Categorizing Ensembles of Real-Valued Functions

Florian Bruse[2]   Martin Lange[1]   Sören Möller[1]

[1]Universität Kassel, Germany

[2]TU Munich, Germany

Theorietag 2025
3/4 March, 2025

Motivation

background: science classes at secondary education level

- students obtain some basic knowledge on natural phenomenon (e.g., yeast growth, electrical currents, . . .), including some data points

Motivation

background: science classes at secondary education level

- students obtain some basic knowledge on natural phenomenon (e.g., yeast growth, electrical currents, . . .), including some data points
- students formulate a hypothesis on further behavior of phenomenon
  - increased temperature entails increased yeast growth

Motivation

background: science classes at secondary education level

- students obtain some basic knowledge on natural phenomenon (e.g., yeast growth, electrical currents, ...), including some data points
- students formulate a hypothesis on further behavior of phenomenon
  - increased temperature entails increased yeast growth
  - x voltage leads to y current

Motivation

background: science classes at secondary education level

- students obtain some basic knowledge on natural phenomenon (e.g., yeast growth, electrical currents, . . .), including some data points
- students formulate a hypothesis on further behavior of phenomenon
  - increased temperature entails increased yeast growth
  - x voltage leads to y current
  - light intensity is constant between 11 am and 1 pm
- students validate hypothesis via experiment, go back to previous step if this fails

Motivation

background: science classes at secondary education level

- students obtain some basic knowledge on natural phenomenon (e.g., yeast growth, electrical currents, . . .), including some data points
- students formulate a hypothesis on further behavior of phenomenon
  - increased temperature entails increased yeast growth
  - x voltage leads to y current
  - light intensity is constant between 11 am and 1 pm
- students validate hypothesis via experiment, go back to previous step if this fails

purpose of project is digital reproduction of this process

reason: some experiments too slow/dangerous/expensive/. . . to do in class

Motivation

background: science classes at secondary education level

- students obtain some basic knowledge on natural phenomenon (e.g., yeast growth, electrical currents, ...), including some data points
- students formulate a hypothesis on further behavior of phenomenon
  - increased temperature entails increased yeast growth
  - x voltage leads to y current
  - light intensity is constant between 11 am and 1 pm
- students validate hypothesis via experiment, go back to previous step if this fails

purpose of project is digital reproduction of this process

reason: some experiments too slow/dangerous/expensive/... to do in class

requires:

- formalization of making statements about natural phenomena
- efficient decidability of logical entailment of such statements

Influences

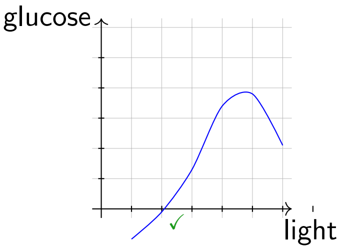
fix (partially ordered) set of variables $\mathcal{V}$, e.g., $\mathsf{temp}, \mathsf{yeastAct}, \mathsf{volt}, \ldots$

influence experiment is set of continuous functions $\mathcal{F} : \mathcal{V} \times \mathcal{V} \rightharpoonup (\mathbb{R} \rightharpoonup \mathbb{R})$ s.t.

Influences

fix (partially ordered) set of variables $\mathcal{V}$, e.g., $\mathsf{temp}, \mathsf{yeastAct}, \mathsf{volt}, \ldots$

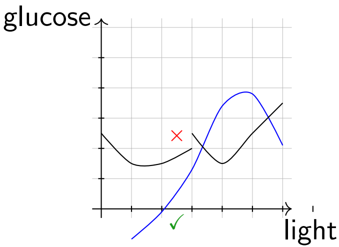
influence experiment is set of continuous functions $\mathcal{F} : \mathcal{V} \times \mathcal{V} \rightharpoonup (\mathbb{R} \rightharpoonup \mathbb{R})$ s.t.



Influences

fix (partially ordered) set of variables $\mathcal{V}$, e.g., $\mathsf{temp}, \mathsf{yeastAct}, \mathsf{volt}, \ldots$

influence experiment is set of continuous functions $\mathcal{F} : \mathcal{V} \times \mathcal{V} \rightharpoonup (\mathbb{R} \rightharpoonup \mathbb{R})$ s.t.

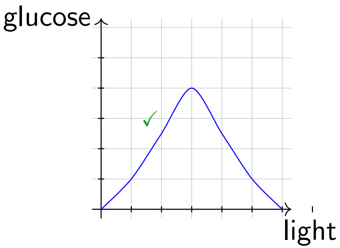


Influences

fix (partially ordered) set of variables $\mathcal{V}$, e.g., $\mathsf{temp}, \mathsf{yeastAct}, \mathsf{volt}, \ldots$

influence experiment is set of continuous functions $\mathcal{F} : \mathcal{V} \times \mathcal{V} \rightharpoonup (\mathbb{R} \rightharpoonup \mathbb{R})$ s.t.

- domain of all functions is interval

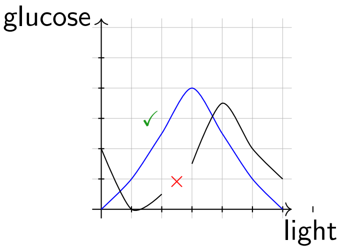


Influences

fix (partially ordered) set of variables $\mathcal{V}$, e.g., $\mathsf{temp}, \mathsf{yeastAct}, \mathsf{volt}, \ldots$

influence experiment is set of continuous functions $\mathcal{F} : \mathcal{V} \times \mathcal{V} \rightharpoonup (\mathbb{R} \rightharpoonup \mathbb{R})$ s.t.

- domain of all functions is interval

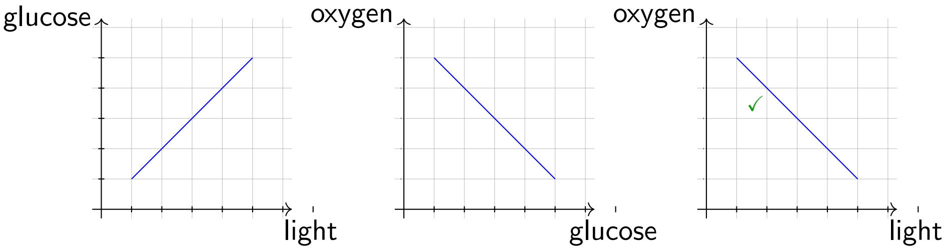


Influences

fix (partially ordered) set of variables $\mathcal{V}$, e.g., $\mathsf{temp}, \mathsf{yeastAct}, \mathsf{volt}, \ldots$

influence experiment is set of continuous functions $\mathcal{F} : \mathcal{V} \times \mathcal{V} \rightharpoonup (\mathbb{R} \rightharpoonup \mathbb{R})$ s.t.

- domain of all functions is interval
- $\mathcal{F}_{a,c}(x) = \mathcal{F}_{b,c}(\mathcal{F}_{a,b}(x))$ for all $a, b, c \in \mathcal{V}$, $x \in \mathbb{R}$ (coherence property)

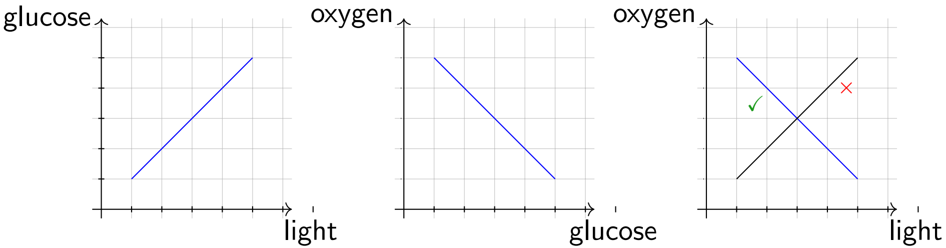


Influences

fix (partially ordered) set of variables $\mathcal{V}$, e.g., $\mathsf{temp}, \mathsf{yeastAct}, \mathsf{volt}, \ldots$

influence experiment is set of continuous functions $\mathcal{F} : \mathcal{V} \times \mathcal{V} \rightharpoonup (\mathbb{R} \rightharpoonup \mathbb{R})$ s.t.

- domain of all functions is interval
- $\mathcal{F}_{a,c}(x) = \mathcal{F}_{b,c}(\mathcal{F}_{a,b}(x))$ for all $a, b, c \in \mathcal{V}$, $x \in \mathbb{R}$ (coherence property)

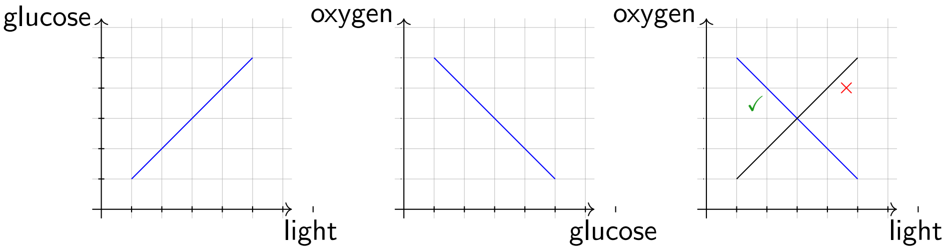


Influences

fix (partially ordered) set of variables $\mathcal{V}$, e.g., temp, yeastAct, volt, ...

influence experiment is set of continuous functions $\mathcal{F} : \mathcal{V} \times \mathcal{V} \rightharpoonup (\mathbb{R} \rightharpoonup \mathbb{R})$ s.t.

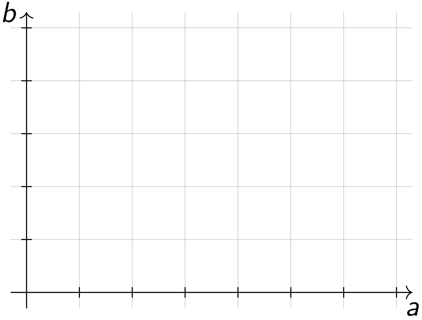
- domain of all functions is interval
- $\mathcal{F}_{a,c}(x) = \mathcal{F}_{b,c}(\mathcal{F}_{a,b}(x))$ for all $a, b, c \in \mathcal{V}$, $x \in \mathbb{R}$ (coherence property)
- functions defined only alongside variable order



Statements

influence statement S is $a \xhookrightarrow{I\,q\,I'} b$ where

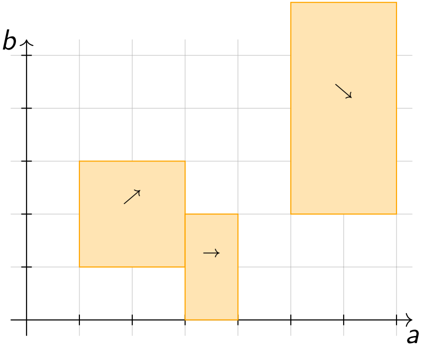
- a, b are variables
- I, I' are closed intervals with bounds in $\mathbb{Q}$
- q is a behaviour from $\{\nearrow, \searrow, \rightarrow, \rightsquigarrow\}$



Statements

influence statement S is $a \xhookrightarrow{I\,q\,I'} b$ where

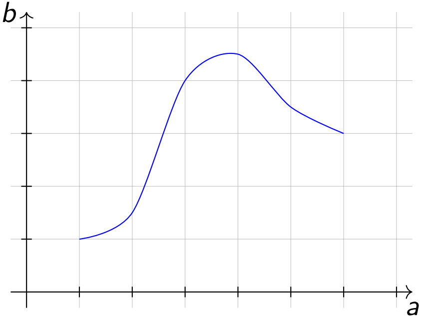
- a, b are variables
- I, I' are closed intervals with bounds in $\mathbb{Q}$
- q is a behaviour from $\{\nearrow, \searrow, \rightarrow, \rightsquigarrow\}$



Statements

influence statement S is $a \xhookrightarrow{I\, q\, I'} b$ where

- a, b are variables
- I, I' are closed intervals with bounds in $\mathbb{Q}$
- q is a behaviour from $\{\nearrow, \searrow, \rightarrow, \rightsquigarrow\}$



semantics: $\mathcal{F} \models S$ (for $S = a \xhookrightarrow{I\, q\, I'} b$) iff

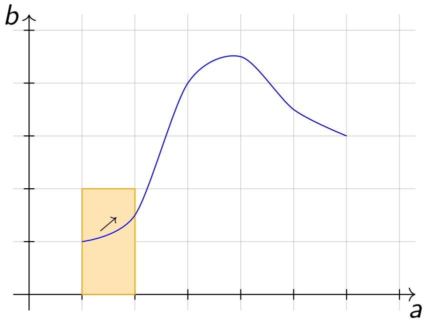
- $\mathcal{F}_{a,b}(x) \in I'$ for all $x \in I$, and
- $\mathcal{F}_{a,b}$ "displays behaviour" q on I

examples:

Statements

influence statement S is $a \xhookrightarrow{I\, q\, I'} b$ where

- a, b are variables
- I, I' are closed intervals with bounds in $\mathbb{Q}$
- q is a behaviour from $\{\nearrow, \searrow, \rightarrow, \rightsquigarrow\}$



semantics: $\mathcal{F} \models S$ (for $S = a \xhookrightarrow{I\, q\, I'} b$) iff

- $\mathcal{F}_{a,b}(x) \in I'$ for all $x \in I$, and
- $\mathcal{F}_{a,b}$ "displays behaviour" q on I

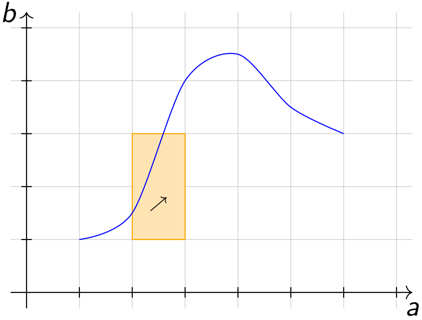
examples:

- $\mathcal{F} \models a \xhookrightarrow{[1,2] \nearrow [0,2]} b$

Statements

influence statement S is $a \xhookrightarrow{I\, q\, I'} b$ where

- a, b are variables
- I, I' are closed intervals with bounds in $\mathbb{Q}$
- q is a behaviour from $\{\nearrow, \searrow, \rightarrow, \rightsquigarrow\}$



semantics: $\mathcal{F} \models S$ (for $S = a \xhookrightarrow{I\, q\, I'} b$) iff

- $\mathcal{F}_{a,b}(x) \in I'$ for all $x \in I$, and
- $\mathcal{F}_{a,b}$ "displays behaviour" q on I

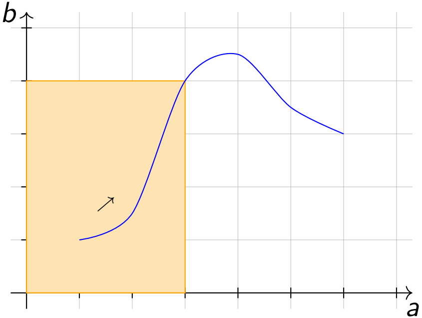
examples:

- $\mathcal{F} \models a \xhookrightarrow{[1,2]\ \nearrow\ [0,2]} b$
- $\mathcal{F} \not\models a \xhookrightarrow{[2,3]\ \nearrow\ [1,3]} b$

Statements

influence statement S is $a \xhookrightarrow{I\,q\,I'} b$ where

- a, b are variables
- I, I' are closed intervals with bounds in $\mathbb{Q}$
- q is a behaviour from $\{\nearrow, \searrow, \rightarrow, \rightsquigarrow\}$



semantics: $\mathcal{F} \models S$ (for $S = a \xhookrightarrow{I\,q\,I'} b$) iff

- $\mathcal{F}_{a,b}(x) \in I'$ for all $x \in I$, and
- $\mathcal{F}_{a,b}$ "displays behaviour" q on I

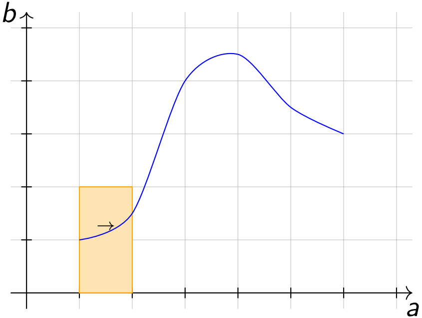
examples:

- $\mathcal{F} \models a \xhookrightarrow{[1,2] \nearrow [0,2]} b$
- $\mathcal{F} \not\models a \xhookrightarrow{[2,3] \nearrow [1,3]} b$
- $\mathcal{F} \not\models a \xhookrightarrow{[0,3] \rightsquigarrow [0,4]} b$

Statements

influence statement S is $a \xhookrightarrow{I\, q\, I'} b$ where

- a, b are variables
- I, I' are closed intervals with bounds in $\mathbb{Q}$
- q is a behaviour from $\{\nearrow, \searrow, \rightarrow, \rightsquigarrow\}$



semantics: $\mathcal{F} \models S$ (for $S = a \xhookrightarrow{I\, q\, I'} b$) iff

- $\mathcal{F}_{a,b}(x) \in I'$ for all $x \in I$, and
- $\mathcal{F}_{a,b}$ "displays behaviour" q on I

examples:

- $\mathcal{F} \models a \xhookrightarrow{[1,2] \nearrow [0,2]} b$
- $\mathcal{F} \not\models a \xhookrightarrow{[2,3] \nearrow [1,3]} b$
- $\mathcal{F} \not\models a \xhookrightarrow{[0,3] \rightsquigarrow [0,4]} b$
- $\mathcal{F} \not\models a \xhookrightarrow{[1,2] \rightarrow [0,2]} b$

Schemes and Hypotheses

influence scheme $\mathcal{C}$ = set of influence statements

semantics: $\mathcal{F} \models \mathcal{C}$ if $\mathcal{F} \models S$ f.a. $S \in \mathcal{C}$

Schemes and Hypotheses

influence scheme $\mathcal{C}$ = set of influence statements

semantics: $\mathcal{F} \models \mathcal{C}$ if $\mathcal{F} \models S$ f.a. $S \in \mathcal{C}$

hypothesis H is a statement

H follows from scheme $\mathcal{C}$, written $\mathcal{C} \models H$ if $\mathcal{F} \models \mathcal{C}$ entails $\mathcal{F} \models H$ f.a. $\mathcal{F}$

Schemes and Hypotheses

influence scheme $\mathcal{C}$ = set of influence statements

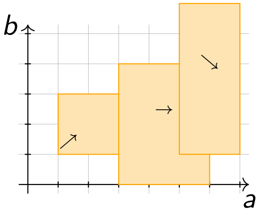
semantics: $\mathcal{F} \models \mathcal{C}$ if $\mathcal{F} \models S$ f.a. $S \in \mathcal{C}$

hypothesis H is a statement

H follows from scheme $\mathcal{C}$, written $\mathcal{C} \models H$ if $\mathcal{F} \models \mathcal{C}$ entails $\mathcal{F} \models H$ f.a. $\mathcal{F}$

consider scheme $\mathcal{C}$ with

- $a \xhookrightarrow{[1,3] \nearrow [1,3]} b$
- $a \xhookrightarrow{[3,6] \rightarrow [0,4]} b$
- $a \xhookrightarrow{[5,7] \searrow [1,6]} b$



have:

Schemes and Hypotheses

influence scheme $\mathcal{C}$ = set of influence statements

semantics: $\mathcal{F} \models \mathcal{C}$ if $\mathcal{F} \models S$ f.a. $S \in \mathcal{C}$

hypothesis H is a statement

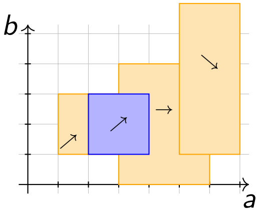
H follows from scheme $\mathcal{C}$, written $\mathcal{C} \models H$ if $\mathcal{F} \models \mathcal{C}$ entails $\mathcal{F} \models H$ f.a. $\mathcal{F}$

consider scheme $\mathcal{C}$ with

- $a \xhookrightarrow{[1,3] \nearrow [1,3]} b$
- $a \xhookrightarrow{[3,6] \rightarrow [0,4]} b$
- $a \xhookrightarrow{[5,7] \searrow [1,6]} b$

have:

- $\mathcal{C} \models a \xhookrightarrow{[2,4] \nearrow [1,3]} b$



Schemes and Hypotheses

influence scheme $\mathcal{C}$ = set of influence statements

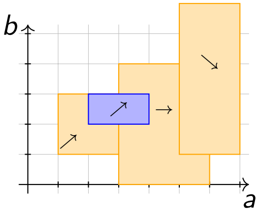
semantics: $\mathcal{F} \models \mathcal{C}$ if $\mathcal{F} \models S$ f.a. $S \in \mathcal{C}$

hypothesis H is a statement

H follows from scheme $\mathcal{C}$, written $\mathcal{C} \models H$ if $\mathcal{F} \models \mathcal{C}$ entails $\mathcal{F} \models H$ f.a. $\mathcal{F}$

consider scheme $\mathcal{C}$ with

- $a \xhookrightarrow{[1,3] \nearrow [1,3]} b$
- $a \xhookrightarrow{[3,6] \rightarrow [0,4]} b$
- $a \xhookrightarrow{[5,7] \searrow [1,6]} b$



have:

- $\mathcal{C} \models a \xhookrightarrow{[2,4] \nearrow [1,3]} b$
- $\mathcal{C} \not\models a \xhookrightarrow{[2,4] \nearrow [2,3]} b$

Schemes and Hypotheses

influence scheme $\mathcal{C}$ = set of influence statements

semantics: $\mathcal{F} \models \mathcal{C}$ if $\mathcal{F} \models S$ f.a. $S \in \mathcal{C}$

hypothesis H is a statement

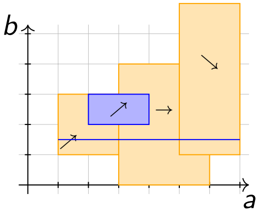
H follows from scheme $\mathcal{C}$, written $\mathcal{C} \models H$ if $\mathcal{F} \models \mathcal{C}$ entails $\mathcal{F} \models H$ f.a. $\mathcal{F}$

consider scheme $\mathcal{C}$ with

- $a \xhookrightarrow{[1,3] \nearrow [1,3]} b$
- $a \xhookrightarrow{[3,6] \rightarrow [0,4]} b$
- $a \xhookrightarrow{[5,7] \searrow [1,6]} b$

have:

- $\mathcal{C} \models a \xhookrightarrow{[2,4] \nearrow [1,3]} b$
- $\mathcal{C} \not\models a \xhookrightarrow{[2,4] \nearrow [2,3]} b$



The Calculus of Influence

goal: (efficient) decidability of entailment ($\models$) (cf. motivation)

first approach: Calculus of Influence (B./Lange/Möller Cade'23 + follow-up)

approximate $\models$ via proof-theoretic rules $\rightsquigarrow$ defines provability relation $\vdash$

$$\text{(fact)}\ \frac{}{S}\ \text{if } S \in \mathcal{C}$$

The Calculus of Influence

goal: (efficient) decidability of entailment ($\models$) (cf. motivation)

first approach: Calculus of Influence (B./Lange/Möller Cade'23 + follow-up)

approximate $\models$ via proof-theoretic rules $\rightsquigarrow$ defines provability relation $\vdash$

$$\text{(fact)}\ \frac{}{S}\ \text{if } S \in \mathcal{C}$$

The Calculus of Influence

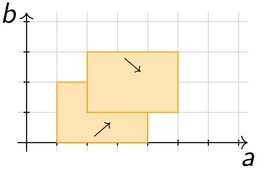
goal: (efficient) decidability of entailment ($\models$) (cf. motivation)

first approach: Calculus of Influence (B./Lange/Möller Cade'23 + follow-up)

approximate $\models$ via proof-theoretic rules $\rightsquigarrow$ defines provability relation $\vdash$

$$\text{(fact)}\ \frac{}{S}\ \text{if } S \in \mathcal{C}$$

$$(\text{int}^+)\ \frac{a \xhookrightarrow{I_1\, q\, I_1'} b \qquad a \xhookrightarrow{I_2\, q'\, I_2'} b}{a \xhookrightarrow{I_1 \cap I_2\ \inf_{\preceq}(q,q')\ I_1' \cap I_2'} b}$$



The Calculus of Influence

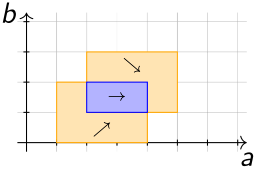
goal: (efficient) decidability of entailment ($\models$) (cf. motivation)

first approach: Calculus of Influence (B./Lange/Möller Cade'23 + follow-up)

approximate $\models$ via proof-theoretic rules $\rightsquigarrow$ defines provability relation $\vdash$

$$(\text{fact}) \; \frac{}{S} \; \text{if } S \in \mathcal{C}$$

$$(\text{int}^+) \; \frac{a \xhookrightarrow{I_1\, q\, I_1'} b \qquad a \xhookrightarrow{I_2\, q'\, I_2'} b}{a \xhookrightarrow{I_1 \cap I_2\ \inf_{\preceq}(q,q')\ I_1' \cap I_2'} b}$$



The Calculus of Influence

goal: (efficient) decidability of entailment ($\models$) (cf. motivation)

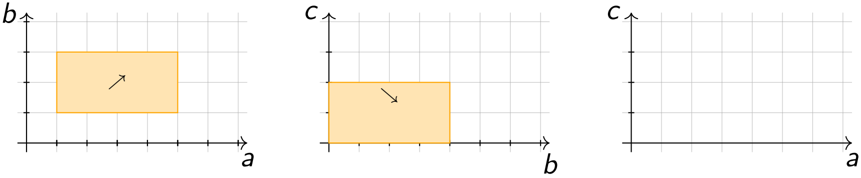
first approach: Calculus of Influence (B./Lange/Möller Cade'23 + follow-up)

approximate $\models$ via proof-theoretic rules $\rightsquigarrow$ defines provability relation $\vdash$

$$(\text{fact})\ \frac{}{S}\ \text{if } S \in \mathcal{C}$$

$$(\text{int}^+)\ \frac{a \xhookrightarrow{I_1\, q\, I_1'} b \qquad a \xhookrightarrow{I_2\, q'\, I_2'} b}{a \xhookrightarrow{I_1 \cap I_2\ \inf_{\preceq}(q,q')\ I_1' \cap I_2'} b}$$

$$(\text{trans})\ \frac{a \xhookrightarrow{I\, q\, I_1} b \qquad b \xhookrightarrow{I_2\, q'\, I'} c}{a \xhookrightarrow{I\, q \otimes q'\, I'} c}\ \text{if } I_1 \subseteq I_2$$



The Calculus of Influence

goal: (efficient) decidability of entailment ($\models$) (cf. motivation)

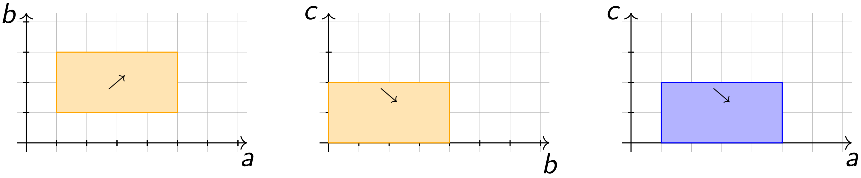
first approach: Calculus of Influence (B./Lange/Möller Cade'23 + follow-up)

approximate $\models$ via proof-theoretic rules $\rightsquigarrow$ defines provability relation $\vdash$

$$\text{(fact)}\ \frac{}{S}\ \text{if } S \in \mathcal{C}$$

$$(\text{int}^+)\ \frac{a \xhookrightarrow{I_1\, q\, I_1'} b \qquad a \xhookrightarrow{I_2\, q'\, I_2'} b}{a \xhookrightarrow{I_1 \cap I_2\ \inf_{\preceq}(q,q')\ I_1' \cap I_2'} b}$$

$$\text{(trans)}\ \frac{a \xhookrightarrow{I\, q\, I_1} b \qquad b \xhookrightarrow{I_2\, q'\, I'} c}{a \xhookrightarrow{I\ q \otimes q'\ I'} c}\ \text{if } I_1 \subseteq I_2$$



Limits of the Calculus

Theorem 1

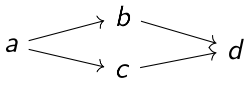
On diamond-free, elementary schemes, the Calculus of Influence is complete, i.e., $\mathcal{C} \models H$ *iff* $\mathcal{C} \vdash H$.

Limits of the Calculus

Theorem 1

On diamond-free, elementary schemes, the Calculus of Influence is complete, i.e., $\mathcal{C} \models H$ *iff* $\mathcal{C} \vdash H$.

- a scheme is diamond-free if variable order has no diamonds like below

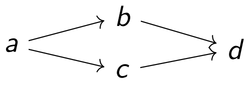


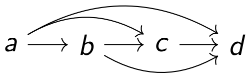
Limits of the Calculus

Theorem 1

On diamond-free, elementary schemes, the Calculus of Influence is complete, i.e., $\mathcal{C} \models H$ *iff* $\mathcal{C} \vdash H$.

- a scheme is diamond-free if variable order has no diamonds like below
- a scheme is elementary if statements only on elementary variable pairs (i.e. not on $a \mapsto c, a \mapsto d, b \mapsto d$)





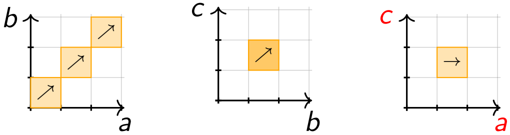
Limits of the Calculus

Theorem 1

On diamond-free, elementary schemes, the Calculus of Influence is complete, i.e., $\mathcal{C} \models H$ *iff* $\mathcal{C} \vdash H$.

- a scheme is diamond-free if variable order has no diamonds like below
- a scheme is elementary if statements only on elementary variable pairs (i.e. not on $a \mapsto c, a \mapsto d, b \mapsto d$)

failure for e.g., non-elementary schemes: backwards reasoning



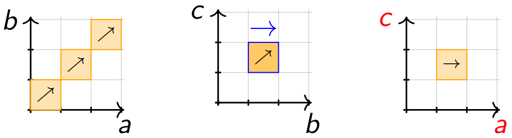
Limits of the Calculus

Theorem 1

On diamond-free, elementary schemes, the Calculus of Influence is complete, i.e., $\mathcal{C} \models H$ iff $\mathcal{C} \vdash H$.

- a scheme is diamond-free if variable order has no diamonds like below
- a scheme is elementary if statements only on elementary variable pairs (i.e. not on $a \mapsto c, a \mapsto d, b \mapsto d$)

failure for e.g., non-elementary schemes: backwards reasoning

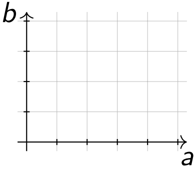


A New Approach

new, direct approach: compute whether $\mathcal{C} \models H$ by exhaustive testing

problem: in general, have infinitely many experiments $\mathcal{F}$ with $\mathcal{F} \models \mathcal{C}$

however: differences often (but not always) negligible w.r.t. H

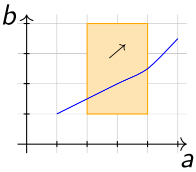


A New Approach

new, direct approach: compute whether $\mathcal{C} \models H$ by exhaustive testing

problem: in general, have infinitely many experiments $\mathcal{F}$ with $\mathcal{F} \models \mathcal{C}$

however: differences often (but not always) negligible w.r.t. H



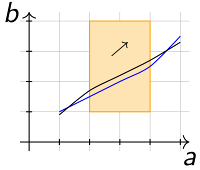
the blue

A New Approach

new, direct approach: compute whether $\mathcal{C} \models H$ by exhaustive testing

problem: in general, have infinitely many experiments $\mathcal{F}$ with $\mathcal{F} \models \mathcal{C}$

however: differences often (but not always) negligible w.r.t. H



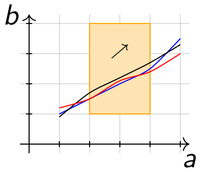
the blue, black

A New Approach

new, direct approach: compute whether $\mathcal{C} \models H$ by exhaustive testing

problem: in general, have infinitely many experiments $\mathcal{F}$ with $\mathcal{F} \models \mathcal{C}$

however: differences often (but not always) negligible w.r.t. H



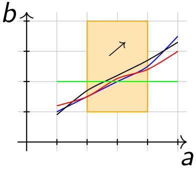
the blue, black and red functions are quite similar

A New Approach

new, direct approach: compute whether $\mathcal{C} \models H$ by exhaustive testing

problem: in general, have infinitely many experiments $\mathcal{F}$ with $\mathcal{F} \models \mathcal{C}$

however: differences often (but not always) negligible w.r.t. H



the blue, black and red functions are quite similar

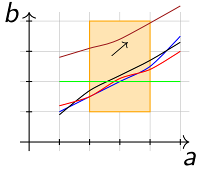
the green one is different (it is constant)

A New Approach

new, direct approach: compute whether $\mathcal{C} \models H$ by exhaustive testing

problem: in general, have infinitely many experiments $\mathcal{F}$ with $\mathcal{F} \models \mathcal{C}$

however: differences often (but not always) negligible w.r.t. H



the blue, black and red functions are quite similar

the green one is different (it is constant), and so is the brown one (much larger b-value)

Equivalence for Influence Experiments

want: equivalence relation $\sim_{\text{good}}$ between influence experiments s.t.

- for all H (with integer bounds), all $\mathcal{F} \sim_{\text{good}} \mathcal{F}'$, have $\mathcal{F} \models H$ iff $\mathcal{F}' \models H$
  i.e., $\sim_{\text{good}}$ refines equivalence w.r.t. integer-bounded statements

Equivalence for Influence Experiments

want: equivalence relation $\sim_{\text{good}}$ between influence experiments s.t.

- for all H (with integer bounds), all $\mathcal{F} \sim_{\text{good}} \mathcal{F}'$, have $\mathcal{F} \models H$ iff $\mathcal{F}' \models H$
  i.e., $\sim_{\text{good}}$ refines equivalence w.r.t. integer-bounded statements
- $\sim$ has finite index
- equivalence classes of $\sim_{\text{good}}$ are effectively enumerable

Equivalence for Influence Experiments

want: equivalence relation $\sim_{\text{good}}$ between influence experiments s.t.

- for all H (with integer bounds), all $\mathcal{F} \sim_{\text{good}} \mathcal{F}'$, have $\mathcal{F} \models H$ iff $\mathcal{F}' \models H$
  i.e., $\sim_{\text{good}}$ refines equivalence w.r.t. integer-bounded statements
- $\sim$ has finite index
- equivalence classes of $\sim_{\text{good}}$ are effectively enumerable
- for each equivalence class have an effectively constructable representative

Equivalence for Influence Experiments

want: equivalence relation $\sim_{\text{good}}$ between influence experiments s.t.

- for all H (with integer bounds), all $\mathcal{F} \sim_{\text{good}} \mathcal{F}'$, have $\mathcal{F} \models H$ iff $\mathcal{F}' \models H$
  i.e., $\sim_{\text{good}}$ refines equivalence w.r.t. integer-bounded statements
- $\sim$ has finite index
- equivalence classes of $\sim_{\text{good}}$ are effectively enumerable
- for each equivalence class have an effectively constructable representative

plus ideally: $\sim_{\text{good}}$ compatible with coherence property (to be made precise later)

Equivalence for Influence Experiments

want: equivalence relation $\sim_{\text{good}}$ between influence experiments s.t.

- for all H (with integer bounds), all $\mathcal{F} \sim_{\text{good}} \mathcal{F}'$, have $\mathcal{F} \models H$ iff $\mathcal{F}' \models H$
  i.e., $\sim_{\text{good}}$ refines equivalence w.r.t. integer-bounded statements
- $\sim$ has finite index
- equivalence classes of $\sim_{\text{good}}$ are effectively enumerable
- for each equivalence class have an effectively constructable representative

plus ideally: $\sim_{\text{good}}$ compatible with coherence property (to be made precise later)

yields: whether $\mathcal{C} \models H$ reduces to testing finitely many cases

equivalence relation that satisfies above requirements is good

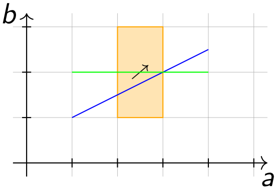
The Case of One Variable Pair

statements already introduce some equivalence: influences that satisfy statement vs. those that don't (cf. previous example)

The Case of One Variable Pair

statements already introduce some equivalence: influences that satisfy statement vs. those that don't (cf. previous example)

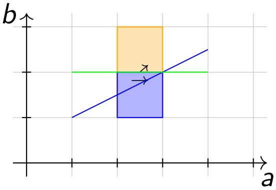
clearly too coarse:



The Case of One Variable Pair

statements already introduce some equivalence: influences that satisfy statement vs. those that don't (cf. previous example)

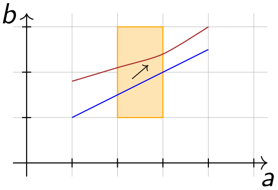
clearly too coarse:



The Case of One Variable Pair

statements already introduce some equivalence: influences that satisfy statement vs. those that don't (cf. previous example)

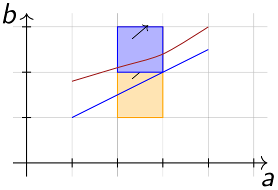
clearly too coarse:



The Case of One Variable Pair

statements already introduce some equivalence: influences that satisfy statement vs. those that don't (cf. previous example)

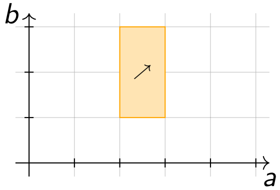
clearly too coarse:



The Case of One Variable Pair

statements already introduce some equivalence: influences that satisfy statement vs. those that don't (cf. previous example)

clearly too coarse:

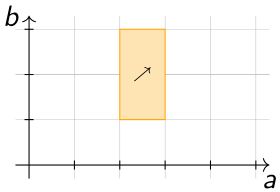


idea: refine this equivalence "up to integer grid precision"

The Case of One Variable Pair

statements already introduce some equivalence: influences that satisfy statement vs. those that don't (cf. previous example)

clearly too coarse:



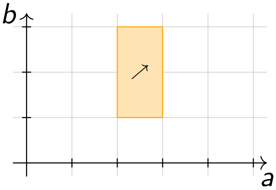
idea: refine this equivalence "up to integer grid precision"

- chop domain of statements to intervals of length 1

The Case of One Variable Pair

statements already introduce some equivalence: influences that satisfy statement vs. those that don't (cf. previous example)

clearly too coarse:



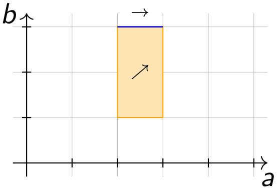
idea: refine this equivalence "up to integer grid precision"

- chop domain of statements to intervals of length 1
- categorize exact range up to integer bounds

The Case of One Variable Pair

statements already introduce some equivalence: influences that satisfy statement vs. those that don't (cf. previous example)

clearly too coarse:



idea: refine this equivalence "up to integer grid precision"

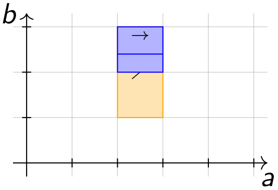
- chop domain of statements to intervals of length 1
- categorize exact range up to integer bounds

example classes above (with representative drawn)

The Case of One Variable Pair

statements already introduce some equivalence: influences that satisfy statement vs. those that don't (cf. previous example)

clearly too coarse:



idea: refine this equivalence "up to integer grid precision"

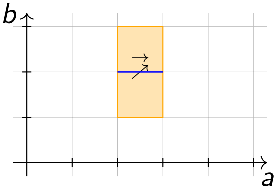
- chop domain of statements to intervals of length 1
- categorize exact range up to integer bounds

example classes above (with representative drawn)

The Case of One Variable Pair

statements already introduce some equivalence: influences that satisfy statement vs. those that don't (cf. previous example)

clearly too coarse:



idea: refine this equivalence "up to integer grid precision"

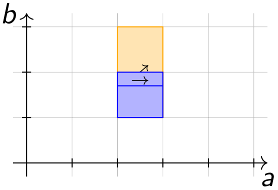
- chop domain of statements to intervals of length 1
- categorize exact range up to integer bounds

example classes above (with representative drawn)

The Case of One Variable Pair

statements already introduce some equivalence: influences that satisfy statement vs. those that don't (cf. previous example)

clearly too coarse:



idea: refine this equivalence "up to integer grid precision"

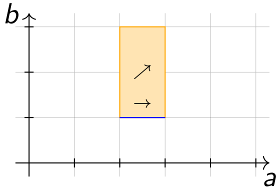
- chop domain of statements to intervals of length 1
- categorize exact range up to integer bounds

example classes above (with representative drawn)

The Case of One Variable Pair

statements already introduce some equivalence: influences that satisfy statement vs. those that don't (cf. previous example)

clearly too coarse:



idea: refine this equivalence "up to integer grid precision"

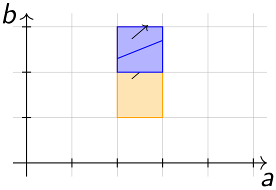
- chop domain of statements to intervals of length 1
- categorize exact range up to integer bounds

example classes above (with representative drawn)

The Case of One Variable Pair

statements already introduce some equivalence: influences that satisfy statement vs. those that don't (cf. previous example)

clearly too coarse:



idea: refine this equivalence "up to integer grid precision"

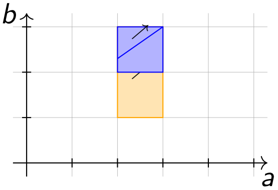
- chop domain of statements to intervals of length 1
- categorize exact range up to integer bounds

example classes above (with representative drawn)

The Case of One Variable Pair

statements already introduce some equivalence: influences that satisfy statement vs. those that don't (cf. previous example)

clearly too coarse:



idea: refine this equivalence "up to integer grid precision"

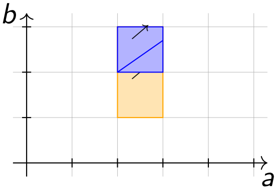
- chop domain of statements to intervals of length 1
- categorize exact range up to integer bounds

example classes above (with representative drawn)

The Case of One Variable Pair

statements already introduce some equivalence: influences that satisfy statement vs. those that don't (cf. previous example)

clearly too coarse:



idea: refine this equivalence "up to integer grid precision"

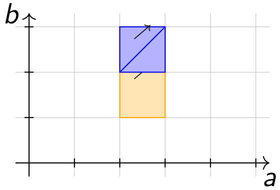
- chop domain of statements to intervals of length 1
- categorize exact range up to integer bounds

example classes above (with representative drawn)

The Case of One Variable Pair

statements already introduce some equivalence: influences that satisfy statement vs. those that don't (cf. previous example)

clearly too coarse:



idea: refine this equivalence "up to integer grid precision"

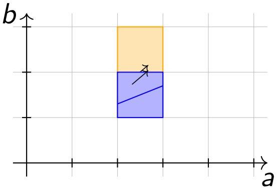
- chop domain of statements to intervals of length 1
- categorize exact range up to integer bounds

example classes above (with representative drawn)

The Case of One Variable Pair

statements already introduce some equivalence: influences that satisfy statement vs. those that don't (cf. previous example)

clearly too coarse:



idea: refine this equivalence "up to integer grid precision"

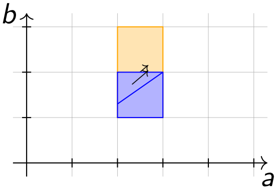
- chop domain of statements to intervals of length 1
- categorize exact range up to integer bounds

example classes above (with representative drawn)

The Case of One Variable Pair

statements already introduce some equivalence: influences that satisfy statement vs. those that don't (cf. previous example)

clearly too coarse:



idea: refine this equivalence "up to integer grid precision"

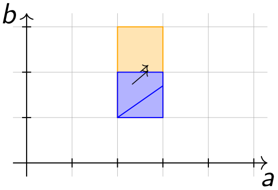
- chop domain of statements to intervals of length 1
- categorize exact range up to integer bounds

example classes above (with representative drawn)

The Case of One Variable Pair

statements already introduce some equivalence: influences that satisfy statement vs. those that don't (cf. previous example)

clearly too coarse:



idea: refine this equivalence "up to integer grid precision"

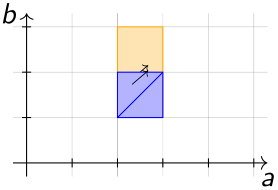
- chop domain of statements to intervals of length 1
- categorize exact range up to integer bounds

example classes above (with representative drawn)

The Case of One Variable Pair

statements already introduce some equivalence: influences that satisfy statement vs. those that don't (cf. previous example)

clearly too coarse:



idea: refine this equivalence "up to integer grid precision"

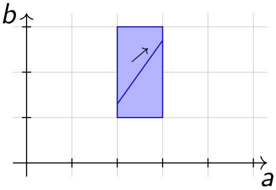
- chop domain of statements to intervals of length 1
- categorize exact range up to integer bounds

example classes above (with representative drawn)

The Case of One Variable Pair

statements already introduce some equivalence: influences that satisfy statement vs. those that don't (cf. previous example)

clearly too coarse:



idea: refine this equivalence "up to integer grid precision"

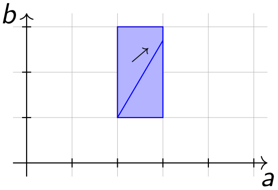
- chop domain of statements to intervals of length 1
- categorize exact range up to integer bounds

example classes above (with representative drawn)

The Case of One Variable Pair

statements already introduce some equivalence: influences that satisfy statement vs. those that don't (cf. previous example)

clearly too coarse:



idea: refine this equivalence "up to integer grid precision"

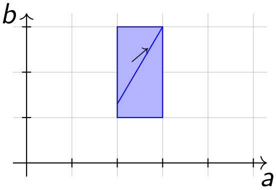
- chop domain of statements to intervals of length 1
- categorize exact range up to integer bounds

example classes above (with representative drawn)

The Case of One Variable Pair

statements already introduce some equivalence: influences that satisfy statement vs. those that don't (cf. previous example)

clearly too coarse:



idea: refine this equivalence "up to integer grid precision"

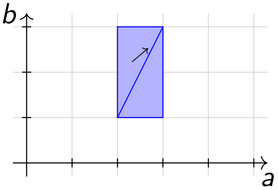
- chop domain of statements to intervals of length 1
- categorize exact range up to integer bounds

example classes above (with representative drawn)

The Case of One Variable Pair

statements already introduce some equivalence: influences that satisfy statement vs. those that don't (cf. previous example)

clearly too coarse:



idea: refine this equivalence "up to integer grid precision"

- chop domain of statements to intervals of length 1
- categorize exact range up to integer bounds

example classes above (with representative drawn)

One Pair is not Enough

define $\sim_{\mathsf{rng}}$ as natural product of equivalence classes "up to integer precision" induced by individual classes in scheme

One Pair is not Enough

define $\sim_{\text{rng}}$ as natural product of equivalence classes "up to integer precision" induced by individual classes in scheme

Theorem 2

For experiments with one variable pair, $\sim_{rng}$ is good.

One Pair is not Enough

define $\sim_{\text{rng}}$ as natural product of equivalence classes "up to integer precision" induced by individual classes in scheme

Theorem 2

For experiments with one variable pair, $\sim_{rng}$ is good.

unfortunately, not enough in case of several pairs:

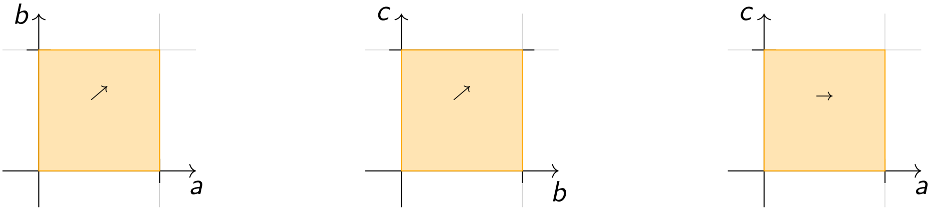
One Pair is not Enough

define $\sim_{\mathrm{rng}}$ as natural product of equivalence classes "up to integer precision" induced by individual classes in scheme

Theorem 2

For experiments with one variable pair, $\sim_{rng}$ is good.

unfortunately, not enough in case of several pairs:



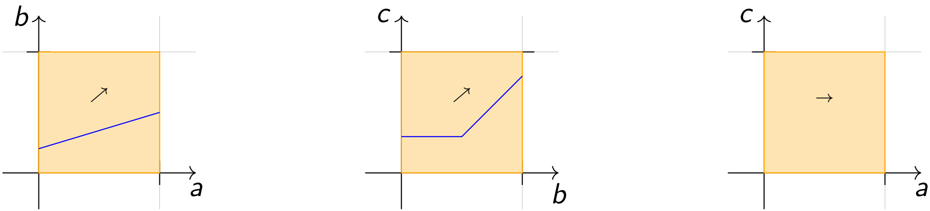
One Pair is not Enough

define $\sim_{\text{rng}}$ as natural product of equivalence classes "up to integer precision" induced by individual classes in scheme

Theorem 2

For experiments with one variable pair, $\sim_{rng}$ is good.

unfortunately, not enough in case of several pairs:



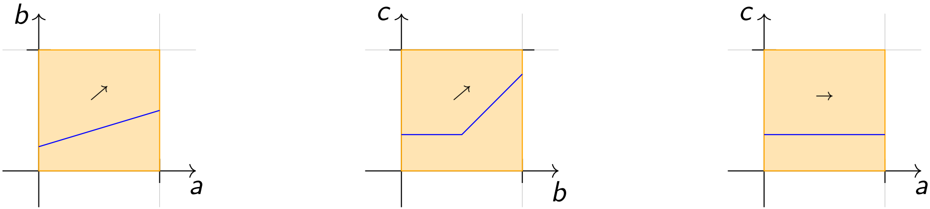
One Pair is not Enough

define $\sim_{\text{rng}}$ as natural product of equivalence classes "up to integer precision" induced by individual classes in scheme

Theorem 2

For experiments with one variable pair, $\sim_{rng}$ is good.

unfortunately, not enough in case of several pairs:



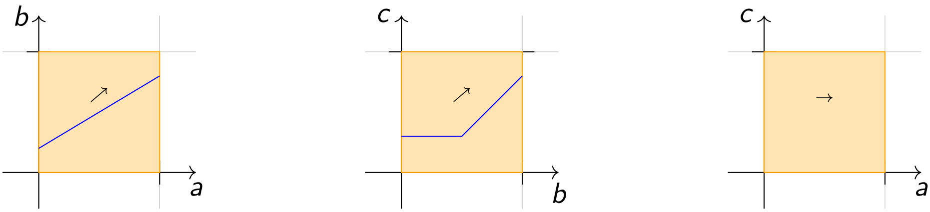
One Pair is not Enough

define $\sim_{\text{rng}}$ as natural product of equivalence classes “up to integer precision” induced by individual classes in scheme

Theorem 2

For experiments with one variable pair, $\sim_{rng}$ is good.

unfortunately, not enough in case of several pairs:



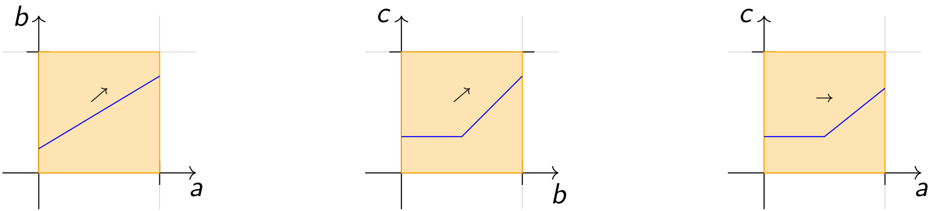
One Pair is not Enough

define $\sim_{\text{rng}}$ as natural product of equivalence classes "up to integer precision" induced by individual classes in scheme

Theorem 2

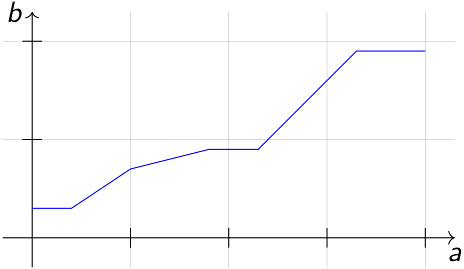
For experiments with one variable pair, $\sim_{rng}$ is good.

unfortunately, not enough in case of several pairs:



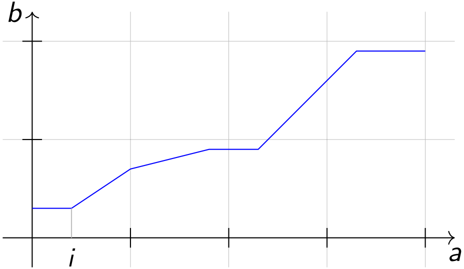
Points of Interest

goal: characterize points related directly to integer-value coordinates



Points of Interest

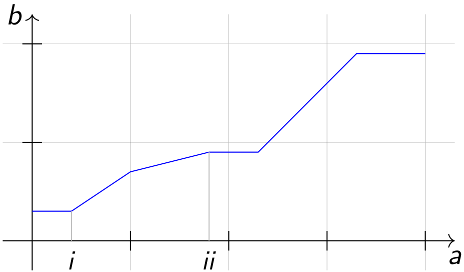
goal: characterize points related directly to integer-value coordinates



- rightmost a value with same b-value as $a = 0$

Points of Interest

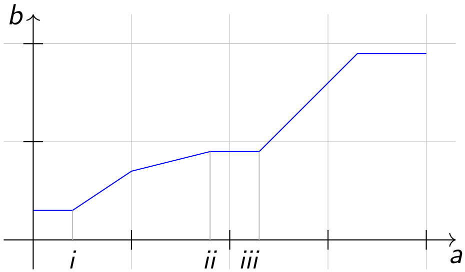
goal: characterize points related directly to integer-value coordinates



- (i) rightmost a value with same b-value as $a = 0$
- (ii) leftmost a with same b-value as $a = 2$

Points of Interest

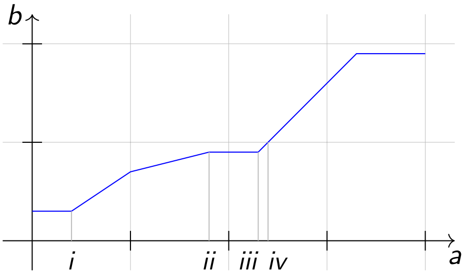
goal: characterize points related directly to integer-value coordinates



- (i) rightmost a value with same b-value as $a = 0$
- (ii) leftmost a with same b-value as $a = 2$
- (iii) rightmost a with same b-value as $a = 2$

Points of Interest

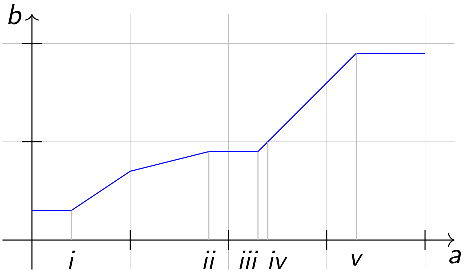
goal: characterize points related directly to integer-value coordinates



- (i) rightmost a value with same b-value as $a = 0$
- (ii) leftmost a with same b-value as $a = 2$
- (iii) rightmost a with same b-value as $a = 2$
- (iv) a-value where b-value is 2

Points of Interest

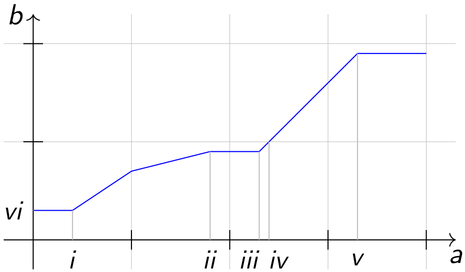
goal: characterize points related directly to integer-value coordinates



- (i) rightmost a value with same b-value as $a = 0$
- (ii) leftmost a with same b-value as $a = 2$
- (iii) rightmost a with same b-value as $a = 2$
- (iv) a-value where b-value is 2
- (v) leftmost a-value with same b-value as $a = 4$

Points of Interest

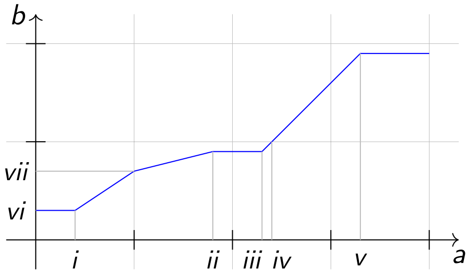
goal: characterize points related directly to integer-value coordinates



- i rightmost a value with same b-value as $a = 0$
- ii leftmost a with same b-value as $a = 2$
- iii rightmost a with same b-value as $a = 2$
- iv a-value where b-value is 2
- v leftmost a-value with same b-value as $a = 4$
- vi b-value of $a = 0$

Points of Interest

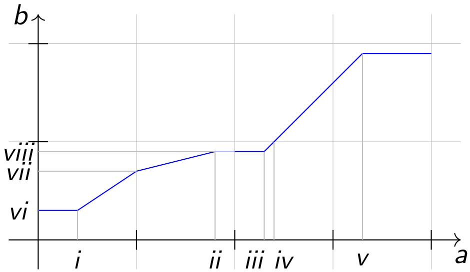
goal: characterize points related directly to integer-value coordinates



- (i) rightmost a value with same b-value as $a = 0$
- (ii) leftmost a with same b-value as $a = 2$
- (iii) rightmost a with same b-value as $a = 2$
- (iv) a-value where b-value is 2
- (v) leftmost a-value with same b-value as $a = 4$
- (vi) b-value of $a = 0$
- (vii) b-value of $a = 1$

Points of Interest

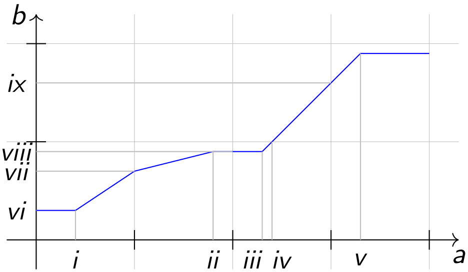
goal: characterize points related directly to integer-value coordinates



- (i) rightmost a value with same b-value as $a = 0$
- (ii) leftmost a with same b-value as $a = 2$
- (iii) rightmost a with same b-value as $a = 2$
- (iv) a-value where b-value is 2
- (v) leftmost a-value with same b-value as $a = 4$
- (vi) b-value of $a = 0$
- (vii) b-value of $a = 1$
- (viii) b-value of $a = 2$

Points of Interest

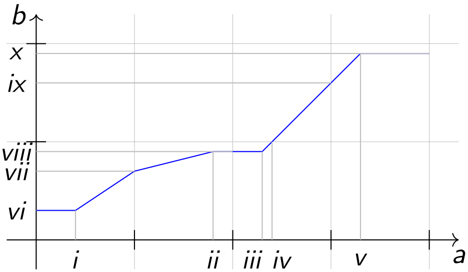
goal: characterize points related directly to integer-value coordinates



- (i) rightmost a value with same b-value as $a = 0$
- (ii) leftmost a with same b-value as $a = 2$
- (iii) rightmost a with same b-value as $a = 2$
- (iv) a-value where b-value is 2
- (v) leftmost a-value with same b-value as $a = 4$
- (vi) b-value of $a = 0$
- (vii) b-value of $a = 1$
- (viii) b-value of $a = 2$
- (ix) b-value of $a = 3$

Points of Interest

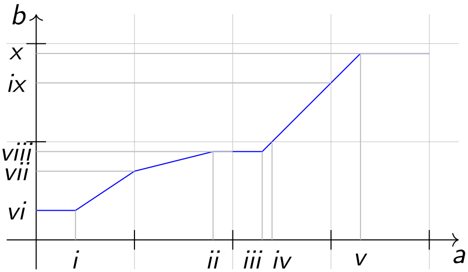
goal: characterize points related directly to integer-value coordinates



- (i) rightmost a value with same b-value as $a = 0$
- (ii) leftmost a with same b-value as $a = 2$
- (iii) rightmost a with same b-value as $a = 2$
- (iv) a-value where b-value is 2
- (v) leftmost a-value with same b-value as $a = 4$
- (vi) b-value of $a = 0$
- (vii) b-value of $a = 1$
- (viii) b-value of $a = 2$
- (ix) b-value of $a = 3$
- (x) b-value of $a = 4$

Points of Interest

goal: characterize points related directly to integer-value coordinates



- (i) rightmost a value with same b-value as $a = 0$
- (ii) leftmost a with same b-value as $a = 2$
- (iii) rightmost a with same b-value as $a = 2$
- (iv) a-value where b-value is 2
- (v) leftmost a-value with same b-value as $a = 4$
- (vi) b-value of $a = 0$
- (vii) b-value of $a = 1$
- (viii) b-value of $a = 2$
- (ix) b-value of $a = 3$
- (x) b-value of $a = 4$

exact value of points not important, just their order and relation to integer values

Categorizing Experiments using Points of Interest

pre-categorize functions using $\sim_{\mathsf{rng}}$ (make ranges exact "up to integer values")

Categorizing Experiments using Points of Interest

pre-categorize functions using $\sim_{\mathsf{rng}}$ (make ranges exact "up to integer values")

for each class, generate finitely many points of interest as follows (here for $\nearrow$):

- $\nearrow$ function $\mathcal{F}_{a,b}$ induces PoI on a-axis:
  - last point in $[i, i+1]$ to have same b-value as function on $a = i$

Categorizing Experiments using Points of Interest

pre-categorize functions using $\sim_{\text{rng}}$ (make ranges exact "up to integer values")

for each class, generate finitely many points of interest as follows (here for $\nearrow$):

- $\nearrow$ function $\mathcal{F}_{a,b}$ induces PoI on a-axis:
  - last point in $[i, i+1]$ to have same b-value as function on $a = i$
  - first point in $[i, i+1]$ to have same b-value as function on $a = i$

Categorizing Experiments using Points of Interest

pre-categorize functions using $\sim_{\text{rng}}$ (make ranges exact "up to integer values")

for each class, generate finitely many points of interest as follows (here for $\nearrow$):

- $\nearrow$ function $\mathcal{F}_{a,b}$ induces PoI on a-axis:
  - last point in $[i, i+1]$ to have same b-value as function on $a = i$
  - first point in $[i, i+1]$ to have same b-value as function on $a = i$
- function $\mathcal{F}_{a,b}$ induces PoI on b-axis:
  - b-value of function on $a = i$, for each integer i

Categorizing Experiments using Points of Interest

pre-categorize functions using $\sim_{\text{rng}}$ (make ranges exact "up to integer values")

for each class, generate finitely many points of interest as follows (here for $\nearrow$):

- $\nearrow$ function $\mathcal{F}_{a,b}$ induces PoI on a-axis:
  - last point in $[i, i+1]$ to have same b-value as function on $a = i$
  - first point in $[i, i+1]$ to have same b-value as function on $a = i$
- function $\mathcal{F}_{a,b}$ induces PoI on b-axis:
  - b-value of function on $a = i$, for each integer i
- points transfer transitively (not shown), e.g., last/first points with same value as PoI also PoI

Categorizing Experiments using Points of Interest

pre-categorize functions using $\sim_{\text{rng}}$ (make ranges exact "up to integer values")

for each class, generate finitely many points of interest as follows (here for $\nearrow$):

- $\nearrow$ function $\mathcal{F}_{a,b}$ induces PoI on a-axis:
  - last point in $[i, i+1]$ to have same b-value as function on $a = i$
  - first point in $[i, i+1]$ to have same b-value as function on $a = i$
- function $\mathcal{F}_{a,b}$ induces PoI on b-axis:
  - b-value of function on $a = i$, for each integer i
- points transfer transitively (not shown), e.g., last/first points with same value as PoI also PoI

  important: set remains finite in spite of this

generate relation $\sim_{\text{PoI}}$ by making influences equivalent if their PoI have the same order and same relation to integer values

Conjecture 3

$\sim_{\text{PoI}}$ is good, even on non-elementary schemes with diamonds.

Conclusion

problem: want to effectively decide $\models$ for statements on real-valued functions

proof-theoretic approach not feasible due to required backwards reasoning etc.

Conclusion

problem: want to effectively decide $\models$ for statements on real-valued functions

proof-theoretic approach not feasible due to required backwards reasoning etc.

introduced equivalence relation on influence experiments that (hopefully) allows to decide $\models$

Conclusion

problem: want to effectively decide $\models$ for statements on real-valued functions

proof-theoretic approach not feasible due to required backwards reasoning etc.

introduced equivalence relation on influence experiments that (hopefully) allows to decide $\models$

- pro: does away with rules of calculus

Conclusion

problem: want to effectively decide $\models$ for statements on real-valued functions

proof-theoretic approach not feasible due to required backwards reasoning etc.

introduced equivalence relation on influence experiments that (hopefully) allows to decide $\models$

- pro: does away with rules of calculus
- pro: decidability for a wider range of schemes

Conclusion

problem: want to effectively decide $\models$ for statements on real-valued functions

proof-theoretic approach not feasible due to required backwards reasoning etc.

introduced equivalence relation on influence experiments that (hopefully) allows to decide $\models$

- pro: does away with rules of calculus
- pro: decidability for a wider range of schemes
- con: complexity likely not very good (many classes even for few functions)

  maybe possible to avoid generating all of them/do iterative refinement

Conclusion

problem: want to effectively decide $\models$ for statements on real-valued functions

proof-theoretic approach not feasible due to required backwards reasoning etc.

introduced equivalence relation on influence experiments that (hopefully) allows to decide $\models$

- pro: does away with rules of calculus
- pro: decidability for a wider range of schemes
- con: complexity likely not very good (many classes even for few functions)

  maybe possible to avoid generating all of them/do iterative refinement

questions?