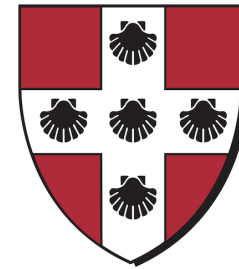


Modalities, Cohesion, and Information Flow

Alex Kavvos

Department of Mathematics and Computer Science, Wesleyan University



12th Panhellenic Logic Symposium, Anogeia, Greece, 26 June 2019

[arXiv:1809.07897](https://arxiv.org/abs/1809.07897)

Language-based Information Flow Control


The Needham-Schroeder-Lowe protocol for authentication, given public keys:

1. $A \rightarrow B : \text{Alice}, E_{k_B}(\text{Alice} \parallel \text{nonce}_A)$
2. $B \rightarrow A : E_{k_A}(\text{nonce}_A \parallel \text{Bob} \parallel \text{nonce}_B)$
3. $A \rightarrow B : E_{k_B}(\text{nonce}_B \parallel k_s)$
4. $A \rightarrow B : E_{k_s}(m_1 \parallel \text{Alice} \parallel \text{counter})$
5. etc.

Some code implementing Alice:

Alice's private key

Bob's public key



```

proc Alice(pv : Key, pb : Key, Bob-I : ?[Msg], Bob-O : ![Msg]) {
  val nonce-a = generateNonce()
  val pk = generateSymmetricKey()
  Bob-O ! (Alice, encrypt(pb, (Alice, nonce-a)))
  val (nonce-a-b, name, nonce-b) = decrypt(pv, Bob-I ? ())
  if (nonce-a == nonce-a-b && name == Bob) {
    Bob-O ! encrypt(pb, (nonce-b, pv)) ← Message #3
    Alice-session(pk, Bob-I, Bob-O)
  }
}

```

Language-based Information Flow Control

The Needham-Schroeder-Lowe protocol for authentication, given public keys:

1. $A \rightarrow B : \text{Alice}, E_{k_B}(\text{Alice} \parallel \text{nonce}_A)$
2. $B \rightarrow A : E_{k_A}(\text{nonce}_A \parallel \text{Bob} \parallel \text{nonce}_B)$
3. $A \rightarrow B : E_{k_B}(\text{nonce}_B \parallel k_s)$
4. $A \rightarrow B : E_{k_s}(m_1 \parallel \text{Alice} \parallel \text{counter})$
5. etc.

Some code implementing Alice:

Alice's private key

Bob's public key

```

proc Alice(pv : Key, pb : Key, Bob-I : ?[Msg], Bob-O : ![Msg]) {
  val nonce-a = generateNonce()
  val pk = generateSymmetricKey()
  Bob-O ! (Alice, encrypt(pb, (Alice, nonce-a)))
  val (nonce-a-b, name, nonce-b) = decrypt(pv, Bob-I ? ())
  if (nonce-a == nonce-a-b && name == Bob) {
    Bob-O ! encrypt(pb, (nonce-b, pv)) ← Message #3
    Alice-session(pk, Bob-I, Bob-O)
  }
}

```

OH NO

Language-based Information Flow Control

The Needham-Schroeder-Lowe protocol for authentication, given public keys:

1. $A \rightarrow B : \text{Alice}, E_{k_B}(\text{Alice} \parallel \text{nonce}_A)$
2. $B \rightarrow A : E_{k_A}(\text{nonce}_A \parallel \text{Bob} \parallel \text{nonce}_B)$
3. $A \rightarrow B : E_{k_B}(\text{nonce}_B \parallel k_s)$
4. $A \rightarrow B : E_{k_s}(m_1 \parallel \text{Alice} \parallel \text{counter})$
5. etc.

Some code implementing Alice:

Alice's private key

Bob's public key

```

proc Alice(pv : ♦ Key, pb : Key, Bob-I : ?[Msg], Bob-O : ![Msg]) {
  val nonce-a = generateNonce()
  val pk = generateSymmetricKey()
  Bob-O ! (Alice, encrypt(pb, (Alice, nonce-a)))
  val (nonce-a-b, name, nonce-b) = decrypt(pv, Bob-I ? ())
  if (nonce-a == nonce-a-b && name == Bob) {
    Bob-O ! encrypt(pb, (nonce-b, pv)) ← Message #3
    Alice-session(pk, Bob-I, Bob-O)
  }
}

```

Language-based Information Flow Control

The Needham-Schroeder-Lowe protocol for authentication, given public keys:

1. $A \rightarrow B : \text{Alice}, E_{k_B}(\text{Alice} \parallel \text{nonce}_A)$
2. $B \rightarrow A : E_{k_A}(\text{nonce}_A \parallel \text{Bob} \parallel \text{nonce}_B)$
3. $A \rightarrow B : E_{k_B}(\text{nonce}_B \parallel k_s)$
4. $A \rightarrow B : E_{k_s}(m_1 \parallel \text{Alice} \parallel \text{counter})$
5. etc.

Some code implementing Alice:

Alice's private key

Bob's public key

```

proc Alice(pv : ◆ Key, pb : Key, Bob-I : ?[Msg], Bob-O : ![Msg]) {
  val nonce-a = generateNonce()
  val pk = generateSymmetricKey()
  Bob-O ! (Alice, encrypt(pb, (Alice, nonce-a)))
  val (nonce-a-b, name, nonce-b) = decrypt(pv, Bob-I ? ())
  if (nonce-a == nonce-a-b && name == Bob) {
    Bob-O ! encrypt(pb, (nonce-b, pv)) ← Message #3
    Alice-session(pk, Bob-I, Bob-O)
  }
}

```

Language-based Information Flow Control

The Needham-Schroeder-Lowe protocol for authentication, given public keys:

1. $A \rightarrow B : \text{Alice}, E_{k_B}(\text{Alice} \parallel \text{nonce}_A)$
2. $B \rightarrow A : E_{k_A}(\text{nonce}_A \parallel \text{Bob} \parallel \text{nonce}_B)$
3. $A \rightarrow B : E_{k_B}(\text{nonce}_B \parallel k_s)$
4. $A \rightarrow B : E_{k_s}(m_1 \parallel \text{Alice} \parallel \text{counter})$
5. etc.

Some code implementing Alice:

Alice's private key

Bob's public key

```

proc Alice(pv : ◆ Key, pb : Key, Bob-I : ?[Msg], Bob-O : ![Msg]) {
  val nonce-a = generateNonce()
  val pk = generateSymmetricKey()
  Bob-O ! (Alice, encrypt(pb, (Alice, nonce-a)))
  val (nonce-a-b, name, nonce-b) = decrypt(pv, Bob-I ? ())
  if (nonce-a == nonce-a-b && name == Bob) {
    Bob-O ! encrypt(pb, (nonce-b, pv)) ← Message #3
    Alice-session(pk, Bob-I, Bob-O)
  }
}
  
```

TYPE ERROR

Modalities for Information Flow Control

- ✦ Modalities = unary operations on types. $T(A)$ $\Box A$ $\Diamond A$ $\|A\|$
- ✦ They can be used to control information flow, as in previous example.
One can copy techniques from the **proof theory of modal logic**.
- ✦ The hard part is proving **noninterference**:

*[...] High-security data does not “interfere”
with the calculation of low-security outputs [...]*

- ✦ A notion due to [Goguen and Meseguer, 1982]. This definition:
from seminal paper on **dependency core calculus** [Abadi et al 1999].

Modalities for Information Flow: an example

❖ An example: for each type A , a type $\blacklozenge A$ \longleftarrow "high security A "

❖ Can always get a $\blacklozenge A$:
$$\frac{\Gamma \vdash M : A}{\Gamma \vdash [M] : \blacklozenge A}$$

❖ I can use a high-security value when computing another high-security value:

$$\frac{\Gamma \vdash M : \blacklozenge A \quad \Gamma, x : A \vdash N : \blacklozenge C}{\Gamma \vdash \text{let } x = M \text{ in } N : \blacklozenge C}$$

❖ Reduction: $\text{let } x = [M] \text{ in } N \rightarrow N[M/x]$

❖ Noninterference:

If $x : \blacklozenge A \vdash E : \text{Bool}$ and $\vdash M, N : \blacklozenge A$ then
 $E[M/x]$ and $E[N/x]$ compute the same boolean value.

How can one go about proving this?

Modalities for Information Flow: an example

❖ An example: for each type A , a type $\blacklozenge A$ \longleftarrow “high security A ”

❖ Can always get a $\blacklozenge A$:
$$\frac{\Gamma \vdash M : A}{\Gamma \vdash [M] : \blacklozenge A}$$

❖ I can use a high-security value when computing another high-security value:

$$\frac{\Gamma \vdash M : \blacklozenge A \quad \Gamma, x : A \vdash N : \blacklozenge C}{\Gamma \vdash \text{let } x = M \text{ in } N : \blacklozenge C}$$

❖ Reduction: $\text{let } x = [M] \text{ in } N \rightarrow N[M/x]$

❖ Noninterference:

a.k.a.
“Moggi’s monadic
metalanguage”

If $x : \blacklozenge A \vdash E : \text{Bool}$ and $\vdash M, N : \blacklozenge A$ then
 $E[M/x]$ and $E[N/x]$ compute the same boolean value.

How can one go about proving this?

Proving noninterference

- ❖ In the last ten to fifteen years: **logical relations**.
- ❖ Beautiful, but long and complicated syntactic proofs.
- ❖ This talk: using **categorical algebra** to simplify these proofs.
- ❖ Main result: one can use basic **axiomatic cohesion** to reason about information flow and prove noninterference results.
- ❖ **Axiomatic cohesion**: a theory developed by F. William Lawvere.
Aim: axiomatic desc. of all sorts of **geometric/topological spaces**.

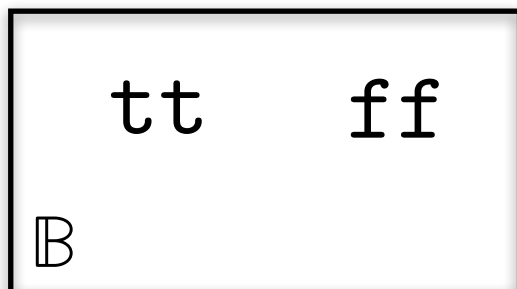
Cohesion

Spaces

(= points + cohesion)

Sets

(= points)



Cohesion

Spaces

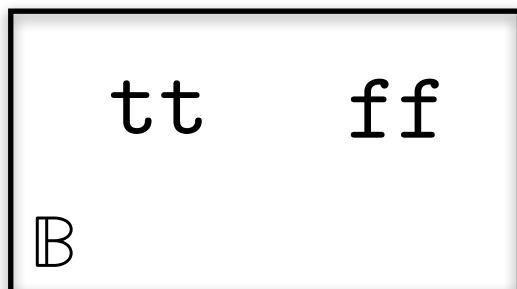
(= points + cohesion)

U



Sets

(= points)



Cohesion

Spaces

(= points + cohesion)

$X = \text{space}$
 $U(X) = \text{points of space } X \text{ (forget cohesion)}$

U

Sets

(= points)

tt ff

\mathbb{B}

Cohesion

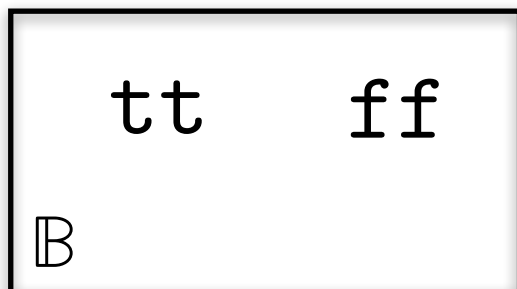
Spaces

(= points + cohesion)

U

Sets

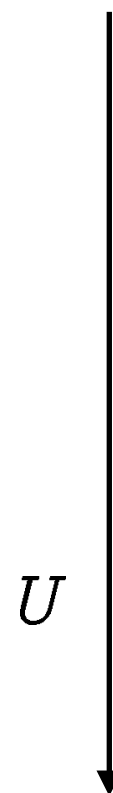
(= points)



Cohesion

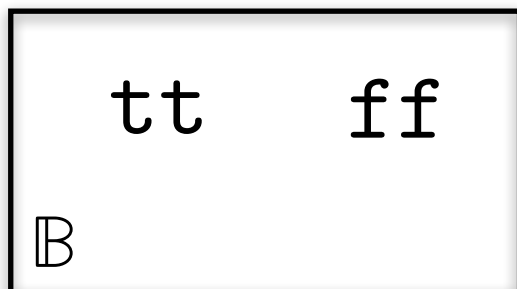
Spaces

(= points + cohesion)



Sets

(= points)



Cohesion

Spaces

(= points + cohesion)

Δ

U

S = set of points
 $\Delta(S)$ = discrete space on S (minimum cohesion)

Sets

(= points)

tt ff

\mathbb{B}

Cohesion

Spaces

(= points + cohesion)

Δ

U

S = set of points
 $\Delta(S)$ = discrete space on S (minimum cohesion)

Sets

(= points)

tt ff

\mathbb{B}

tt ff

$\Delta(\mathbb{B})$

Cohesion

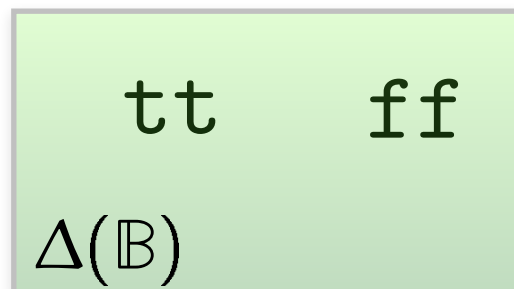
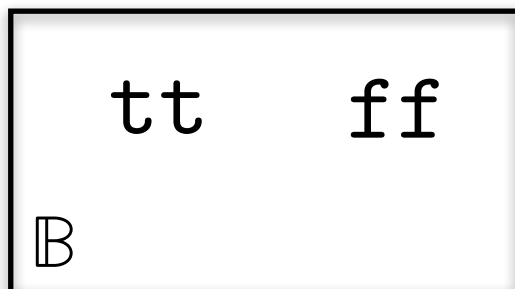
Spaces

(= points + cohesion)



Sets

(= points)



Cohesion

Spaces

(= points + cohesion)

Δ

U

∇

Sets

(= points)

tt

ff

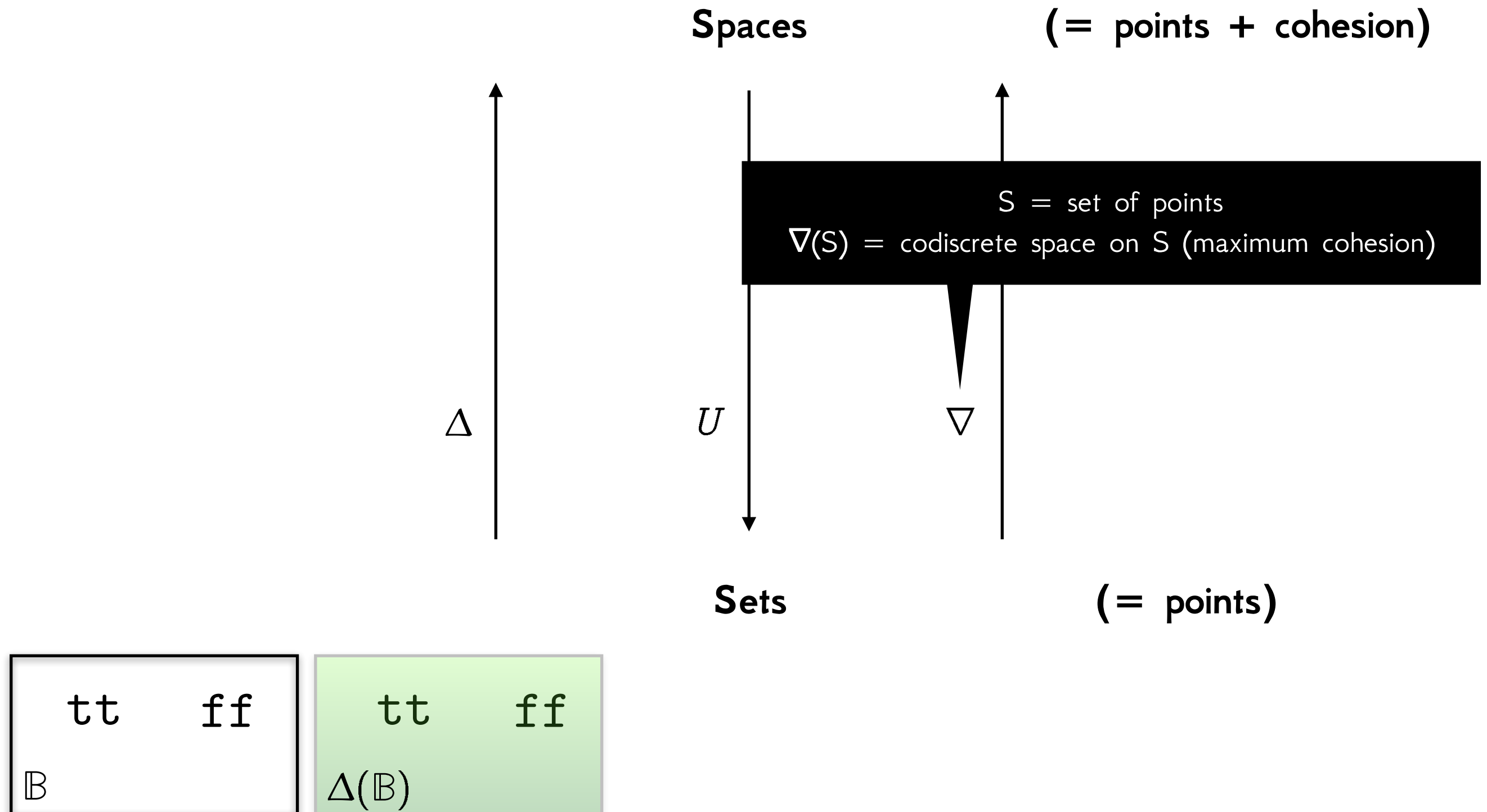
\mathbb{B}

tt

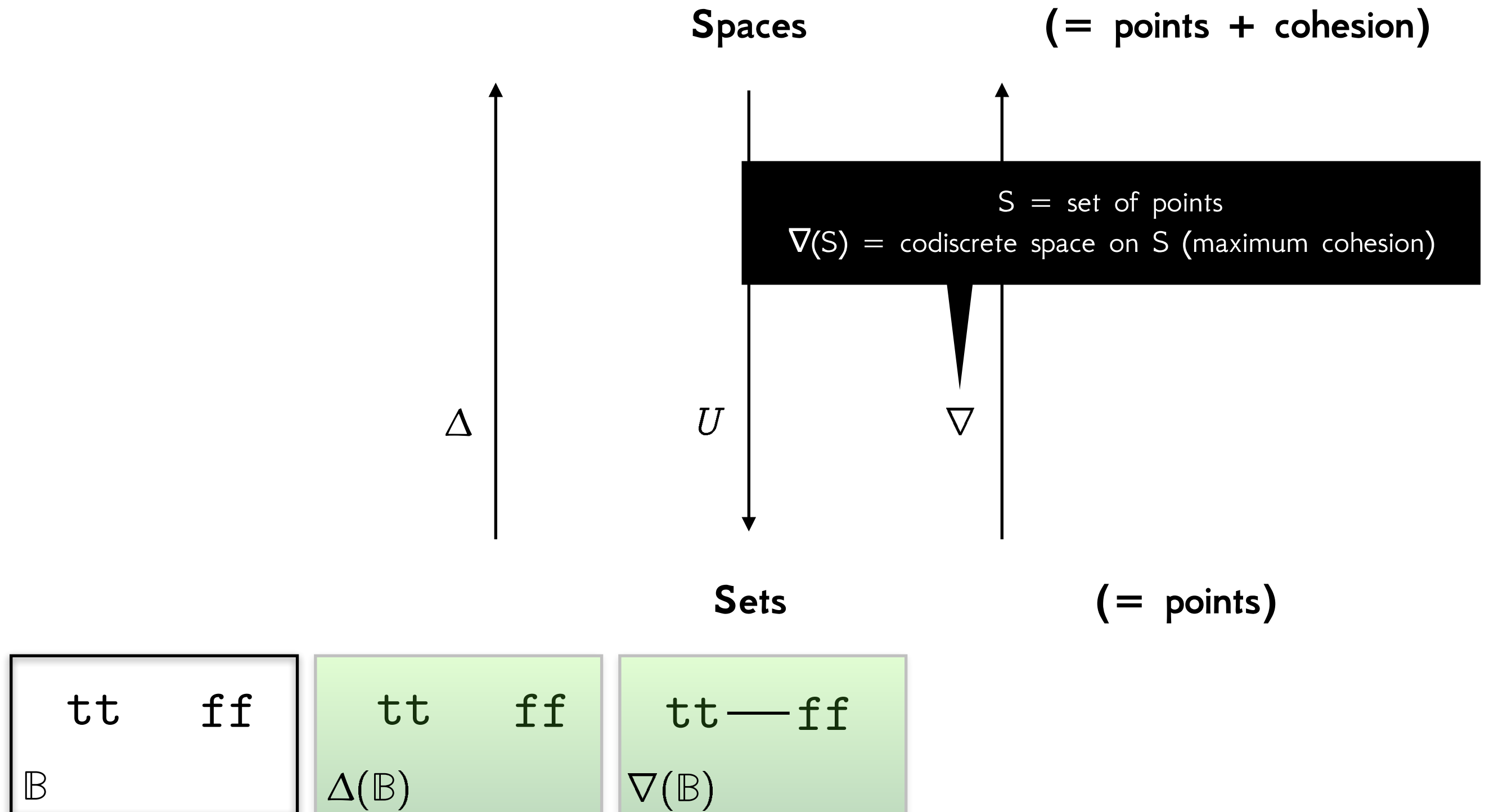
ff

$\Delta(\mathbb{B})$

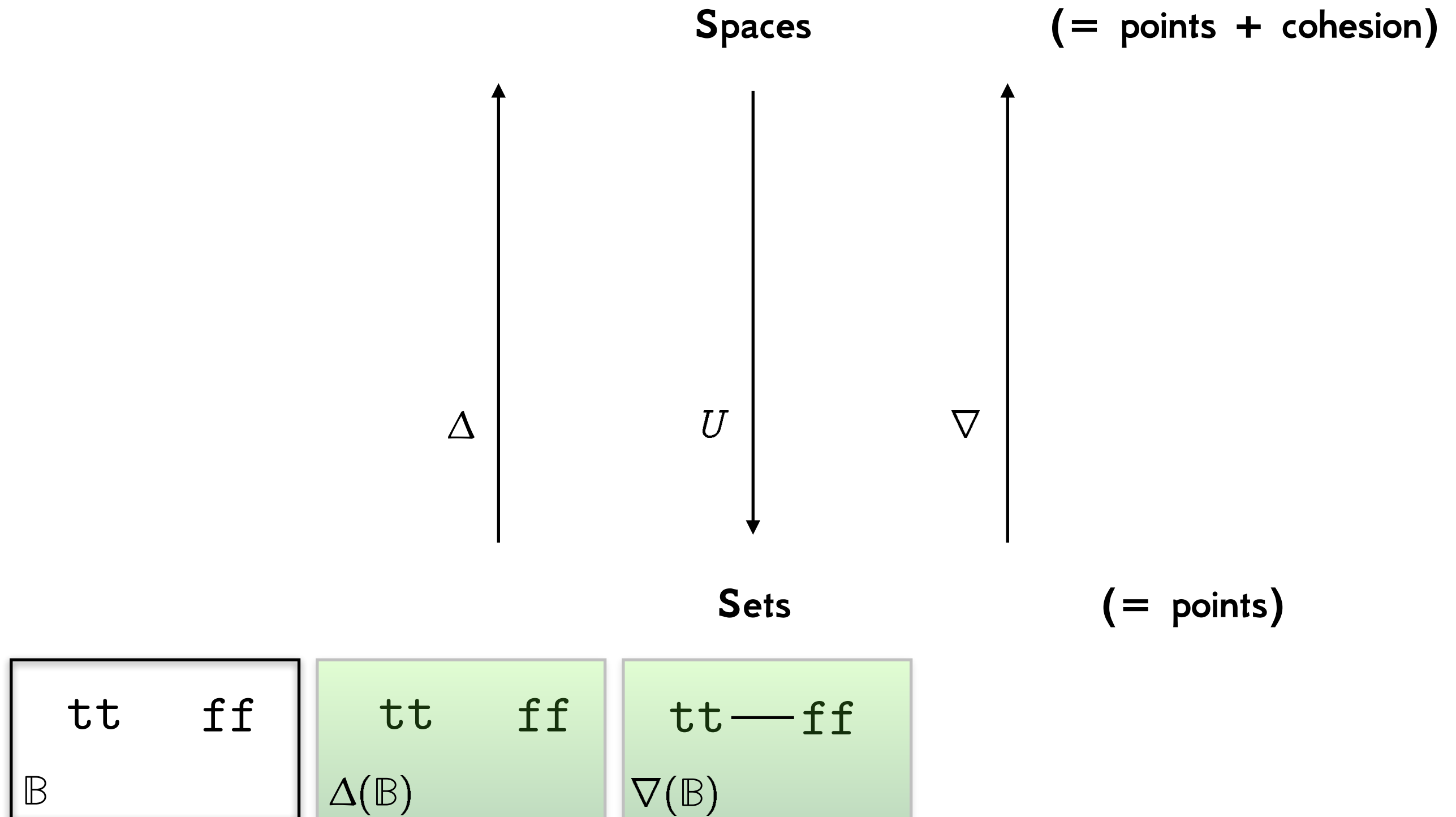
Cohesion



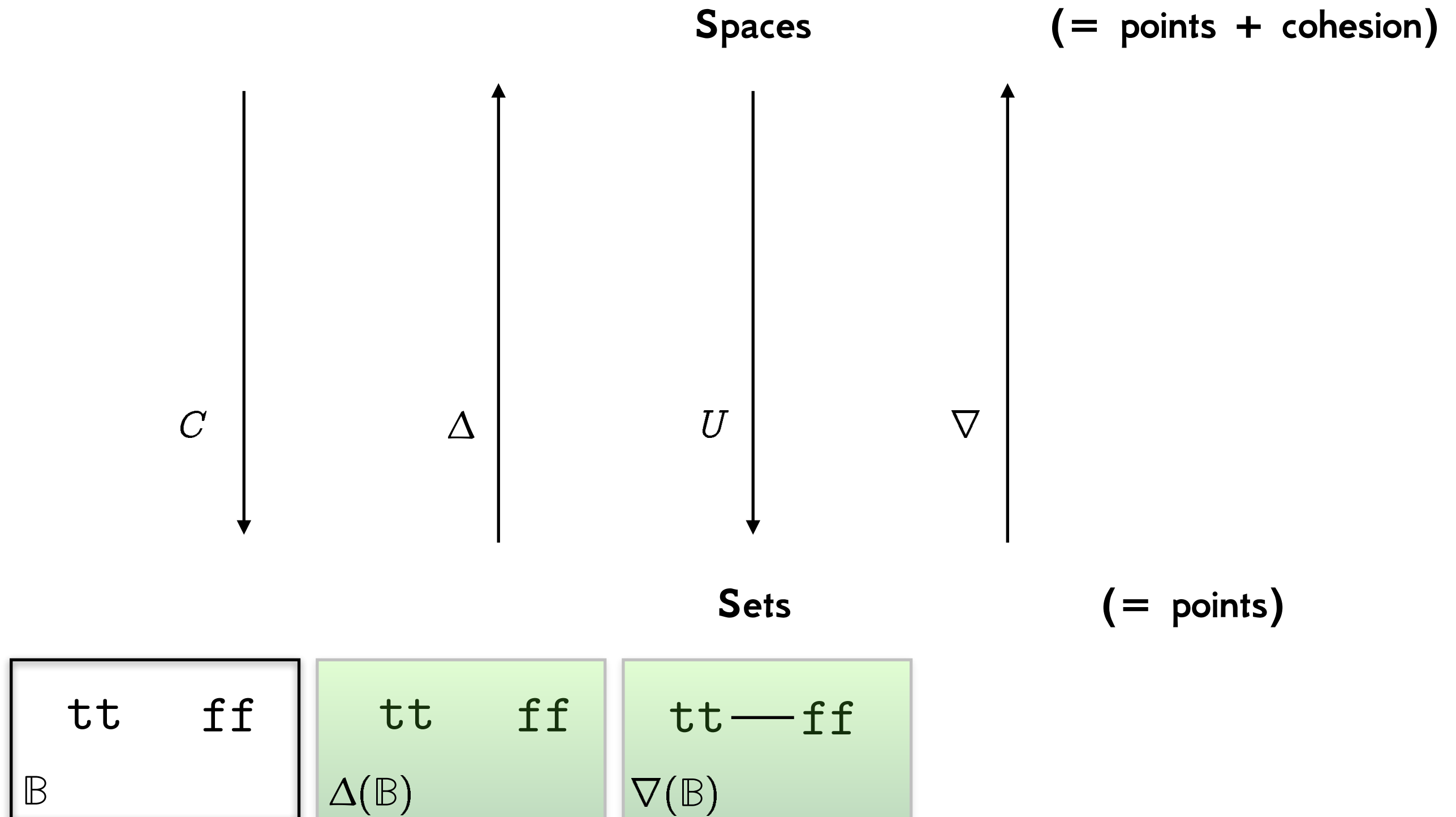
Cohesion



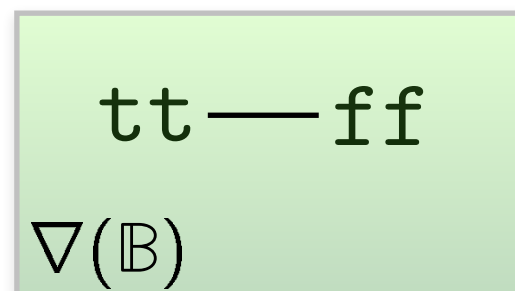
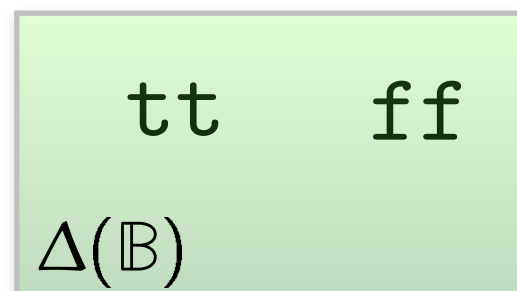
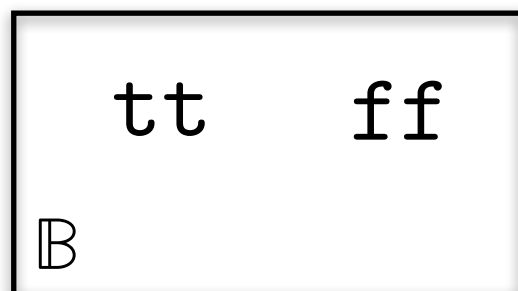
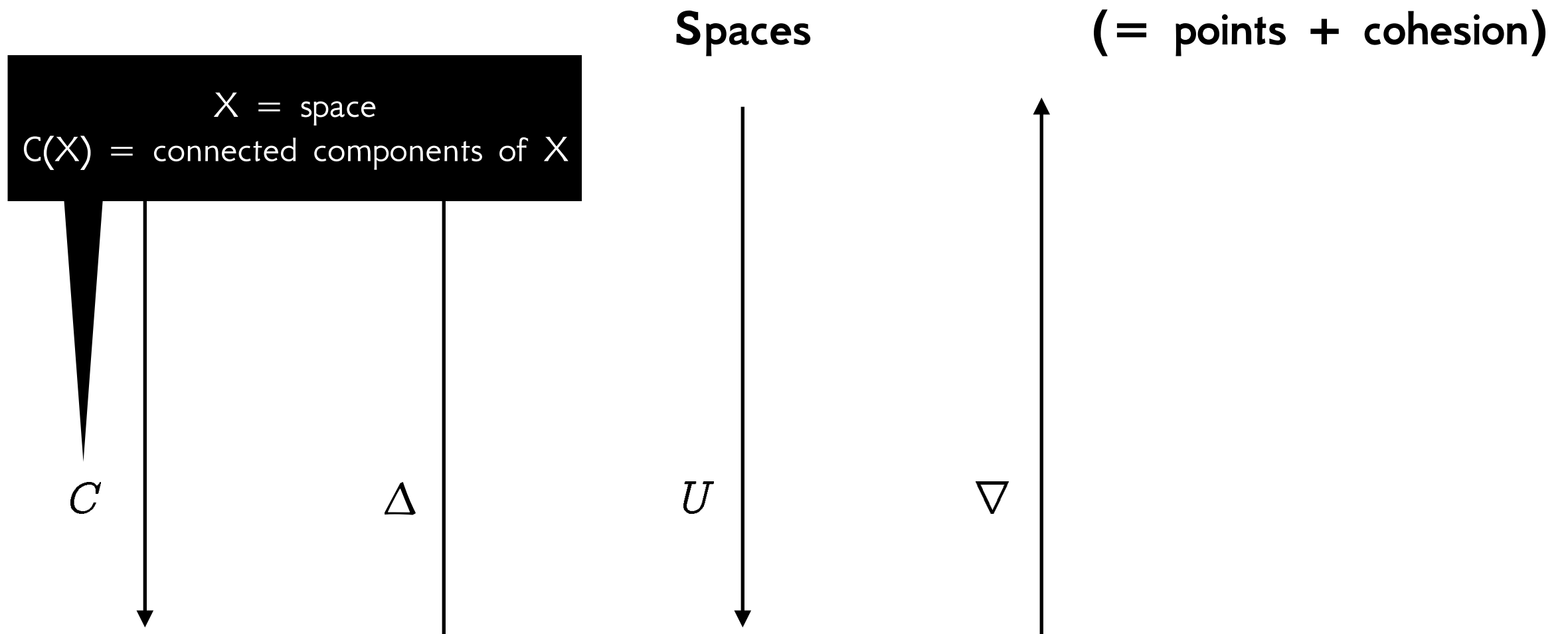
Cohesion



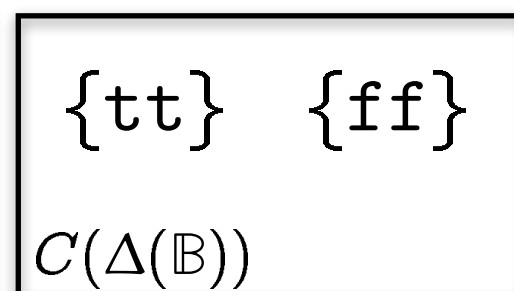
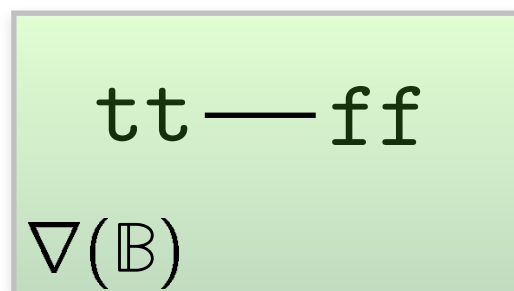
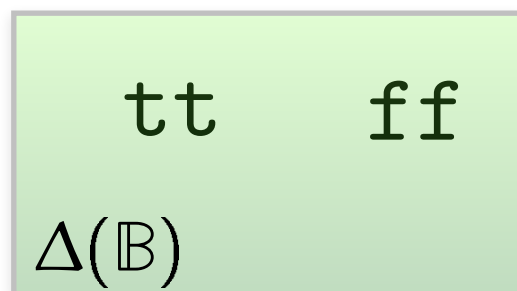
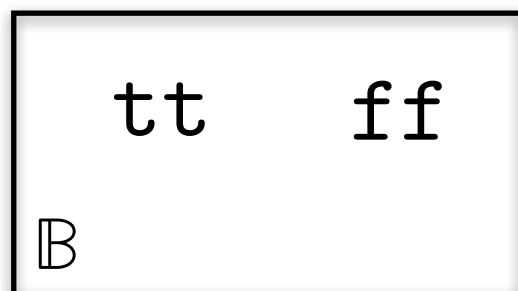
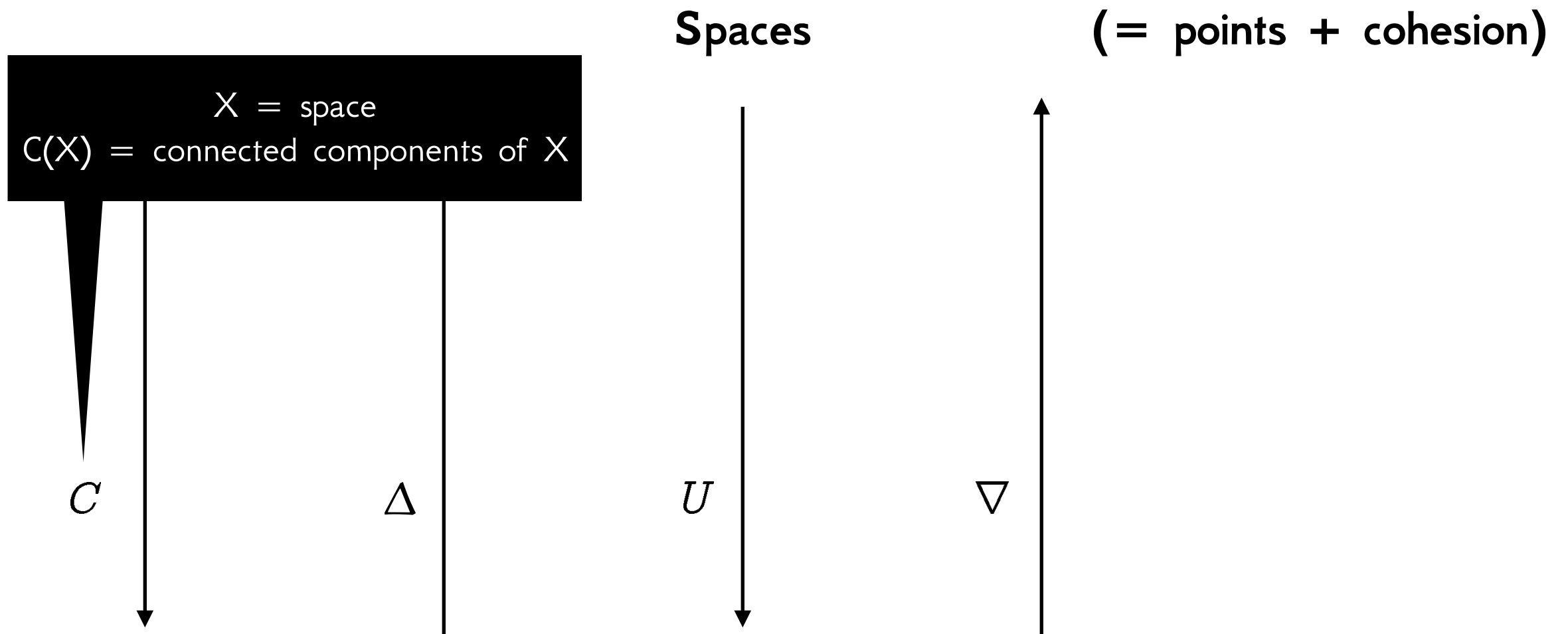
Cohesion



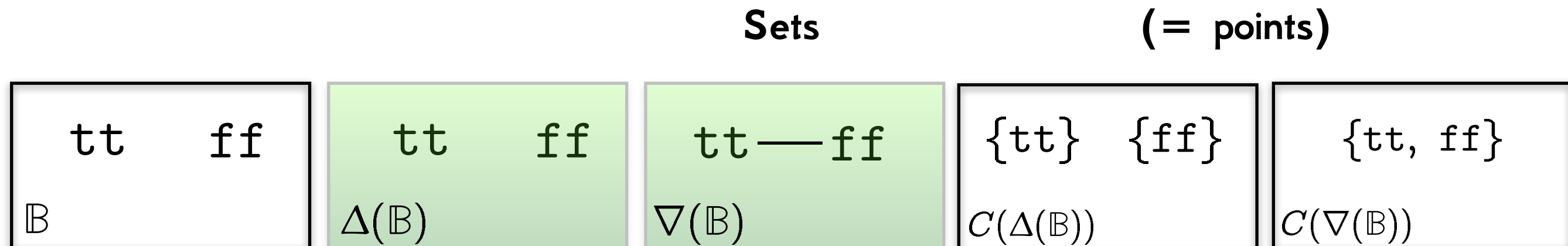
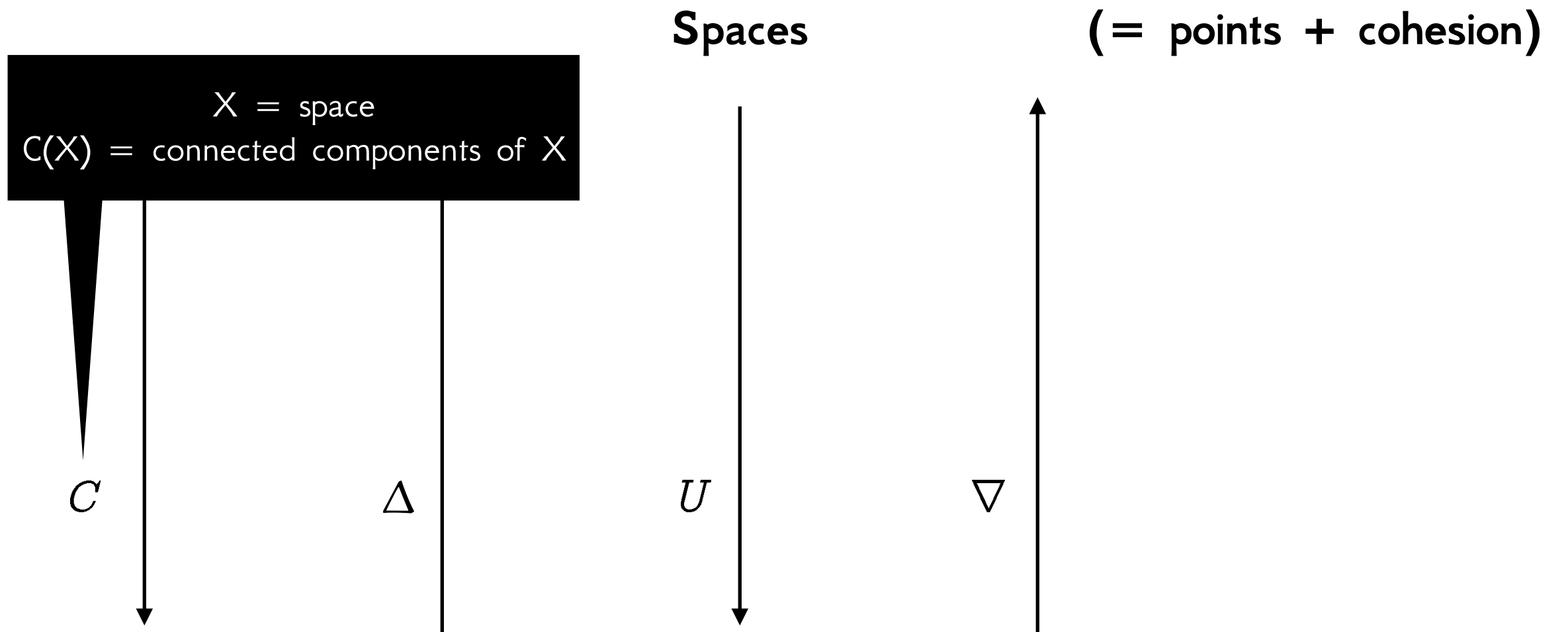
Cohesion



Cohesion



Cohesion



CRIB

$U(X)$ = points of space X
(forget cohesion)

$\Delta(S)$ = discrete space on S
(min. cohesion)

$\nabla(S)$ = codiscrete space on S
(max. cohesion)

$C(X)$ = connected components of X

Cohesion

Spaces

(= points + cohesion)

C

Δ

U

∇

Sets

(= points)

tt ff

\mathbb{B}

tt ff

$\Delta(\mathbb{B})$

tt — ff

$\nabla(\mathbb{B})$

$\{tt\}$ $\{ff\}$

$C(\Delta(\mathbb{B}))$

$\{tt, ff\}$

$C(\nabla(\mathbb{B}))$

CRIB

$U(X)$ = points of space X
(forget cohesion)

$\Delta(S)$ = discrete space on S
(min. cohesion)

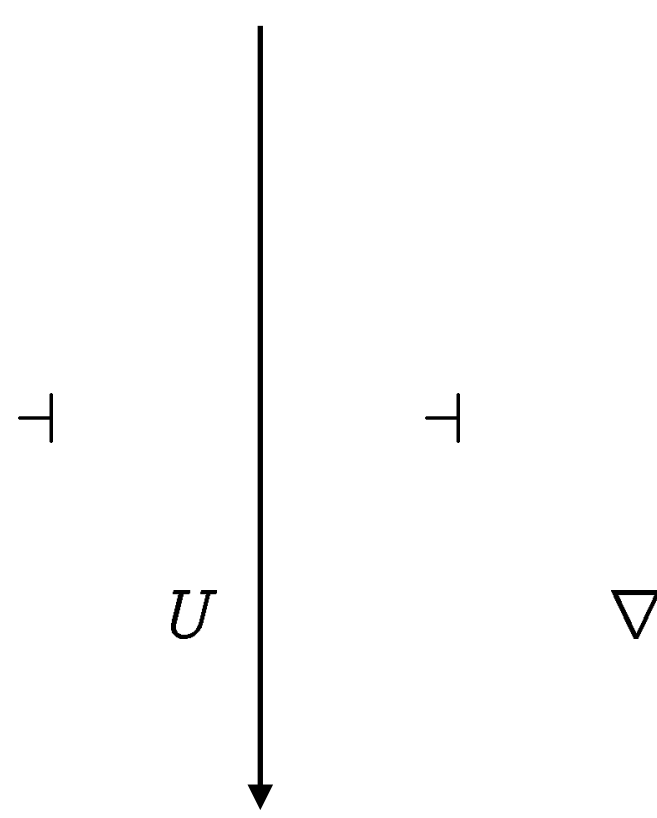
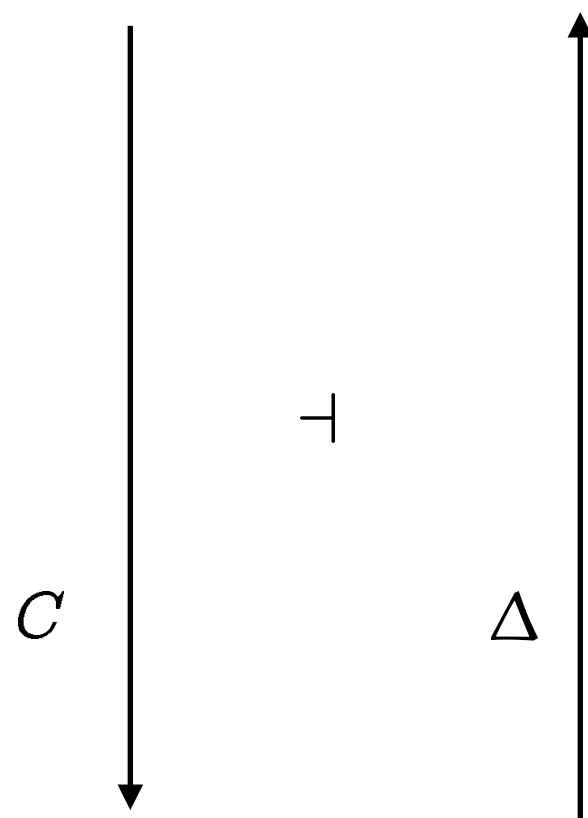
$\nabla(S)$ = codiscrete space on S
(max. cohesion)

$C(X)$ = connected components of X

Cohesion

Spaces

(= points + cohesion)



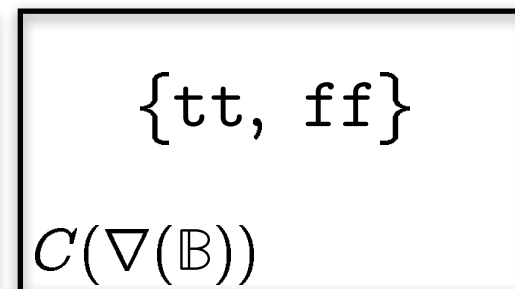
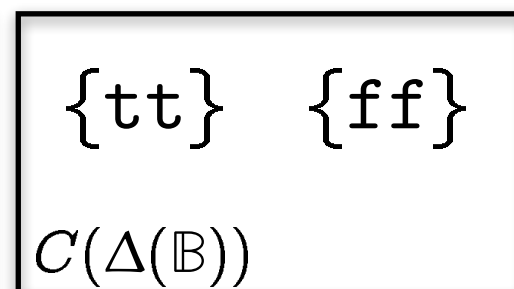
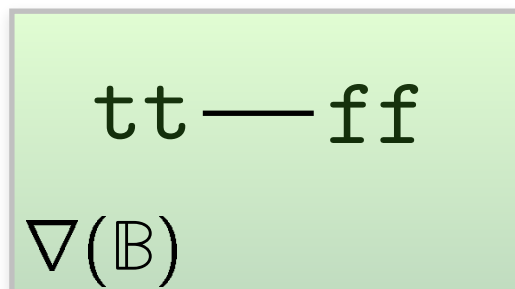
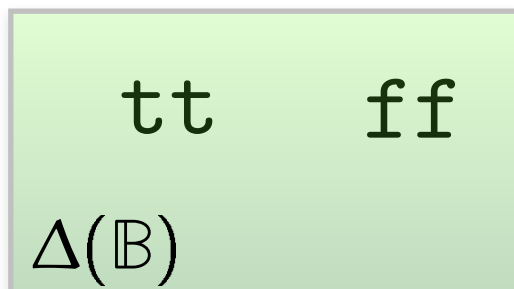
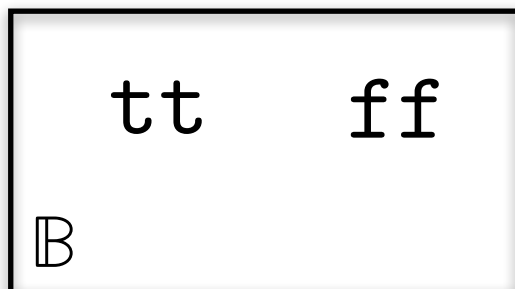
$$\frac{S \rightarrow U(X)}{\Delta(S) \rightarrow X}$$

$$\frac{U(X) \rightarrow S}{X \rightarrow \nabla(S)}$$

$$\frac{X \rightarrow \Delta(S)}{C(X) \rightarrow S}$$

Sets

(= points)



CRIB

$U(X)$ = points of space X
(forget cohesion)

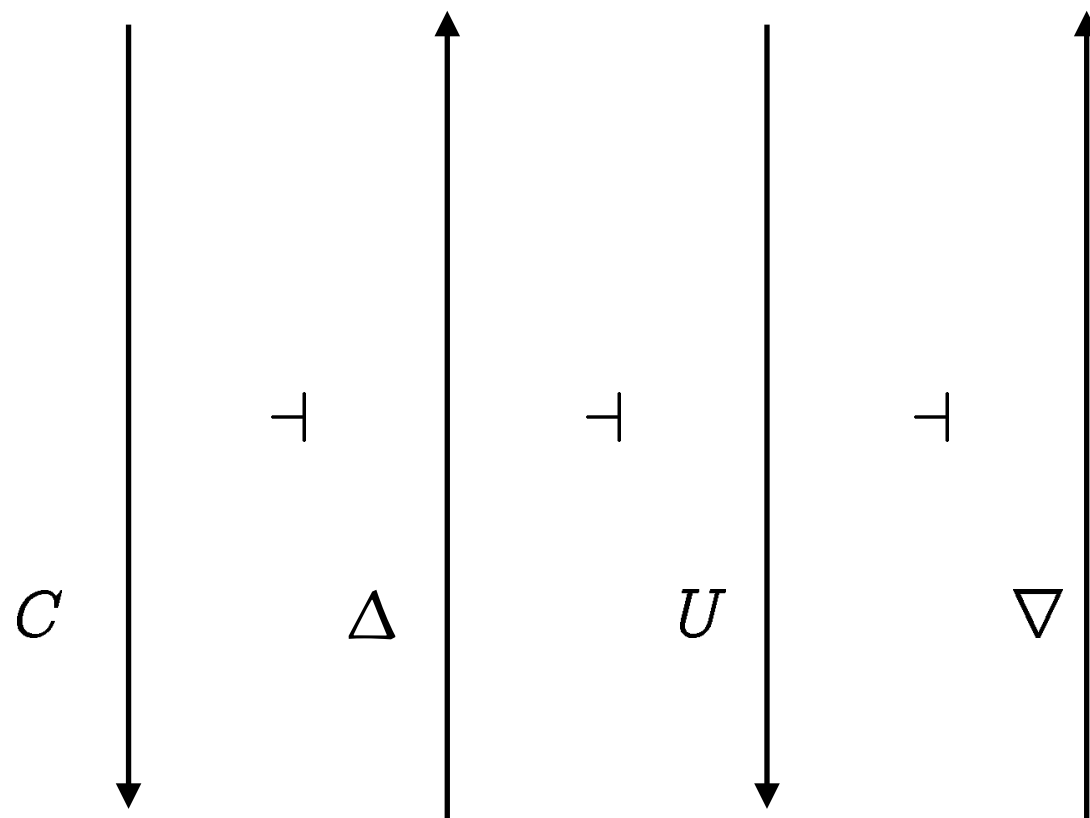
$\Delta(S)$ = discrete space on S
(min. cohesion)

$\nabla(S)$ = codiscrete space on S
(max. cohesion)

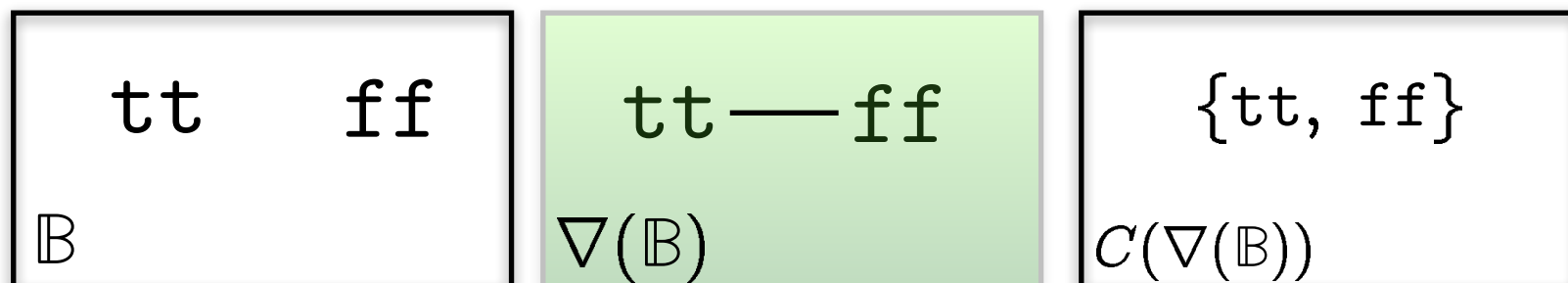
$C(X)$ = connected components of X

Cohesion

Spaces



Sets



CRIB

$U(X)$ = points of space X
(forget cohesion)

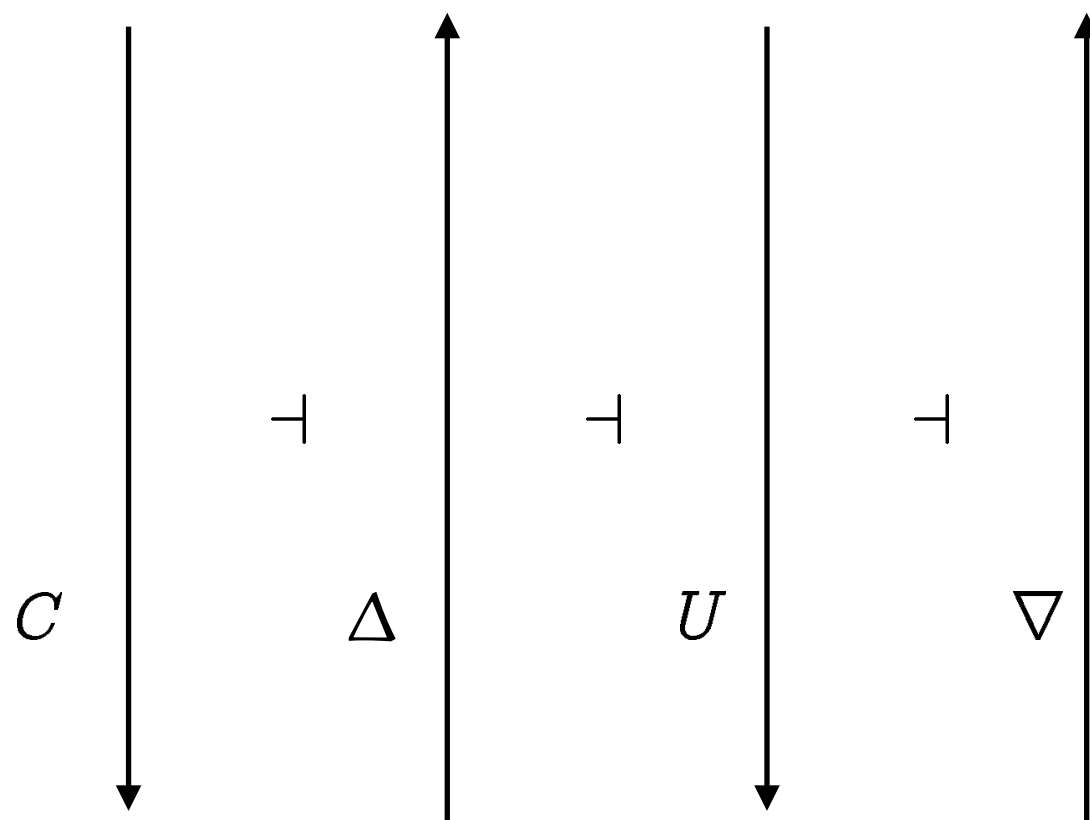
$\Delta(S)$ = discrete space on S
(min. cohesion)

$\nabla(S)$ = codiscrete space on S
(max. cohesion)

$C(X)$ = connected components of X

Cohesion

Spaces



Sets

tt ff
 \mathbb{B}

tt — ff
 $\nabla(\mathbb{B})$

$\{tt, ff\}$
 $C(\nabla(\mathbb{B}))$

in the codiscrete space $\nabla(S)$ on S
everything is “stuck together”
 \Rightarrow there is ≤ 1 connected component

CRIB

$U(X)$ = points of space X
(forget cohesion)

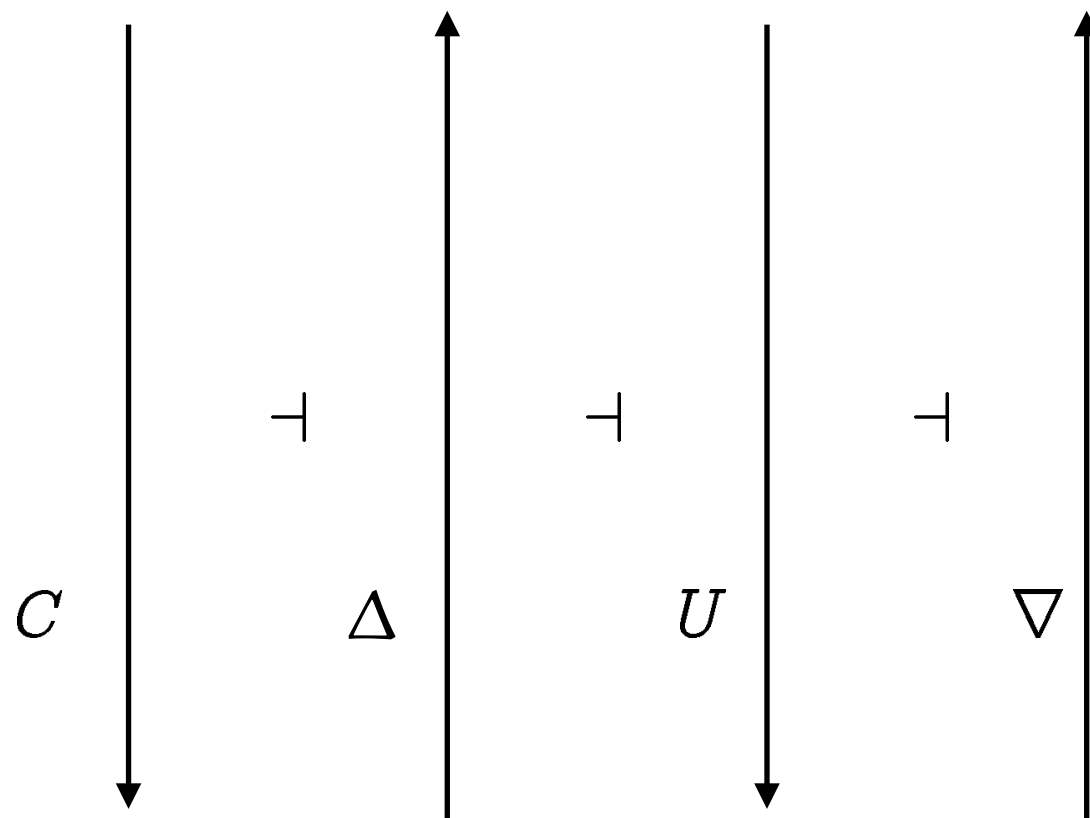
$\Delta(S)$ = discrete space on S
(min. cohesion)

$\nabla(S)$ = codiscrete space on S
(max. cohesion)

$C(X)$ = connected components of X

Cohesion

Spaces



Sets

**Axiom of
CONTRACTIBLE CODISCRETENESS:**

$$\forall S. |C(\nabla S)| \leq 1$$

(For category theorists: the
canonical $C(\nabla S) \xrightarrow{!} 1$
is a monic arrow.)

tt ff

\mathbb{B}

tt — ff

$\nabla(\mathbb{B})$

{tt, ff}

$C(\nabla(\mathbb{B}))$

in the codiscrete space $\nabla(S)$ on S
everything is “stuck together”
 \Rightarrow there is ≤ 1 connected component

Cohesion

CRIB

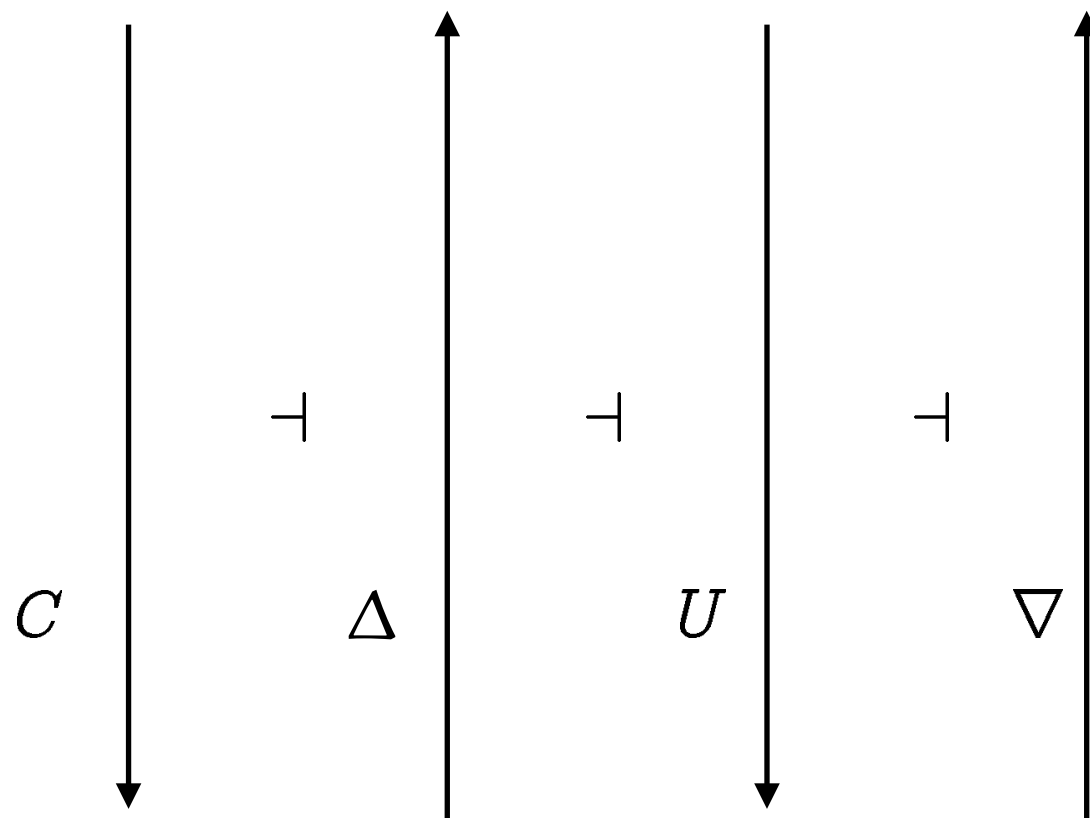
$U(X)$ = points of space X
(forget cohesion)

$\Delta(S)$ = discrete space on S
(min. cohesion)

$\nabla(S)$ = codiscrete space on S
(max. cohesion)

$C(X)$ = connected components of X

Spaces



Sets

CLAIM: This is all one needs to reason about information flow.

Axiom of CONTRACTIBLE CODISCRETENESS:

$$\forall S. |C(\nabla S)| \leq 1$$

(For category theorists: the canonical $C(\nabla S) \xrightarrow{!} 1$ is a monic arrow.)

tt ff

\mathbb{B}

tt — ff

$\nabla(\mathbb{B})$

{tt, ff}

$C(\nabla(\mathbb{B}))$

in the codiscrete space $\nabla(S)$ on S
everything is “stuck together”
 \Rightarrow there is ≤ 1 connected component

CRIB

$U(X)$ = points of space X
(forget cohesion)

$\Delta(S)$ = discrete space on S
(min. cohesion)

$\nabla(S)$ = codiscrete space on S
(max. cohesion)

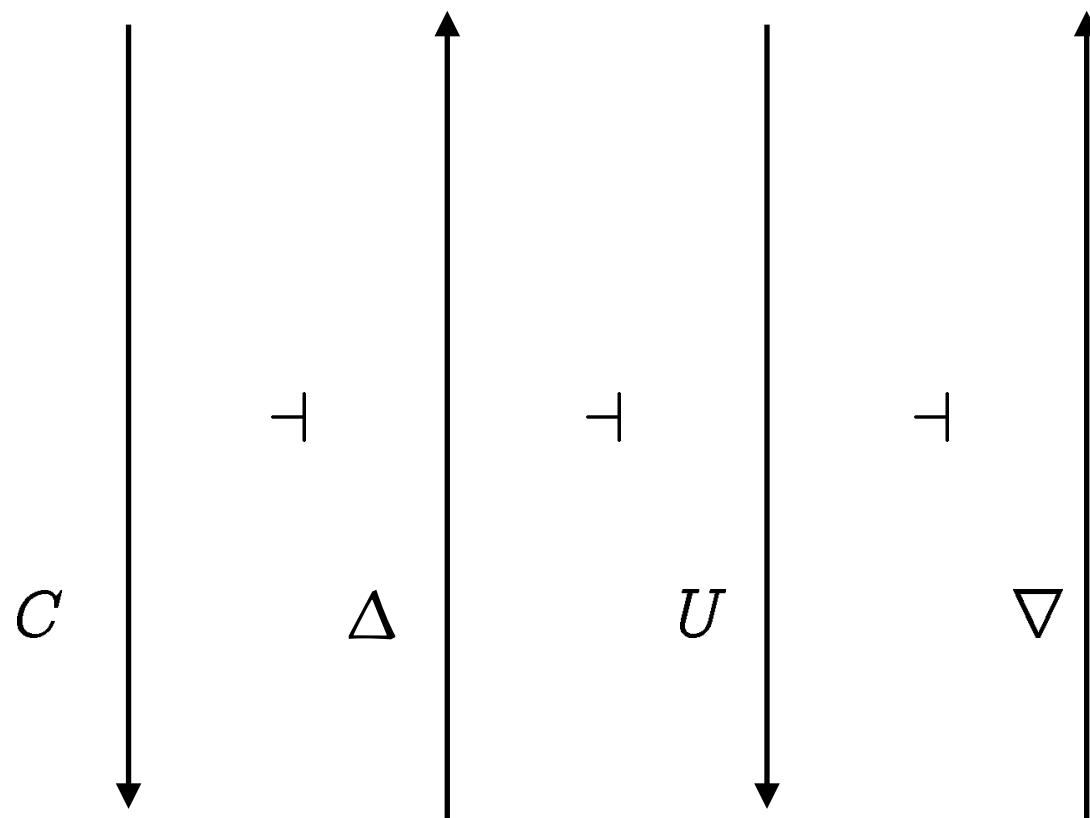
$C(X)$ = connected components of X

Cohesion

CLAIM: This is all one needs to reason about information flow.

$$\forall S. |C(\nabla S)| \leq 1$$

Spaces



Sets

tt ff
 \mathbb{B}

tt — ff
 $\nabla(\mathbb{B})$

{tt, ff}
 $C(\nabla(\mathbb{B}))$

in the codiscrete space $\nabla(S)$ on S
everything is “stuck together”
 \Rightarrow there is ≤ 1 connected component

CRIB

$U(X)$ = points of space X
(forget cohesion)

$\Delta(S)$ = discrete space on S
(min. cohesion)

$\nabla(S)$ = codiscrete space on S
(max. cohesion)

$C(X)$ = connected components of X

Cohesion

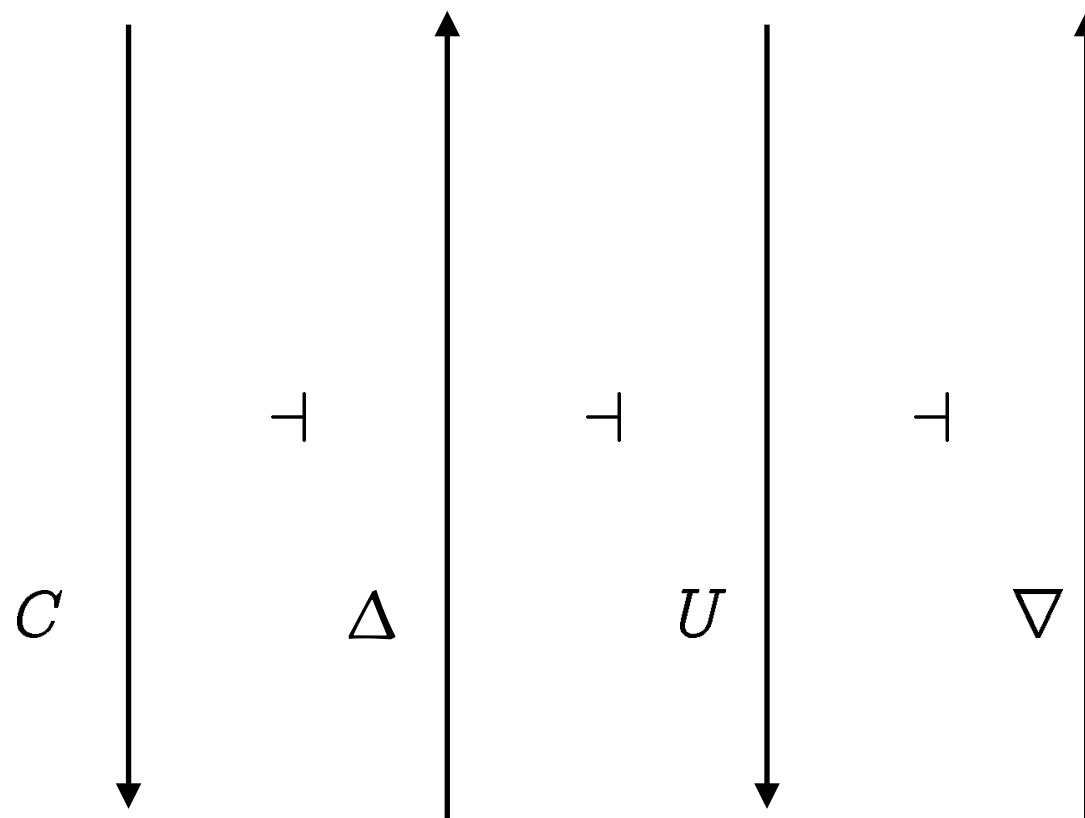
CLAIM: This is all one needs to reason about information flow.

$$\forall S. |C(\nabla S)| \leq 1$$

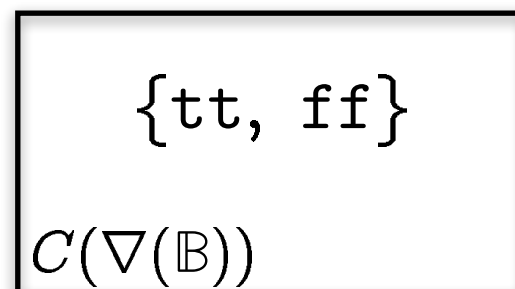
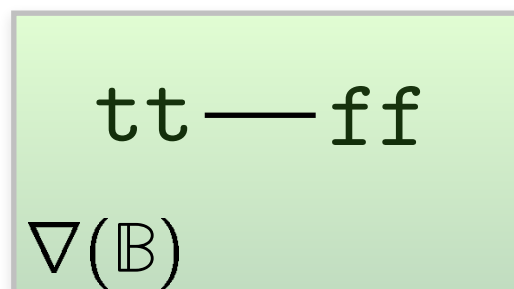
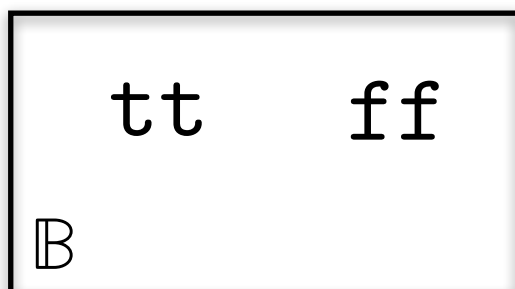
"X redacted"

Define: $\blacklozenge X = \nabla(UX)$

Spaces



Sets



in the codiscrete space $\nabla(S)$ on S
everything is "stuck together"
 \Rightarrow there is ≤ 1 connected component

CRIB

$U(X)$ = points of space X
(forget cohesion)

$\Delta(S)$ = discrete space on S
(min. cohesion)

$\nabla(S)$ = codiscrete space on S
(max. cohesion)

$C(X)$ = connected components of X

Cohesion

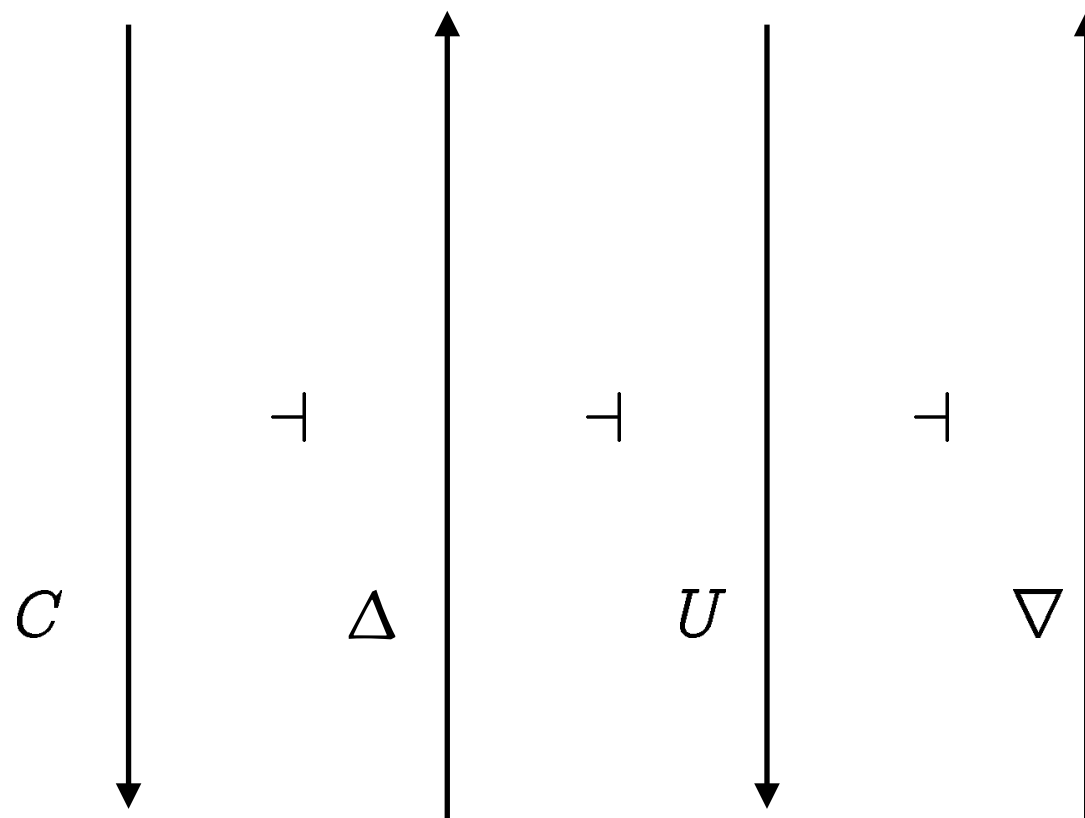
CLAIM: This is all one needs to reason about information flow.

$$\forall S. |C(\nabla S)| \leq 1$$

"X redacted"

Define: $\blacklozenge X = \nabla(UX)$

Spaces



Sets

Theorem: every $f : \blacklozenge X \rightarrow \Delta S$ continuous is a point of S (maybe)

tt ff

\mathbb{B}

tt — ff

$\nabla(\mathbb{B})$

$\{tt, ff\}$

$C(\nabla(\mathbb{B}))$

in the codiscrete space $\nabla(S)$ on S
everything is "stuck together"
 \Rightarrow there is ≤ 1 connected component

CRIB

$U(X)$ = points of space X
(forget cohesion)

$\Delta(S)$ = discrete space on S
(min. cohesion)

$\nabla(S)$ = codiscrete space on S
(max. cohesion)

$C(X)$ = connected components of X

Cohesion

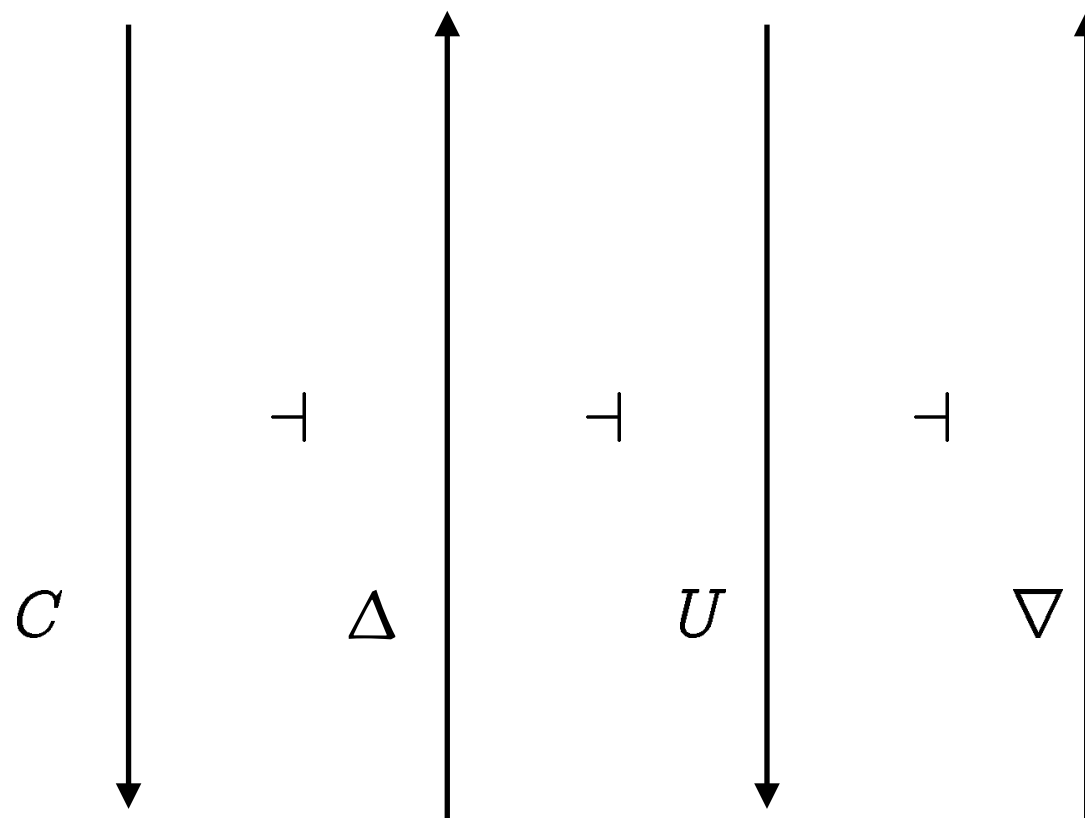
CLAIM: This is all one needs to reason about information flow.

$$\forall S. |C(\nabla S)| \leq 1$$

"X redacted"

Define: $\blacklozenge X = \nabla(UX)$

Spaces



Sets

Theorem: every $f : \blacklozenge X \rightarrow \Delta S$ continuous is a point of S (maybe)

Proof: every such f is by def. a continuous function
 $f : \nabla(UX) \rightarrow \Delta S$

$tt \quad ff$

\mathbb{B}

$tt \text{ — } ff$

$\nabla(\mathbb{B})$

$\{tt, ff\}$

$C(\nabla(\mathbb{B}))$

in the codiscrete space $\nabla(S)$ on S
everything is "stuck together"
 \Rightarrow there is ≤ 1 connected component

CRIB

$U(X)$ = points of space X
(forget cohesion)

$\Delta(S)$ = discrete space on S
(min. cohesion)

$\nabla(S)$ = codiscrete space on S
(max. cohesion)

$C(X)$ = connected components of X

Cohesion

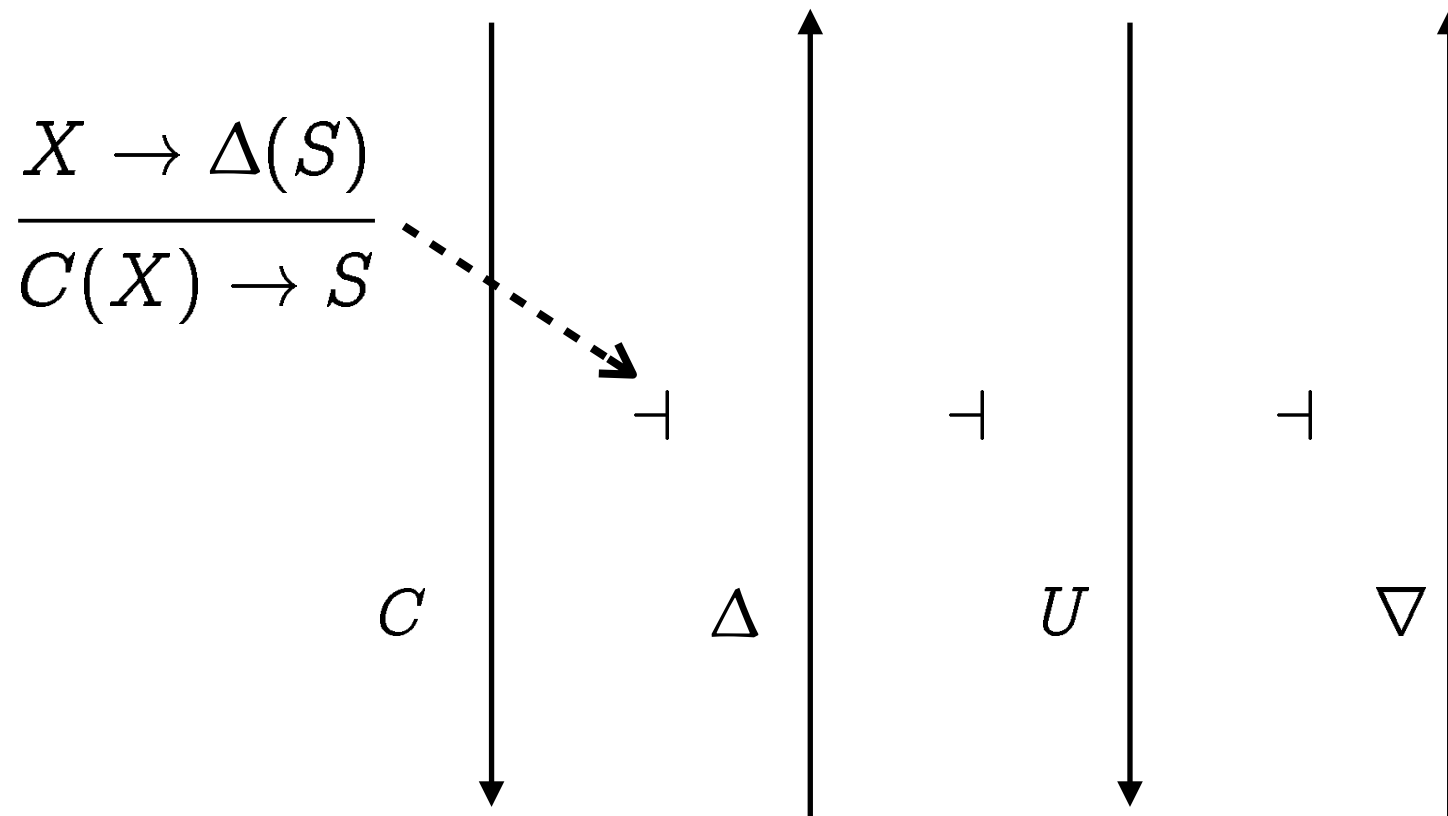
CLAIM: This is all one needs to reason about information flow.

$$\forall S. |C(\nabla S)| \leq 1$$

"X redacted"

Define: $\blacklozenge X = \nabla(UX)$

Spaces



Theorem: every $f : \blacklozenge X \rightarrow \Delta S$ continuous is a point of S (maybe)

Proof: every such f is by def. a continuous function $f : \nabla(UX) \rightarrow \Delta S$

Sets

$tt \quad ff$

\mathbb{B}

$tt \text{ — } ff$

$\nabla(\mathbb{B})$

$\{tt, ff\}$

$C(\nabla(\mathbb{B}))$

in the codiscrete space $\nabla(S)$ on S
everything is "stuck together"
 \Rightarrow there is ≤ 1 connected component

CRIB

$U(X)$ = points of space X
(forget cohesion)

$\Delta(S)$ = discrete space on S
(min. cohesion)

$\nabla(S)$ = codiscrete space on S
(max. cohesion)

$C(X)$ = connected components of X

Cohesion

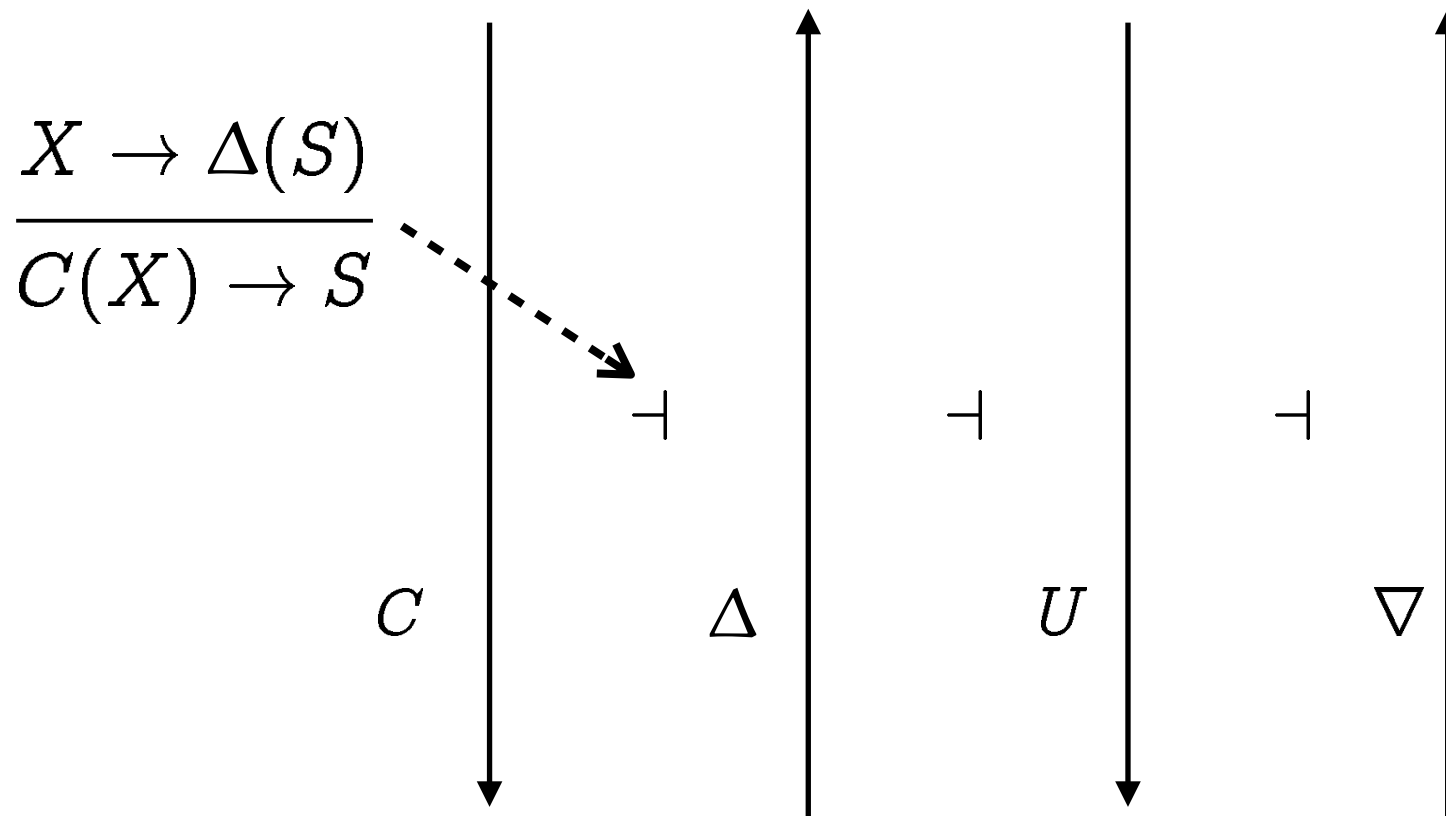
CLAIM: This is all one needs to reason about information flow.

$$\forall S. |C(\nabla S)| \leq 1$$

"X redacted"

Define: $\blacklozenge X = \nabla(UX)$

Spaces



Sets

Theorem: every $f : \blacklozenge X \rightarrow \Delta S$ continuous is a point of S (maybe)

Proof: every such f is by def. a continuous function

$$f : \nabla(UX) \rightarrow \Delta S$$

which is just a set function

$$f : C(\nabla(UX)) \rightarrow S$$

$tt \quad ff$

\mathbb{B}

$tt \text{ — } ff$

$\nabla(\mathbb{B})$

$\{tt, ff\}$

$C(\nabla(\mathbb{B}))$

in the codiscrete space $\nabla(S)$ on S everything is "stuck together"

\Rightarrow there is ≤ 1 connected component

CRIB

$U(X)$ = points of space X
(forget cohesion)

$\Delta(S)$ = discrete space on S
(min. cohesion)

$\nabla(S)$ = codiscrete space on S
(max. cohesion)

$C(X)$ = connected components of X

Cohesion

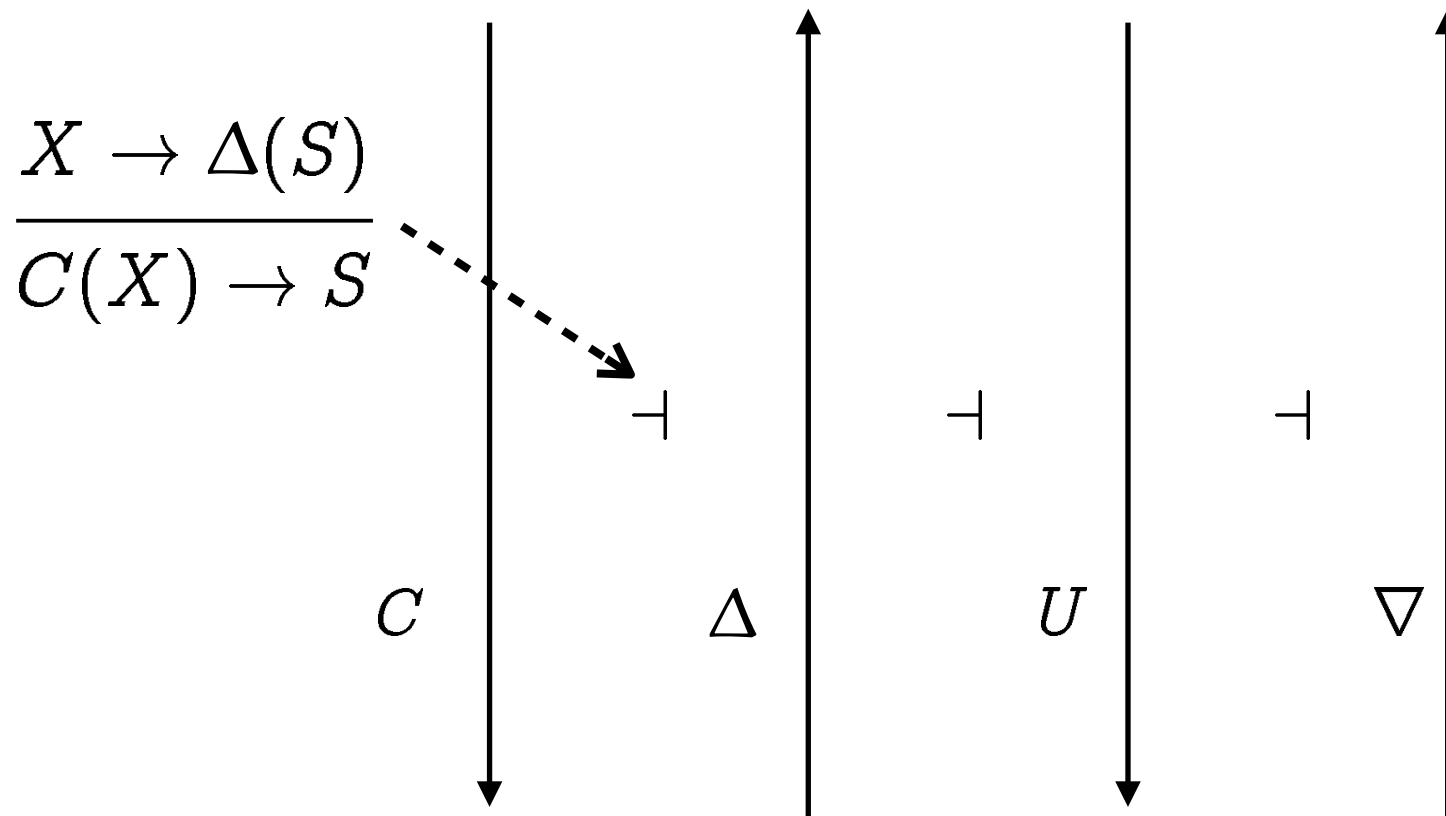
CLAIM: This is all one needs to reason about information flow.

$$\forall S. |C(\nabla S)| \leq 1$$

"X redacted"

Define: $\blacklozenge X = \nabla(UX)$

Spaces



Sets

Theorem: every $f : \blacklozenge X \rightarrow \Delta S$ continuous is a point of S (maybe)

Proof: every such f is by def. a continuous function

$$f : \nabla(UX) \rightarrow \Delta S$$

which is just a set function

$$f : C(\nabla(UX)) \rightarrow S$$

$tt \quad ff$

\mathbb{B}

$tt \text{ — } ff$

$\nabla(\mathbb{B})$

$\{tt, ff\}$

$C(\nabla(\mathbb{B}))$

in the codiscrete space $\nabla(S)$ on S
everything is "stuck together"
 \Rightarrow there is ≤ 1 connected component

CRIB

$U(X)$ = points of space X
(forget cohesion)

$\Delta(S)$ = discrete space on S
(min. cohesion)

$\nabla(S)$ = codiscrete space on S
(max. cohesion)

$C(X)$ = connected components of X

Cohesion

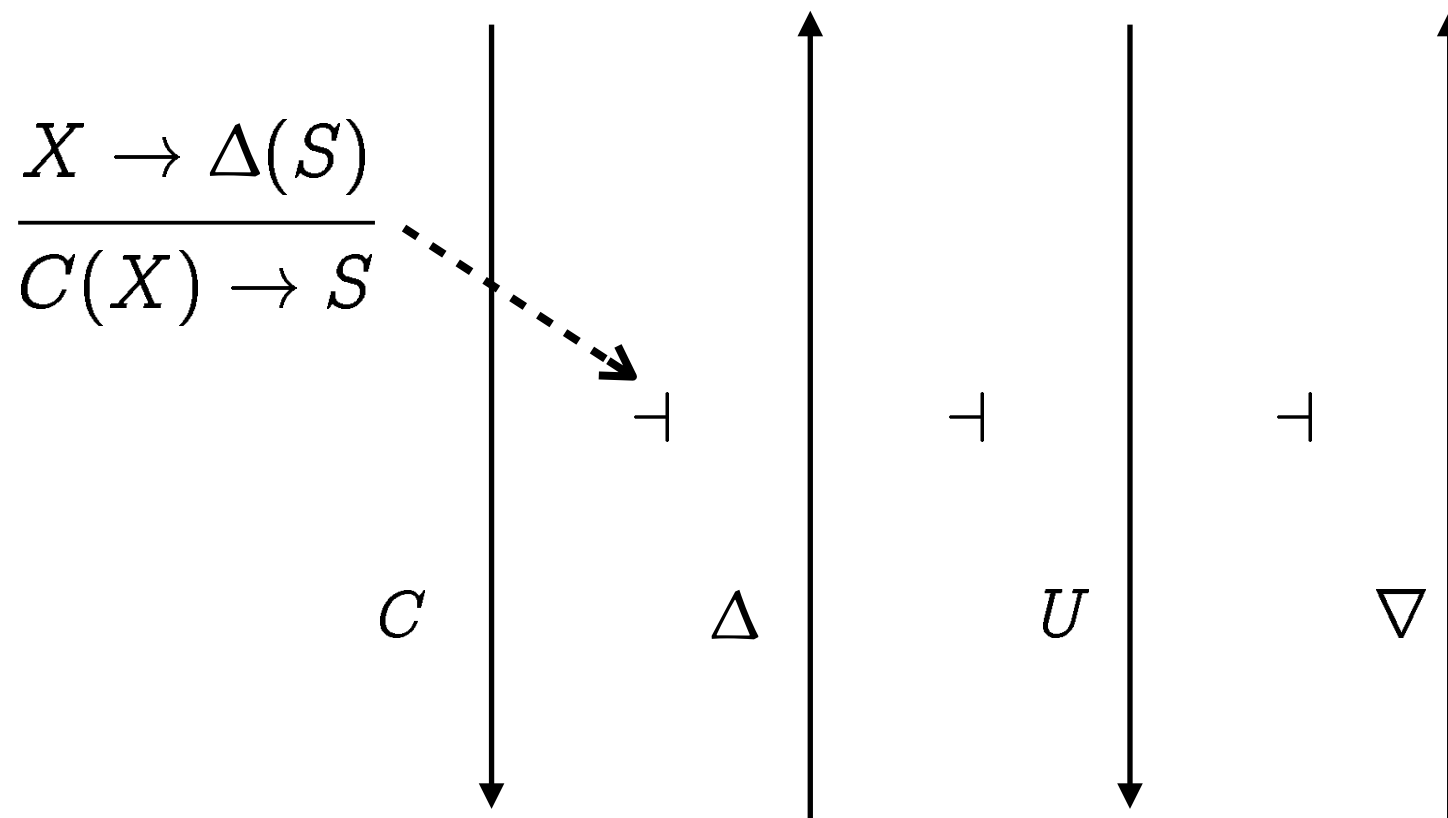
CLAIM: This is all one needs to reason about information flow.

$$\forall S. |C(\nabla S)| \leq 1$$

"X redacted"

Define: $\blacklozenge X = \nabla(UX)$

Spaces



Theorem: every $f : \blacklozenge X \rightarrow \Delta S$ continuous is a point of S (maybe)

Proof: every such f is by def. a continuous function

$$f : \nabla(UX) \rightarrow \Delta S$$

which is just a set function

$$f : \underbrace{C(\nabla(UX))}_{\text{a set of } \leq 1 \text{ element!}} \rightarrow S$$

a set of ≤ 1 element!

Sets

tt ff

\mathbb{B}

tt — ff

$\nabla(\mathbb{B})$

$\{tt, ff\}$

$C(\nabla(\mathbb{B}))$

in the codiscrete space $\nabla(S)$ on S
everything is "stuck together"

\Rightarrow there is ≤ 1 connected component

CRIB

$U(X)$ = points of space X
(forget cohesion)

$\Delta(S)$ = discrete space on S
(min. cohesion)

$\nabla(S)$ = codiscrete space on S
(max. cohesion)

$C(X)$ = connected components of X

Classified sets

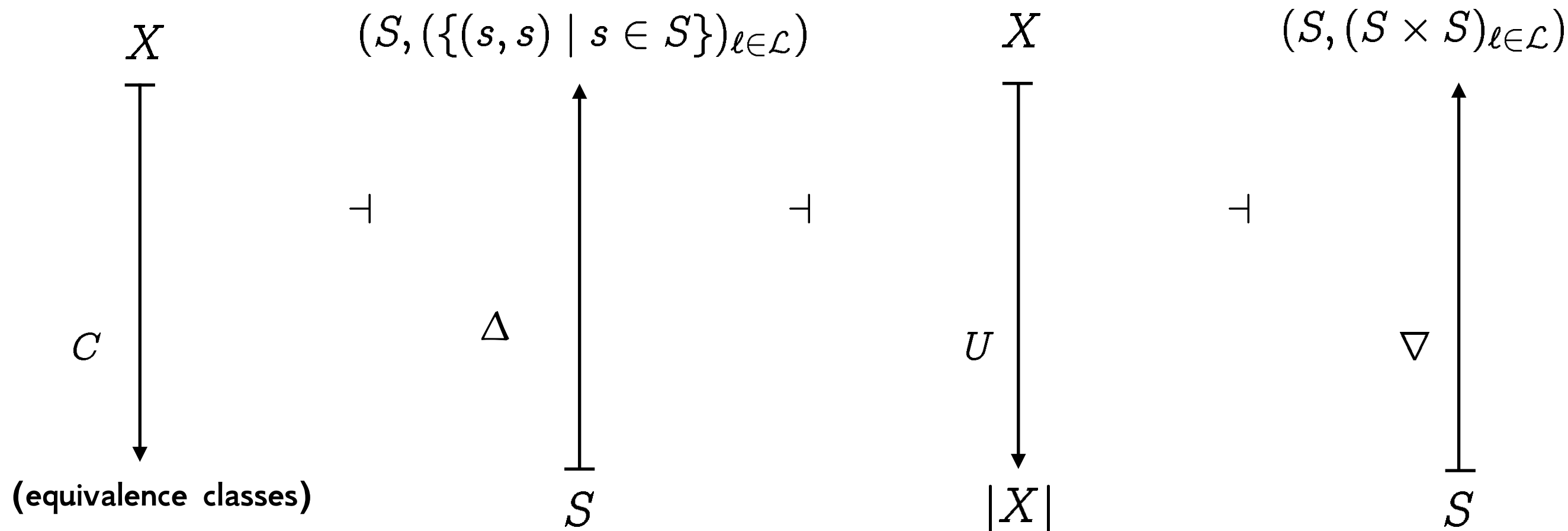
Set of classifications/labels: $\ell \in \mathcal{L}$

must be reflexive

Classified set: $X = (|X|, (R_\ell \subseteq |X| \times |X|)_{\ell \in \mathcal{L}})$

Cont. function: $f : X \rightarrow Y$ s.t. $\forall \ell. a R_\ell b \Rightarrow f(a) R_\ell f(b)$

"f is continuous when it maps inputs indistinguishable at $\ell \in \mathcal{L}$ to outputs indistinguishable at $\ell \in \mathcal{L}$ "



CRIB

$U(X)$ = points of space X
(forget cohesion)

$\Delta(S)$ = discrete space on S
(min. cohesion)

$\nabla(S)$ = codiscrete space on S
(max. cohesion)

$C(X)$ = connected components of X

Classified sets

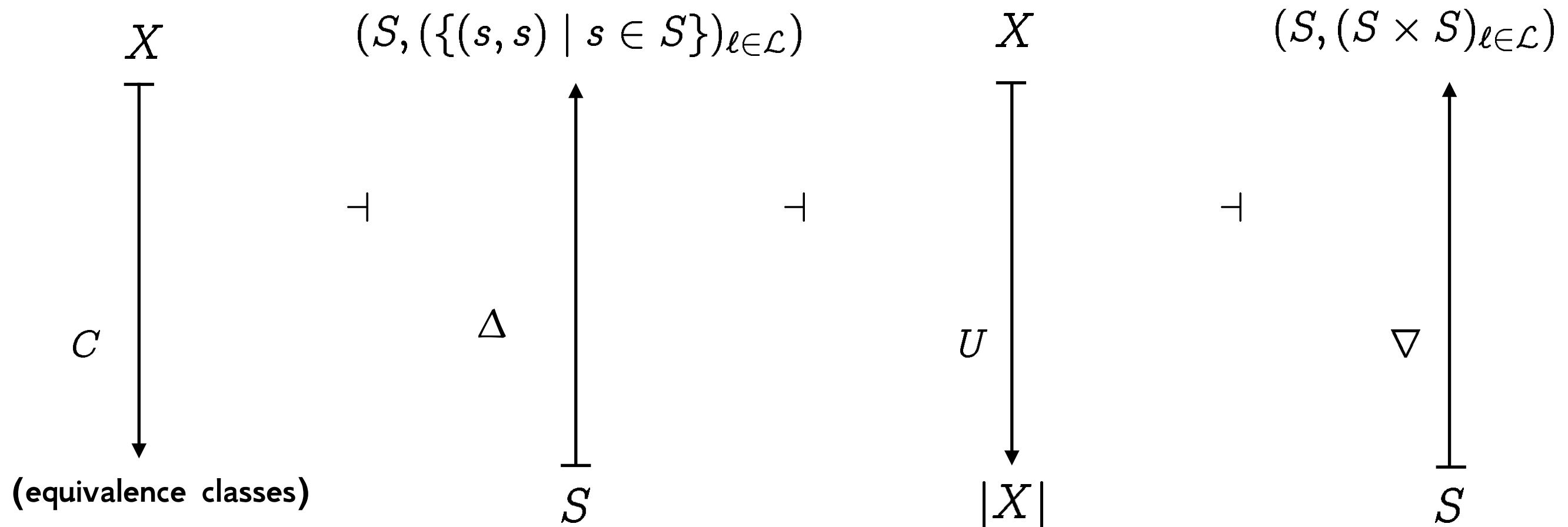
Set of classifications/labels: $\ell \in \mathcal{L}$

must be **reflexive**

Classified set: $X = (|X|, (R_\ell \subseteq |X| \times |X|)_{\ell \in \mathcal{L}})$

Cont. function: $f : X \rightarrow Y$ s.t. $\forall \ell. a R_\ell b \Rightarrow f(a) R_\ell f(b)$

"f is continuous when it maps inputs indistinguishable at $\ell \in \mathcal{L}$ to outputs indistinguishable at $\ell \in \mathcal{L}$ "



Theorem: the category of classified sets is **cartesian closed** and **cohesive over Sets**,
and it satisfies **contractible codiscreteness**.

CRIB

$U(X)$ = points of space X
(forget cohesion)

$\Delta(S)$ = discrete space on S
(min. cohesion)

$\nabla(S)$ = codiscrete space on S
(max. cohesion)

$C(X)$ = connected components of X

Classified sets

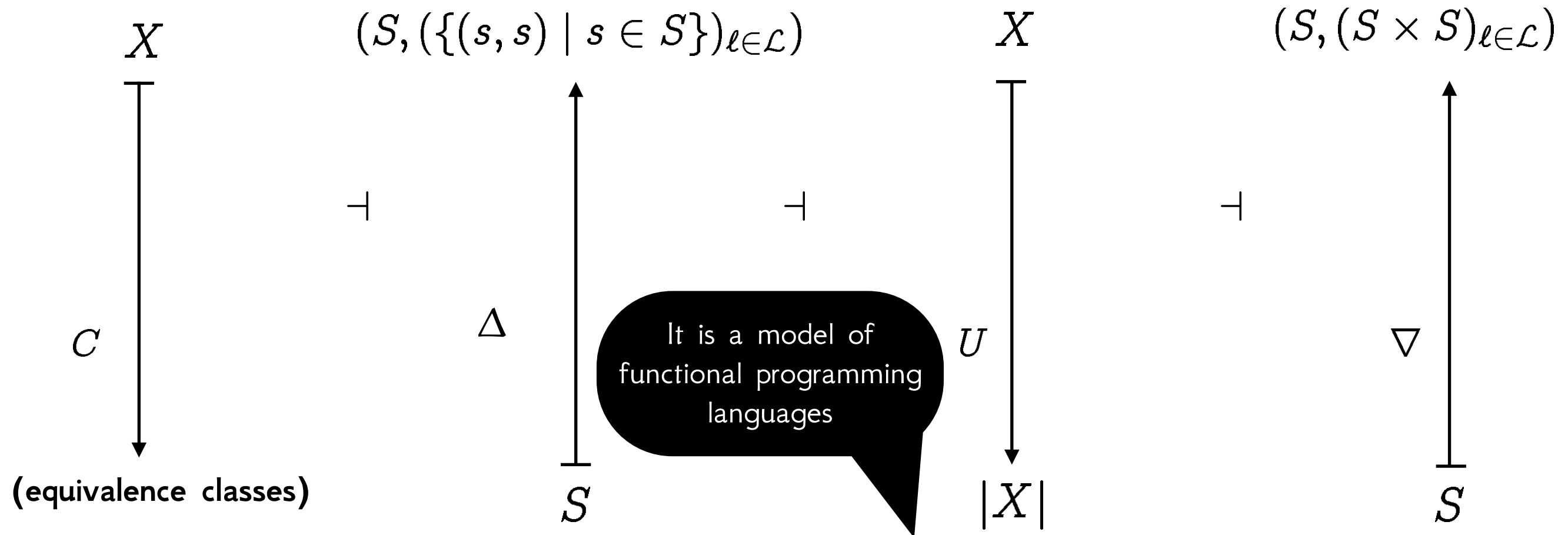
Set of classifications/labels: $\ell \in \mathcal{L}$

must be **reflexive**

Classified set: $X = (|X|, (R_\ell \subseteq |X| \times |X|)_{\ell \in \mathcal{L}})$

Cont. function: $f : X \rightarrow Y$ s.t. $\forall \ell. a R_\ell b \Rightarrow f(a) R_\ell f(b)$

"f is continuous when it maps inputs indistinguishable at $\ell \in \mathcal{L}$ to outputs indistinguishable at $\ell \in \mathcal{L}$ "



Theorem: the category of classified sets is **cartesian closed** and **cohesive over Sets**, and it satisfies **contractible codiscreteness**.

CRIB

$U(X)$ = points of space X
(forget cohesion)

$\Delta(S)$ = discrete space on S
(min. cohesion)

$\nabla(S)$ = codiscrete space on S
(max. cohesion)

$C(X)$ = connected components of X

Classified sets

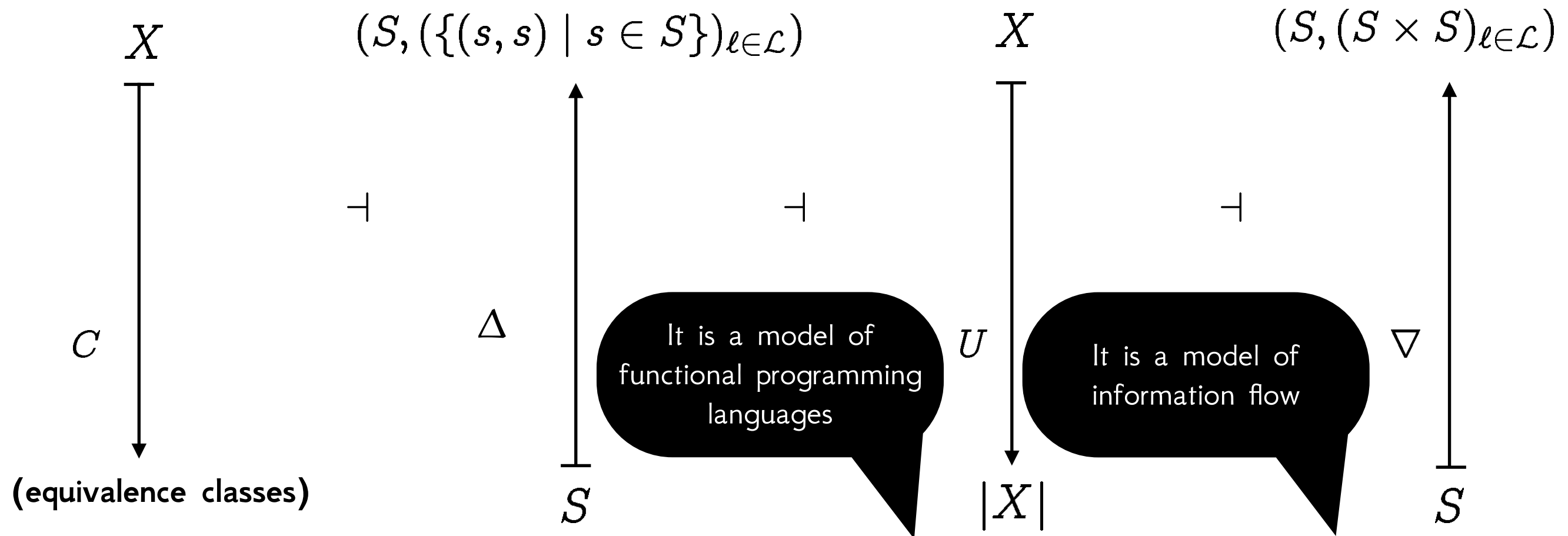
Set of classifications/labels: $\ell \in \mathcal{L}$

must be **reflexive**

Classified set: $X = (|X|, (R_\ell \subseteq |X| \times |X|)_{\ell \in \mathcal{L}})$

Cont. function: $f : X \rightarrow Y$ s.t. $\forall \ell. a R_\ell b \Rightarrow f(a) R_\ell f(b)$

"f is continuous when it maps inputs indistinguishable at $\ell \in \mathcal{L}$ to outputs indistinguishable at $\ell \in \mathcal{L}$ "



Theorem: the category of classified sets is **cartesian closed** and **cohesive over Sets**,
and it satisfies **contractible codiscreteness**.

Cohesion and non-interference

❖ Recall what we were trying to prove:

If $x : \blacklozenge A \vdash E : \text{Bool}$ and $\vdash M, N : \blacklozenge A$ then
 $E[M/x]$ and $E[N/x]$ compute the same boolean value.

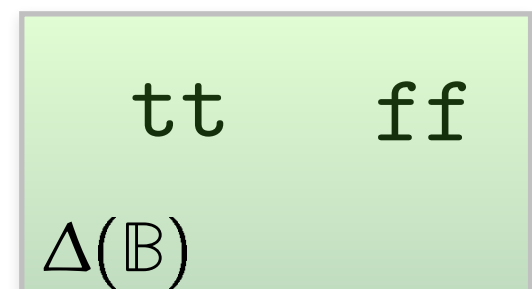
❖ There is a way to map every term to a continuous function between classified sets — a **categorical semantics**:

$$x : \blacklozenge A \vdash E : \text{Bool} \quad \longmapsto \quad \llbracket E \rrbracket : \blacklozenge \llbracket A \rrbracket \rightarrow \Delta \mathbb{B}$$

❖ By the Theorem, this corresponds to an element of \mathbb{B}
So it is essentially a constant function.

$$\blacklozenge X = \nabla(UX)$$

❖ If only this could tell us something about the language...



Cohesion and non-interference

- ❖ Recall what we were trying to prove:

If $x : \blacklozenge A \vdash E : \text{Bool}$ and $\vdash M, N : \blacklozenge A$ then
 $E[M/x]$ and $E[N/x]$ compute the same boolean value.

- ❖ There is a way to map every term to a continuous function between classified sets — a **categorical semantics**:

$$x : \blacklozenge A \vdash E : \text{Bool} \quad \longmapsto \quad \llbracket E \rrbracket : \blacklozenge \llbracket A \rrbracket \rightarrow \Delta \mathbb{B}$$

- ❖ By the Theorem, this corresponds to an element of \mathbb{B}
So it is essentially a constant function.

- ❖ If only this could tell us something about the language...

ADEQUACY, a.k.a
completeness at base types
(automatically holds
when the language has
no recursion; easy to
establish for “algebraic
effects”)

Cohesion and non-interference

- ❖ Recall what we were trying to prove:

If $x : \blacklozenge A \vdash E : \text{Bool}$ and $\vdash M, N : \blacklozenge A$ then
 $E[M/x]$ and $E[N/x]$ compute the same boolean value.



- ❖ There is a way to map every term to a continuous function between classified sets — a **categorical semantics**:

$$x : \blacklozenge A \vdash E : \text{Bool} \quad \longmapsto \quad \llbracket E \rrbracket : \blacklozenge \llbracket A \rrbracket \rightarrow \Delta \mathbb{B}$$

- ❖ By the Theorem, this corresponds to an element of \mathbb{B}
So it is essentially a constant function.

- ❖ If only this could tell us something about the language...

ADEQUACY, a.k.a
completeness at base types
(automatically holds
when the language has
no recursion; easy to
establish for “algebraic
effects”)

Cohesion and non-interference

- ❖ This approach can be leveraged to prove noninterference for multiple type theories for secure information flow:
 - ❖ Moggi's monadic metalanguage [Moggi 1991]
 - ❖ Davies-Pfenning calculus (S4 modality) [D&Pf 2001]
 - ❖ Dependency Core Calculus [Abadi et al. 1999]
 - ❖ Sealing Calculus [Shikuma & Igarashi 2008]
- ❖ The last two are **multi-modal** type theories.

} (A little bit of care is required here w.r.t. adequacy)

Cohesion and multi-modal type theories for information flow

Writing \mathbf{CSet}_π for the category of classified sets over $\pi \subseteq \mathcal{L}$

and $\alpha : \pi \subseteq \pi'$
 $\beta : \pi' \subseteq \pi''$ for the

unique morphisms in $\mathcal{P}(\mathcal{L})$
 we have the two cohesive
 situation on the right.

It's a functor

$$\mathcal{P}(\mathcal{L})^{\text{op}} \longrightarrow \mathbf{Coh}$$

$$\begin{array}{ccccc}
 & & \mathbf{CSet}_{\pi''} & & \\
 & \downarrow C_\beta & \uparrow \Delta_\beta & \downarrow U_\beta & \uparrow \nabla_\beta \\
 & \vdash & & \vdash & \\
 & \mathbf{CSet}_{\pi'} & & \mathbf{CSet}_{\pi'} & \\
 & \downarrow C_\alpha & \uparrow \Delta_\alpha & \downarrow U_\alpha & \uparrow \nabla_\alpha \\
 & \vdash & & \vdash & \\
 & \mathbf{CSet}_\pi & & \mathbf{CSet}_\pi &
 \end{array}$$

$$\mathcal{L} \cup \pi$$

$$\mathcal{L}$$

Cohesion and multi-modal type theories for information flow

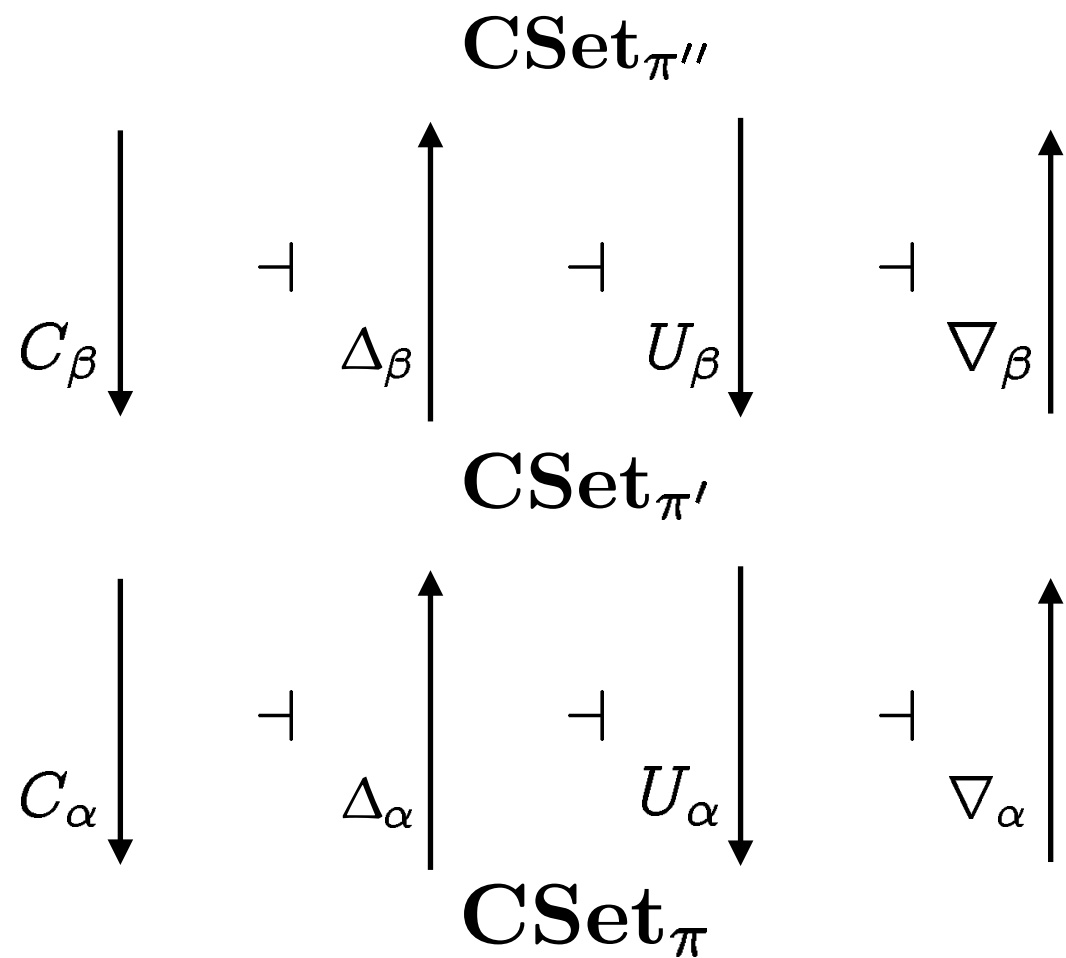
Writing \mathbf{CSet}_π for the category of classified sets over $\pi \subseteq \mathcal{L}$

and $\alpha : \pi \subseteq \pi'$
 $\beta : \pi' \subseteq \pi''$ for the

unique morphisms in $\mathcal{P}(\mathcal{L})$
 we have the two cohesive
 situation on the right.

It's a functor

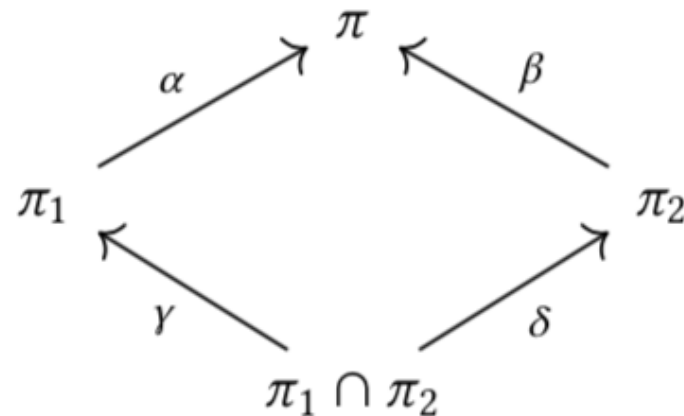
$$\mathcal{P}(\mathcal{L})^{\text{op}} \longrightarrow \mathbf{Coh}$$



Theorem: the category of classified sets over $\mathcal{L} \cup \pi$ is **cohesive** over the category of classified sets over \mathcal{L} and satisfies **contractible codiscreteness**.

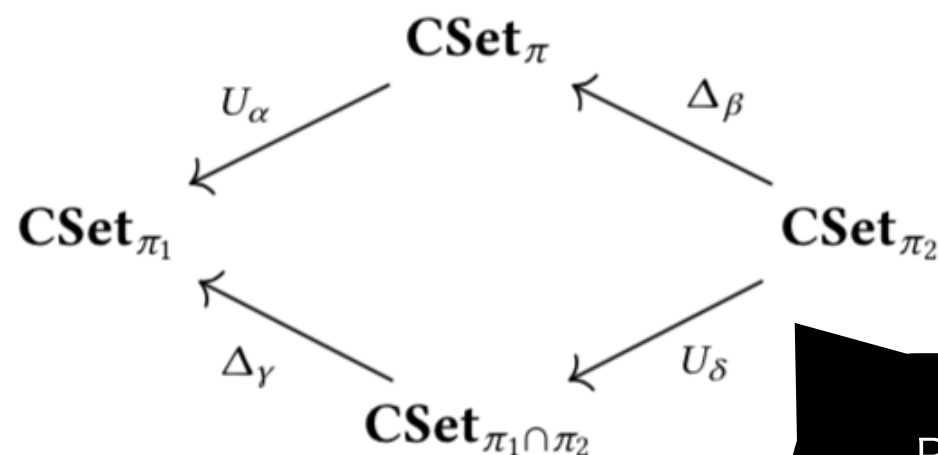
Three fundamental equations

♣ Given

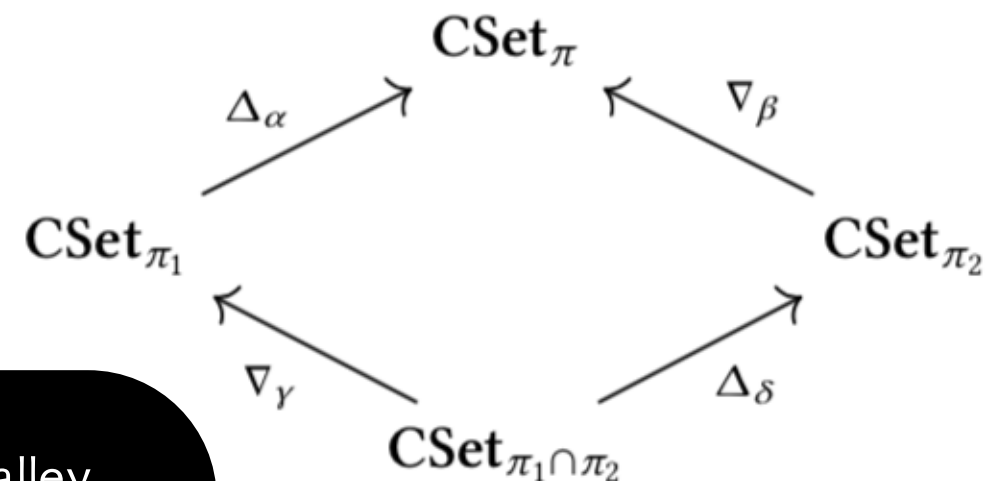


(a pullback) we want:

(1)



(3)



(2) same as (1) but for ∇

Beck-Chevalley
(thanks to D. Spivak)

Observation: These suffice to prove all the laws I have needed so far.

The laws for $\int \dashv \square \dashv \blacklozenge$

PROPOSITION 21.

- (1) *If $\pi \cap \pi' = \emptyset$, then $\square_\pi \square_{\pi'} = \square_{\pi \cup \pi'}$.*
- (2) *If $\pi \cap \pi' = \emptyset$, then $\blacklozenge_\pi \blacklozenge_{\pi'} = \blacklozenge_{\pi \cup \pi'}$.*
- (3) $\square_\pi \square_{\pi'} = \square_{\pi \cup \pi'}$
- (4) $\blacklozenge_\pi \blacklozenge_{\pi'} = \blacklozenge_{\pi \cup \pi'}$
- (5) *If $\pi \subseteq \pi'$, then $\square_{\pi'} \blacklozenge_\pi = \square_{\pi'}$.*
- (6) *If $\pi \subseteq \pi'$, then $\blacklozenge_{\pi'} \square_\pi = \blacklozenge_{\pi'}$.*
- (7) *If $\pi \cap \pi' = \emptyset$, then $\square_\pi \blacklozenge_{\pi'} = \blacklozenge_{\pi'} \square_\pi$.*
- (8) $\square_\pi \blacklozenge_{\pi'} = \blacklozenge_{\pi' - \pi} \square_\pi$.
- (9) $\blacklozenge_\pi \square_{\pi'} = \square_{\pi' - \pi} \blacklozenge_\pi$.

Conclusions

- ❖ One of the most abstract/philosophical parts of categorical algebra, namely **axiomatic cohesion**, is a practical theory of information flow.
- ❖ It can be used to prove properties of LBIFC...
- ❖ ... and, hopefully, it can inspire new languages for LBIFC.
- ❖ Despite the looks of it, **categorical algebra** as a way to reason about programming is still largely unexplored territory.
- ❖ **Multi-modal type theories** have intuitive categorical semantics.