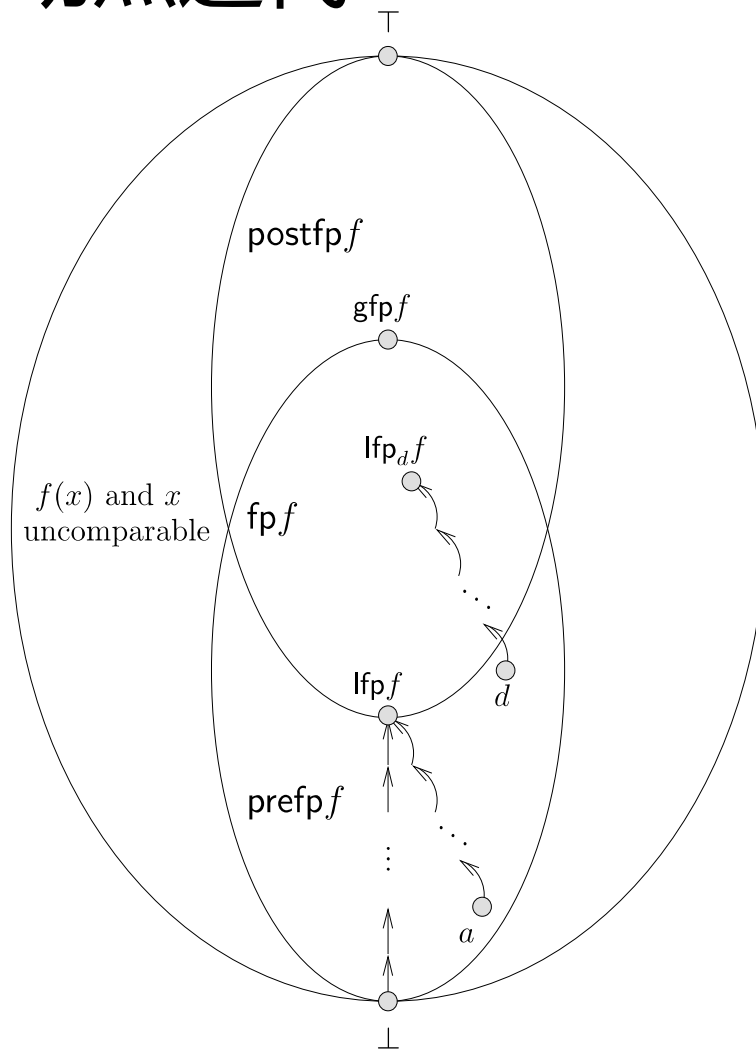
- 若函数 $f : D \rightarrow D$  是  $\sqcap$ -连续的, 则有

$$\text{gfp } f = \sqcap \{f^i(\top) \mid i \in \mathbb{N}\}$$



# 不动点理论

- Kleene不动点迭代

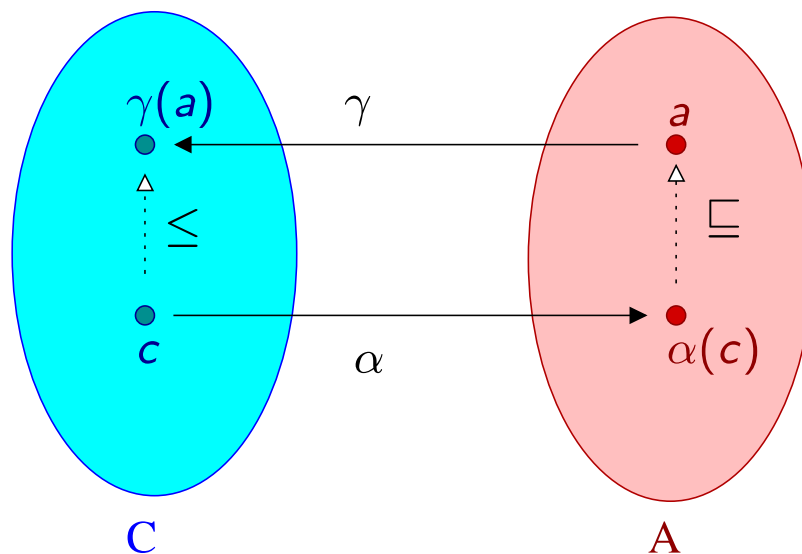


# Galois 连接

- 给定偏序集  $(C, \leq)$  和  $(A, \sqsubseteq)$ , 函数  $\alpha : C \rightarrow A$  及  $\gamma : A \rightarrow C$  构成的函数对  $(\alpha, \gamma)$  称为  $C$  与  $A$  之间的 **Galois 连接**, 当且仅当

$$\forall a \in A, c \in C: \alpha(c) \sqsubseteq a \Leftrightarrow c \leq \gamma(a)$$

$$(C, \leq) \xrightleftharpoons[\alpha]{\gamma} (A, \sqsubseteq)$$

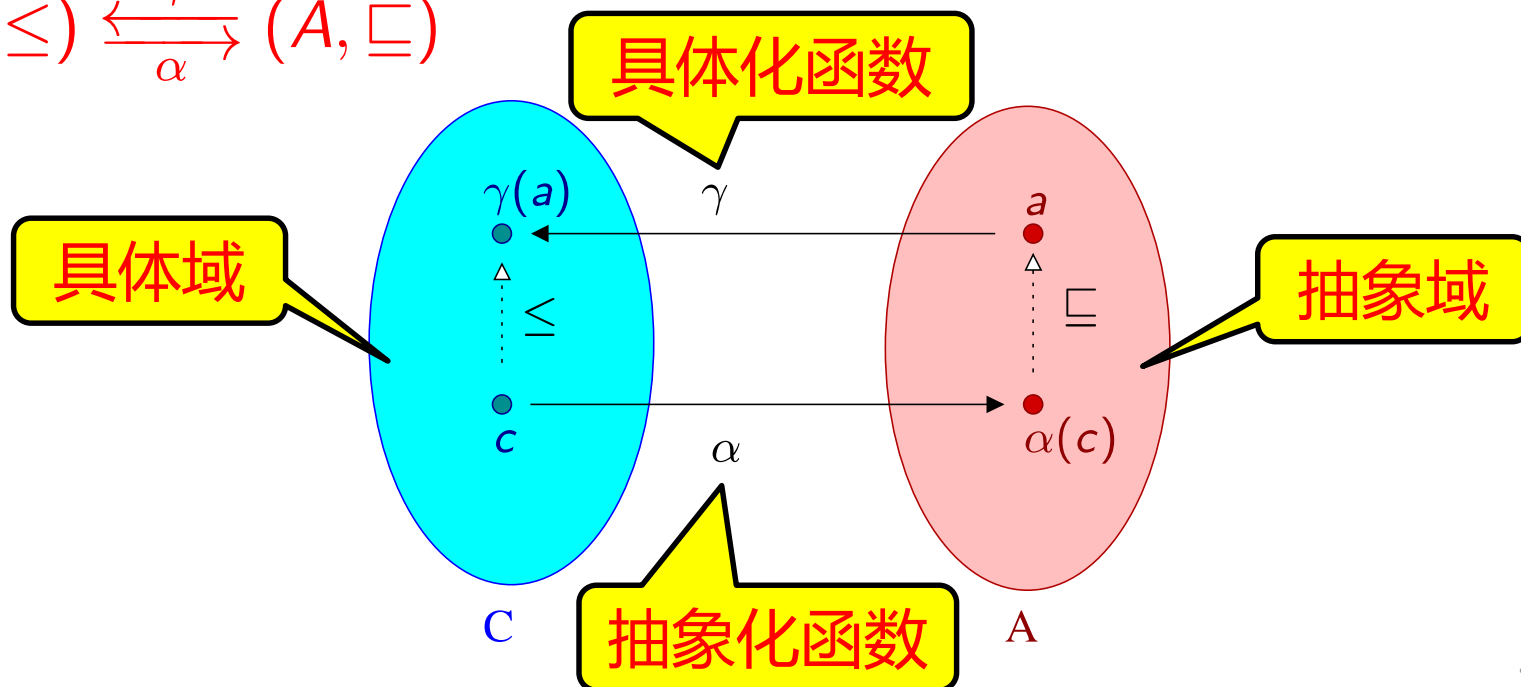


# Galois 连接

- 给定偏序集  $(C, \leq)$  和  $(A, \sqsubseteq)$ , 函数  $\alpha : C \rightarrow A$  及  $\gamma : A \rightarrow C$  构成的函数对  $(\alpha, \gamma)$  称为  $C$  与  $A$  之间的 **Galois 连接**, 当且仅当

$$\forall a \in A, c \in C: \alpha(c) \sqsubseteq a \Leftrightarrow c \leq \gamma(a)$$

$$(C, \leq) \xrightleftharpoons[\alpha]{\gamma} (A, \sqsubseteq)$$



# Galois 连接

- 示例：整数集合 $\mathbb{Z}$ 的区间抽象域

$$(\mathcal{P}(\mathbb{Z}), \subseteq) \xrightleftharpoons[\alpha]{\gamma} (I, \sqsubseteq)$$

$$I \stackrel{\text{def}}{=} (\mathbb{Z} \cup \{-\infty\}) \times (\mathbb{Z} \cup \{+\infty\})$$

$$[a, b] \sqsubseteq [a', b'] \iff a \geq a' \wedge b \leq b'$$



$\alpha$   
 $\gamma$

# Galois 连接

- 示例：整数集合 $\mathbb{Z}$ 的区间抽象域

$$(\mathcal{P}(\mathbb{Z}), \subseteq) \xrightleftharpoons[\alpha]{\gamma} (I, \sqsubseteq)$$

$$I \stackrel{\text{def}}{=} (\mathbb{Z} \cup \{-\infty\}) \times (\mathbb{Z} \cup \{+\infty\})$$

$$[a, b] \sqsubseteq [a', b'] \iff a \geq a' \wedge b \leq b'$$

$$\gamma([a, b]) \stackrel{\text{def}}{=} \{x \in \mathbb{Z} \mid a \leq x \leq b\}$$

$$\alpha(X) \stackrel{\text{def}}{=} [\min X, \max X]$$

# Galois 连接

- 示例：整数集合 $\mathbb{Z}$ 的区间抽象域

$$(\mathcal{P}(\mathbb{Z}), \subseteq) \xrightleftharpoons[\alpha]{\gamma} (I, \sqsubseteq)$$

$(\alpha, \gamma)$  构成了 Galois 连接

$$\begin{aligned} I &\stackrel{\text{def}}{=} (\mathbb{Z} \cup \{-\infty\}) \times (\mathbb{Z} \cup \{+\infty\}) \\ [a, b] \sqsubseteq [a', b'] &\iff a \geq a' \wedge b \leq b' \\ \gamma([a, b]) &\stackrel{\text{def}}{=} \{x \in \mathbb{Z} \mid a \leq x \leq b\} \\ \alpha(X) &\stackrel{\text{def}}{=} [\min X, \max X] \end{aligned}$$

Proof:

$$\alpha(X) \sqsubseteq [a, b]$$

$$\iff \min X \geq a \wedge \max X \leq b$$

$$\iff \forall x \in X: a \leq x \leq b$$

$$\iff \forall x \in X: x \in \{y \mid a \leq y \leq b\}$$

$$\iff \forall x \in X: x \in \gamma([a, b])$$

$$\iff X \subseteq \gamma([a, b])$$



# 近似（抽象）

- 状态抽象  $(C, \leq) \xleftrightarrow[\alpha]{\gamma} (A, \sqsubseteq)$

$$\forall a \in A, c \in C: \alpha(c) \sqsubseteq a \Leftrightarrow c \leq \gamma(a)$$

- $\alpha(c) \sqsubseteq a$  (亦即  $c \leq \gamma(a)$ ) :  $a$  是  $c$  的可靠抽象(或可靠近似)
- $\alpha(x)$  是  $x$  在  $A$  中的最佳抽象 (即最精确的可靠近似)
- $\gamma(a)$  则是在  $C$  中能被  $A$  可靠近似的最不精确的元素

如：

$[0, 10]$  是  $\{0, 1, 2, 5\}$  在整数区间抽象域上的可靠抽象

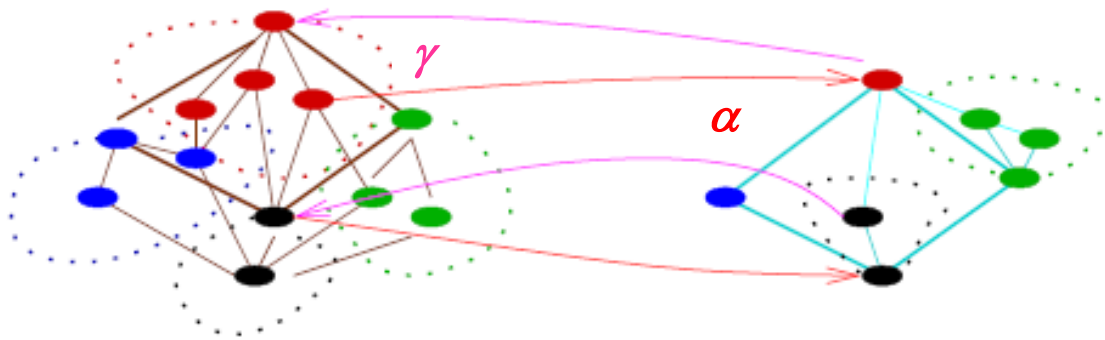
$\alpha(\{0, 1, 2, 5\}) = [0, 5]$  是  $\{0, 1, 2, 5\}$  在整数区间抽象域上的最佳抽象

# 近似（抽象）

- 状态抽象  $(C, \leq) \xleftrightarrow[\alpha]{\gamma} (A, \sqsubseteq)$

$$\forall a \in A, c \in C: \alpha(c) \sqsubseteq a \Leftrightarrow c \leq \gamma(a)$$

- $\alpha(c) \sqsubseteq a$  (亦即  $c \leq \gamma(a)$ ) :  $a$  是  $c$  的可靠抽象(或可靠近似)
- $\alpha(x)$  是  $x$  在  $A$  中的最佳抽象 (即最精确的可靠近似)
- $\gamma(a)$  则是在  $C$  中能被  $A$  可靠近似的最不精确的元素



# 近似（抽象）

## ● 函数抽象 $(C, \leq) \xrightleftharpoons[\alpha]{\gamma} (A, \sqsubseteq)$

设  $f$  是具体域  $C$  上的函数,  $f'$  是抽象域  $A$  上的函数, 则

➤  $f'$  是  $f$  的**可靠抽象**, 当且仅当

$$\forall a, (\alpha \circ f \circ \gamma)(a) \sqsubseteq f'(a), \text{ 即 } \forall a, (f \circ \gamma)(a) \leq (\gamma \circ f')(a)$$

➤  $f'$  是  $f$  的**最佳抽象**, 当且仅当  $f' = \alpha \circ f \circ \gamma$

➤  $f'$  是  $f$  的**精确抽象**, 当且仅当  $\gamma \circ f' = f \circ \gamma$

如：

$\lambda([a,b]).[-\infty, +\infty]$  是函数  $\lambda X.\{x+1 \mid x \in X\}$  在区间抽象域的**可靠抽象**

$\lambda([a,b]).[a+1, b+1]$  是函数  $\lambda X.\{x+1 \mid x \in X\}$  在区间抽象域的**精确抽象**

$g([a,b])=[2a, 2b]$  是函数  $f(X)=\{2x \mid x \in X\}$  在区间抽象域的**最佳抽象**但是  
**不是精确抽象** (因为  $\gamma(g([0,1])) = \{0, 1, 2\}$ ,  $f(\gamma([0,1])) = \{0, 2\}$ )

# 近似（抽象）

- 函数抽象  $(C, \leq) \xrightleftharpoons[\alpha]{\gamma} (A, \sqsubseteq)$

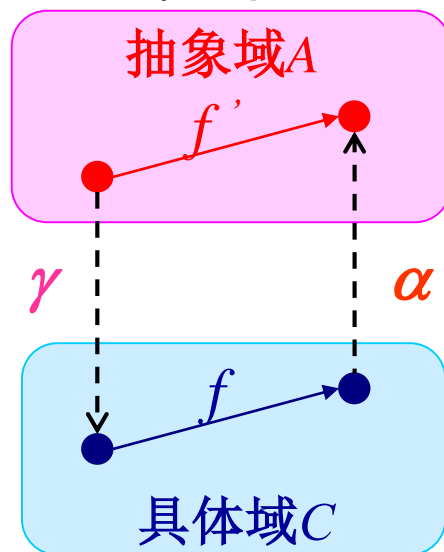
设  $f$  是具体域  $C$  上的函数,  $f'$  是抽象域  $A$  上的函数, 则

➤  $f'$  是  $f$  的可靠抽象, 当且仅当

$$\forall a, (\alpha \circ f \circ \gamma)(a) \sqsubseteq f'(a), \text{ 即 } \forall a, (f \circ \gamma)(a) \leq (\gamma \circ f')(a)$$

➤  $f'$  是  $f$  的最佳抽象, 当且仅当  $f' = \alpha \circ f \circ \gamma$

➤  $f'$  是  $f$  的精确抽象, 当且仅当  $\gamma \circ f' = f \circ \gamma$



# 抽象不动点

- 类似于Tarski不动点定理

## 不动点抽象定理1

给定完全格  $(C, \leq, \vee, \wedge, \perp, \top)$  和  $(A, \sqsubseteq, \sqcup, \sqcap, \perp', \top')$ , 设  $(\alpha, \gamma)$  是两者之间的Galois连接,  $f$  与  $f'$  分别是  $C$  与  $A$  上的单调函数。那么, 若  $f'$  是  $f$  的可靠抽象, 则有

$$\text{lfp}_{\perp} f \leq \gamma(\text{lfp}_{\perp'} f'), \text{ 亦即 } \alpha(\text{lfp}_{\perp} f) \sqsubseteq \text{lfp}_{\perp'} f'$$

在抽象域上所计算得到的最小不动点是可靠的(即, 是具体域上最小不动点的上近似), 进而保证了在抽象域上开展程序分析所得结果的可靠性。

# 抽象不动点

- 类似于Kleene不动点定理

## 不动点抽象定理2

给定CPO  $(C, \leq, \vee, \perp)$  和  $(A, \sqsubseteq, \sqcup, \perp')$ , 设具体化函数  $\gamma: A \rightarrow C$  是单调的,  $f$  与  $f'$  分别是  $C$  与  $A$  上的单调函数。那么, 若  $f'$  是  $f$  的可靠抽象且  $\gamma(\perp')$  是  $f$  的前不动点, 则有

$$\text{lfp}_{\perp} f \leq \text{lfp}_{\gamma(\perp')} f \leq \gamma(\text{lfp}_{\perp} f')$$

- 不要求具体域和抽象域是完全格, 适用于CPO
- 不要求具有Galois连接, 适用于只基于具体化函数的抽象解释框架

# 抽象最小不动点的计算

- 如何在抽象域上有效地计算不动点

## 没有无穷递增链的 Kleene 迭代

若CPO  $(A, \sqsubseteq, \sqcup, \perp')$  上没有无穷递增链,  $f'$  是  $A$  上的单调函数, 且  $a_0$  是  $f'$  的前不动点, 那么Kleene迭代序列  $a_{i+1} = f'(a_i)$  将会在有限时间内收敛于  $\text{lfp}_{a_0} f'$

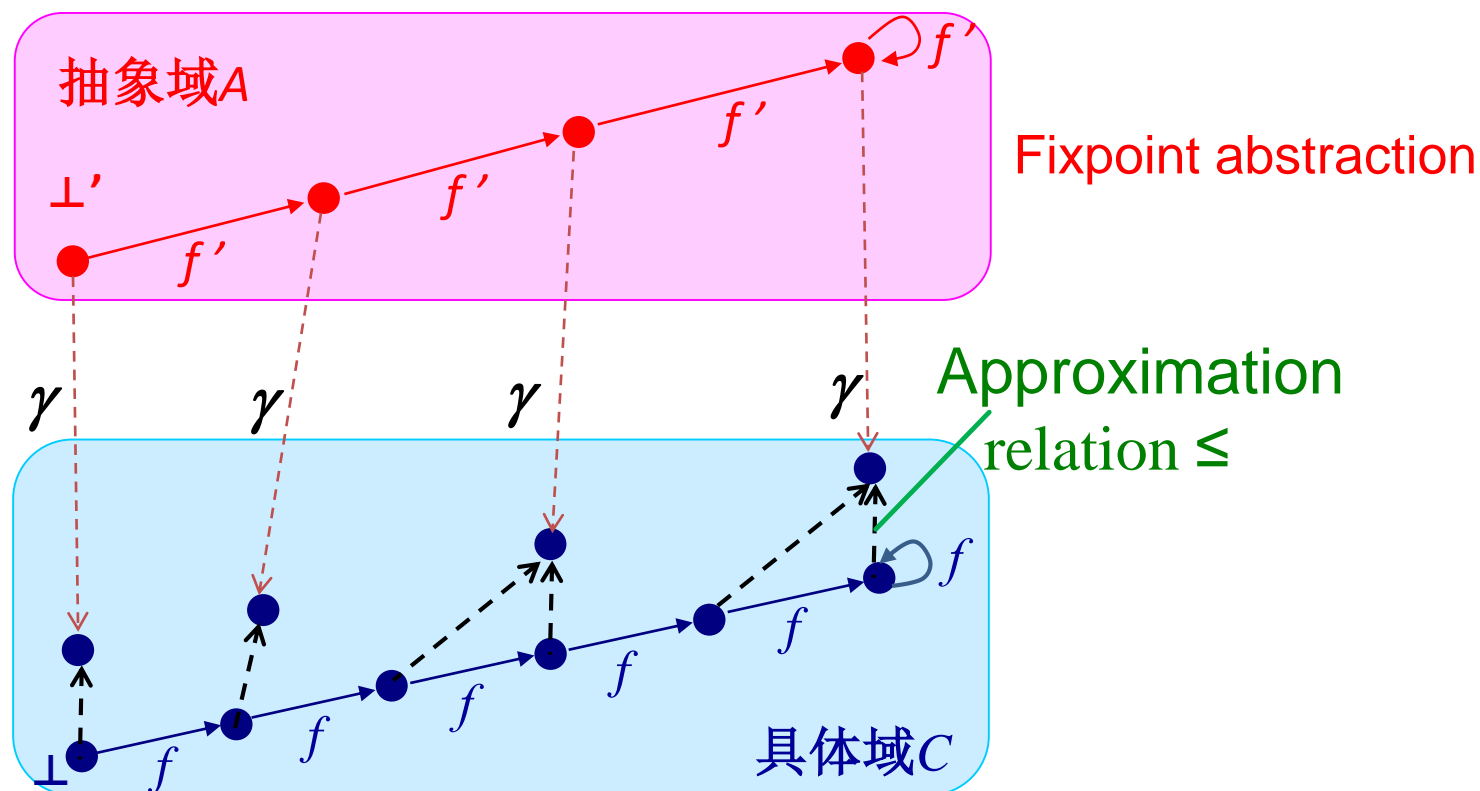
对于

- 有穷抽象域 (即域元素个数是有穷的), 如符号域
- 满足递增链条件的无穷抽象域 (即抽象域对应格的高度是有穷的), 如常量域及仿射等式抽象域

上述定理提供了迭代方法来精确地计算抽象最小不动点, 并且保证了迭代计算的终止性

# 抽象最小不动点的计算

- 本质：在抽象域A中计算函数  $f'$  的最小不动点

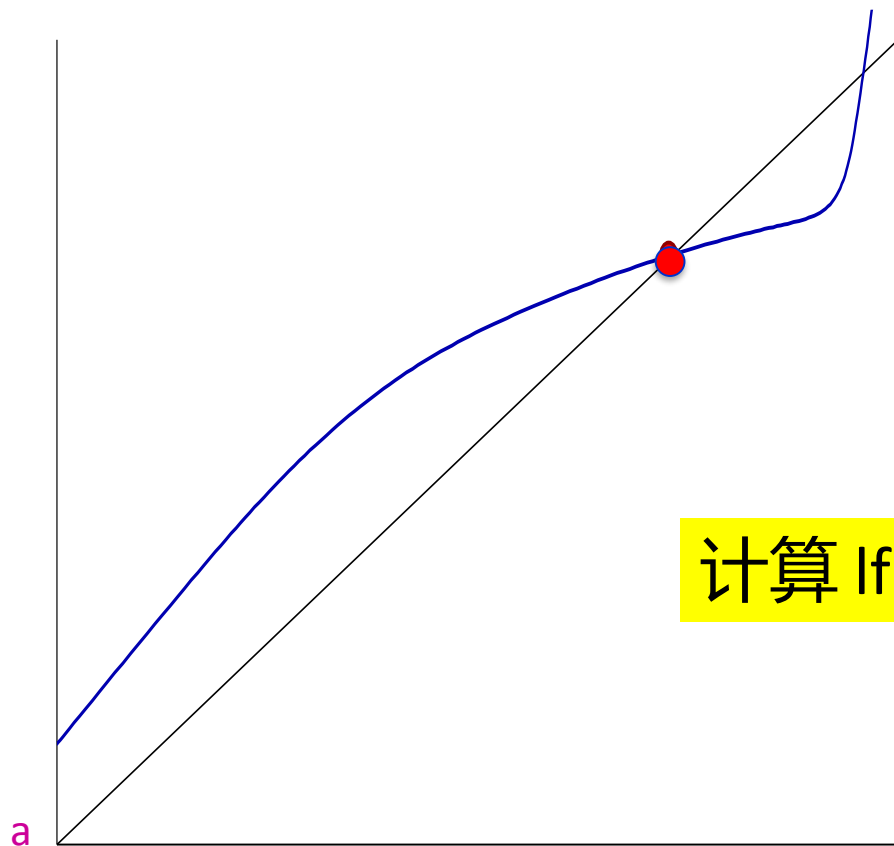


$$f \circ \gamma \leq \gamma \circ f' \Rightarrow \text{lfp}_{\perp} f \leq \gamma(\text{lfp}_{\perp} f')$$



# 抽象最小不动点的计算

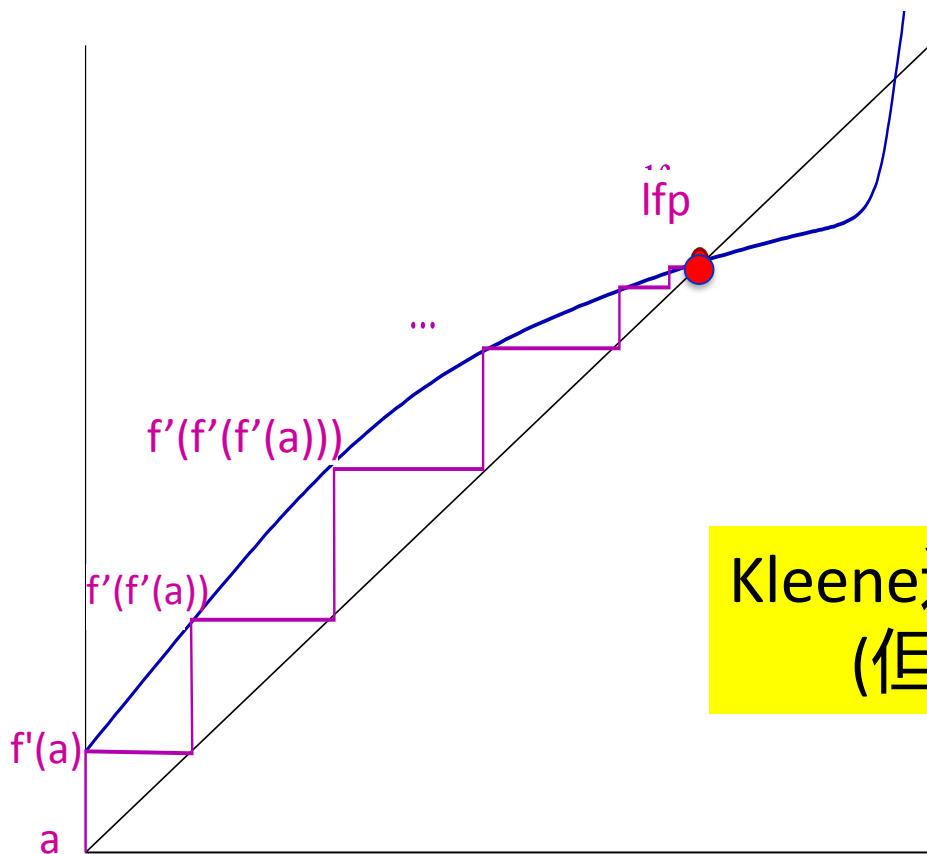
- 有无穷递增链（即抽象域对应格的高度是无穷的，如区间抽象域），怎么办？



计算  $\text{lfp}_a f'$  （如果  $f'$  单调且  $a \sqsubseteq f'(a)$ ）

# 抽象最小不动点的计算

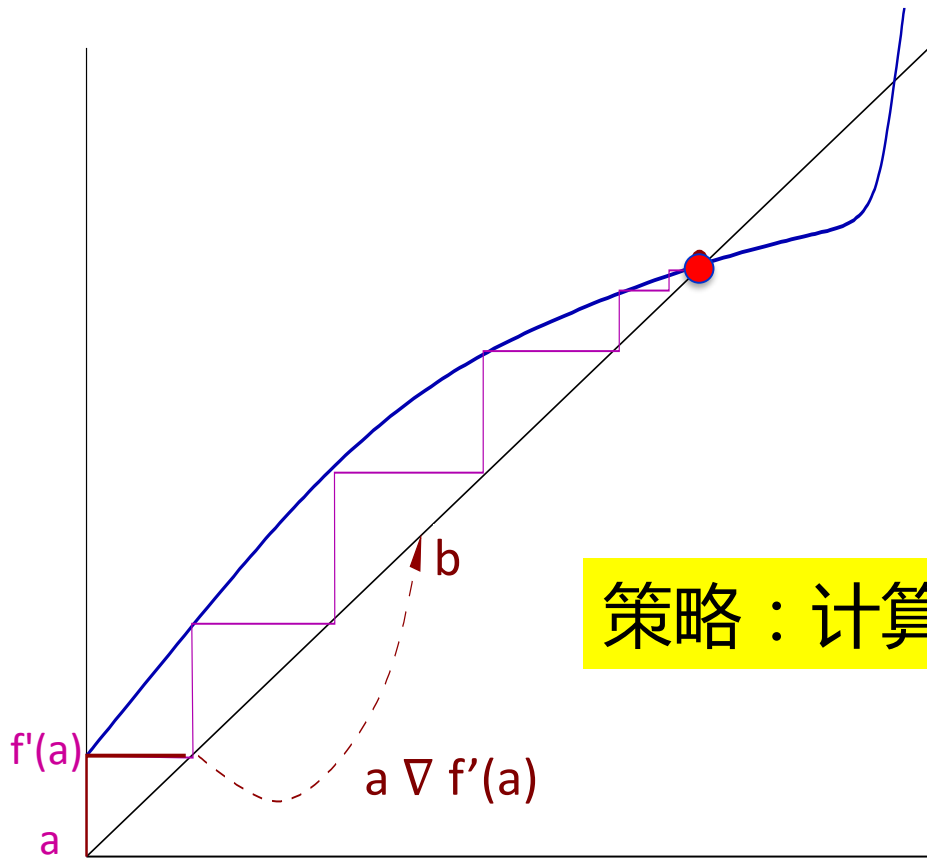
- 有无穷递增链（即抽象域对应格的高度是无穷的，如区间抽象域），怎么办？



Kleene迭代序列 $\{x_n\}$ :  $x_0 \triangleq a$ ,  $x_{n+1} \triangleq f'(x_n)$   
(但是可能存在无穷迭代序列)

# 抽象最小不动点的计算

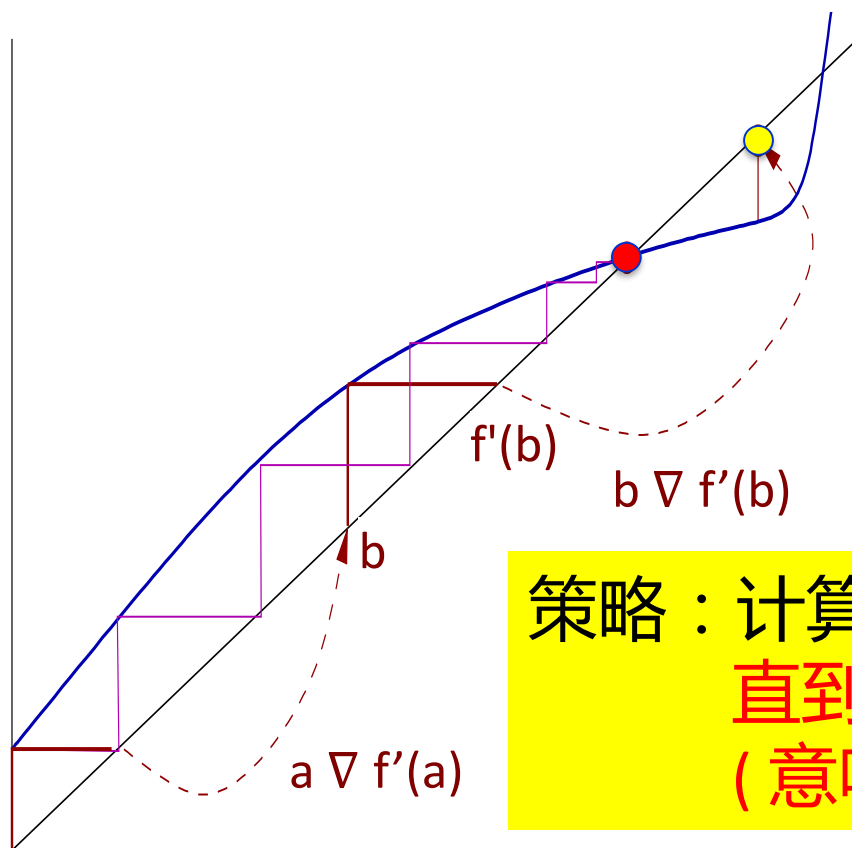
- 有无穷递增链（即抽象域对应格的高度是无穷的，如区间抽象域），怎么办？



策略：计算序列  $\{y_n\}$ :  $y_0 \triangleq a$ ,  $y_{n+1} \triangleq y_n \nabla f'(y_n)$

# 抽象最小不动点的计算

- 有无穷递增链（即抽象域对应格的高度是无穷的，如区间抽象域），怎么办？



策略：计算序列  $\{y_n\}$ :  $y_0 \triangleq a$ ,  $y_{n+1} \triangleq y_n \nabla f'(y_n)$   
直到  $f'(y_n) \sqsubseteq y_n$   
(意味着  $\text{lfp}_a f' \sqsubseteq y_n$ )

# 抽象最小不动点的计算

## ● 加宽算子 (Widening)

➤ 偏序集  $(A, \sqsubseteq)$  上的函数  $\nabla: A \times A \rightarrow A$  称为加宽算子, 当且仅当

–  $\forall x, y \in A, x \sqsubseteq x \nabla y \wedge y \sqsubseteq x \nabla y$ , 且

可靠

– 对于所有链  $(x_i)_{i \in \mathbb{N}}$ , 如下递增链  $(y_i)_{i \in \mathbb{N}}$

$$\begin{cases} y_0 \triangleq x_0 \\ \forall i \in \mathbb{N}, y_{i+1} \triangleq y_i \nabla x_i \end{cases}$$

收敛

将在有穷时间内收敛, 即  $\exists k \in \mathbb{N}, y_{k+1} = y_k$

# 抽象最小不动点的计算

- 加宽算子

- $\forall x, y \in A, x \sqsubseteq x \nabla y \wedge y \sqsubseteq x \nabla y$ , 且
- 对于所有链  $(x_i)_{i \in \mathbb{N}}$ , 如下递增链  $(y_i)_{i \in \mathbb{N}}$

$$\begin{cases} y_0 \triangleq x_0 \\ \forall i \in \mathbb{N}, y_{i+1} \triangleq y_i \nabla x_i \end{cases}$$

将在有穷时间内收敛, 即  $\exists k \in \mathbb{N}, y_{k+1} = y_k$

具体域:  $C \triangleq (\wp(\mathbb{N}), \subseteq, \cup)$

上界抽象域:  $A \triangleq (\mathbb{N} \cup \{\infty\}, \leq, \max)$

$\gamma(x) \triangleq \{a \in \mathbb{N} \mid a < x\}$

$x \nabla y \triangleq \begin{cases} \infty & \text{if } 0 < x < y \\ x & \text{otherwise} \end{cases}$

示例

# 抽象最小不动点的计算

## 基于加宽算子的不动点近似

给定完全格 $(A, \sqsubseteq, \sqcup, \sqcap, \perp', \top')$ , 设函数  $f': A \rightarrow A$  是单调的。那么如下递增链 $(y_i)_{i \in \mathbb{N}}$

$$\begin{cases} y_0 \triangleq \perp' \\ \forall i \in \mathbb{N}, y_{i+1} \triangleq \begin{cases} y_i & \text{if } f'(y_i) \sqsubseteq y_i \\ y_i \sqcup f'(y_i) & \text{otherwise} \end{cases} \end{cases}$$

将在有穷时间内收敛于某个  $y_k$  ( $k \in \mathbb{N}$ ) 且  $y_k$  是  $\text{lfp } f'$  的上近似, 即  $\text{lfp } f' \sqsubseteq y_k$

基于加宽的迭代序列是递增的, 且将在有穷迭代步内收敛于某个  $y_k$ ; 并且  $y_k$  是  $f'$  的后不动点, 从而是最小不动点  $\text{lfp } f$  的上近似

# 抽象最小不动点的计算

## 基于加宽算子的不动点近似的可靠性

给定完全格 $(C, \leq, \vee, \wedge, \perp, \top)$ 和 $(A, \sqsubseteq, \sqcup, \sqcap, \perp', \top')$ , 设具体化函数 $\gamma: A \rightarrow C$ 是单调的, 函数 $f: C \rightarrow C$ 是单调的, 函数 $f': A \rightarrow A$ 是 $f$ 的可靠抽象,  $x \in A$ 是 $\gamma(x)$ 是 $f$ 的前不动点。那么如下链 $(y_i)_{i \in \mathbb{N}}$

$$\left\{ \begin{array}{l} y_0 \triangleq x \\ \forall i \in \mathbb{N}, y_{i+1} \triangleq \begin{cases} y_i & \text{if } f'(y_i) \sqsubseteq y_i \\ y_i \nabla f'(y_i) & \text{otherwise} \end{cases} \end{array} \right.$$

将在有穷时间内收敛于某个  $y_k (k \in \mathbb{N})$  且  $y_k$  是  $\text{lfp}_{\gamma(x)} f$  的上近似, 即  $\text{lfp}_{\gamma(x)} f \sqsubseteq \gamma(y_k)$



# 抽象最小不动点的计算

## ● 变窄算子 (Narrowing)

- 偏序集  $(A, \sqsubseteq)$  上的函数  $\Delta : A \times A \rightarrow A$  称为加宽算子, 当且仅当

–  $\forall x, y \in A, y \sqsubseteq x \Rightarrow y \sqsubseteq (x \Delta y) \sqsubseteq x$ , 且

两次迭代  
结果之间

– 对于任意递减链  $(x_i)_{i \in \mathbb{N}}$ , 如下递减链  $(y_i)_{i \in \mathbb{N}}$

$$\begin{cases} y_0 \triangleq x_0 \\ \forall i \in \mathbb{N}, y_{i+1} \triangleq y_i \Delta x_{i+1} \end{cases}$$

收敛

将在有穷时间内收敛, 即  $\exists k \in \mathbb{N}, y_{k+1} = y_k$

# 抽象最小不动点的计算

- 变窄算子

- $\forall x, y \in A, y \sqsubseteq x \Rightarrow y \sqsubseteq (x \Delta y) \sqsubseteq x$ , 且
- 对于任意递减链  $(x_i)_{i \in \mathbb{N}}$ , 如下递减链  $(y_i)_{i \in \mathbb{N}}$

$$\begin{cases} y_0 \triangleq x_0 \\ \forall i \in \mathbb{N}, y_{i+1} \triangleq y_i \Delta x_{i+1} \end{cases}$$

将在有穷时间内收敛, 即  $\exists k \in \mathbb{N}, y_{k+1} = y_k$

具体域:  $C \triangleq (\wp(\mathbb{N}), \subseteq, \cup)$

上界抽象域:  $A \triangleq (\mathbb{N} \cup \{\infty\}, \leq, \max)$

$\gamma(x) \triangleq \{a \in \mathbb{N} \mid a < x\}$

$x \Delta y \triangleq \begin{cases} y & \text{if } x = \infty \\ x & \text{otherwise} \end{cases} \quad (\text{其中 } y \sqsubseteq x)$

示例

# 抽象最小不动点的计算

## 基于变窄算子的不动点精化

设  $(A, \sqsubseteq)$  为偏序集, 设函数  $f': A \rightarrow A$  是单调的,  $x, y \in A$  且  $x \sqsubseteq y$ ,  $x$  是  $f'$  的不动点,  $y$  是  $f'$  的后不动点。那么如下递减链  $(z_i)_{i \in \mathbb{N}}$

$$\begin{cases} z_0 \triangleq y \\ \forall i \in \mathbb{N}, z_{i+1} \triangleq \begin{cases} z_i & \text{if } z_i \sqsubseteq f'(z_i) \\ z_i \Delta f'(z_i) & \text{otherwise} \end{cases} \end{cases}$$

将在有穷时间内收敛于某个  $z_k (k \in \mathbb{N})$  且  $x \sqsubseteq z_k \sqsubseteq y$ 。

基于变窄的迭代可以在任何迭代步停下来, 因为所有基于变窄的迭代的中间结果都比不动点  $x$  大, 因此中间结果也都是可靠的。等到迭代稳定下来, 只是为了得到最好的精度。

# 抽象最小不动点的计算

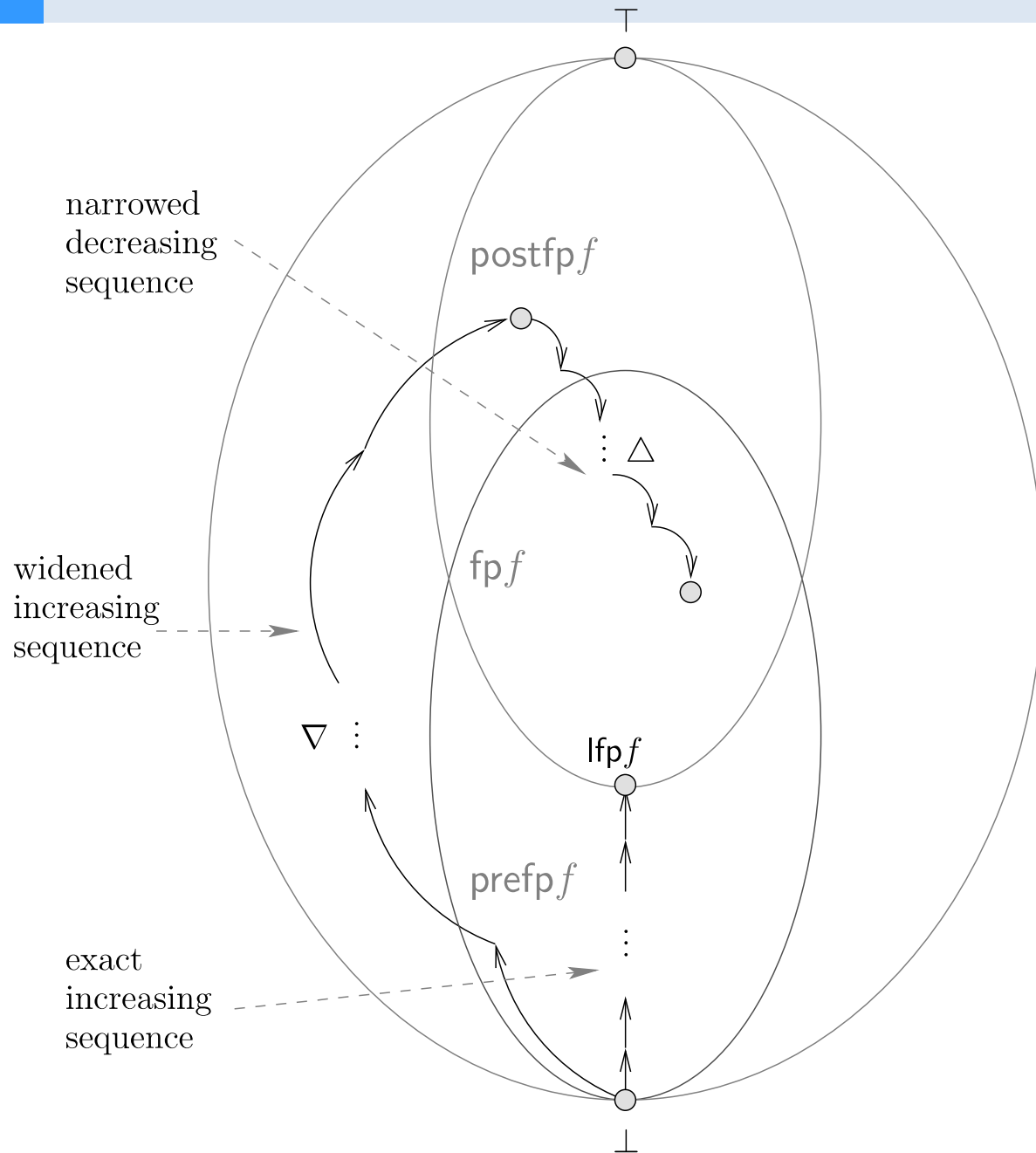
## 基于变窄算子的不动点精化的可靠性

给定完全格 $(C, \leq, \vee, \wedge, \perp, \top)$ 和 $(A, \sqsubseteq, \sqcup, \sqcap, \perp', \top')$ , 设具体化函数 $\gamma: A \rightarrow C$ 是单调的, 函数 $f: C \rightarrow C$ 是单调的, 函数 $f': A \rightarrow A$ 是 $f$ 的可靠抽象,  $x, y \in A$ 且 $\text{lfp}_{\gamma(x)} f \sqsubseteq \gamma(y)$ 。那么如下链 $(z_i)_{i \in \mathbb{N}}$

$$\begin{cases} z_0 \triangleq y \\ \forall i \in \mathbb{N}, z_{i+1} \triangleq \begin{cases} z_i & \text{if } z_i \sqsubseteq f'(z_i) \\ z_i \Delta f'(z_i) & \text{otherwise} \end{cases} \end{cases}$$

将在有穷时间内收敛于某个  $z_k (k \in \mathbb{N})$  且  $z_k$  是  $\text{lfp}_{\gamma(x)} f$  的上近似, 即  $\text{lfp}_{\gamma(x)} f \sqsubseteq \gamma(z_k) \sqsubseteq \gamma(y)$

## 基于加宽/变窄的不动点迭代



**谢谢！**