Two isomorphism criteria

A joint work with V. Marra

Luca Spada

ILLC - University of Amsterdam
and
Department of Mathematics - University of Salerno

www.logica.dmi.unisa.it/lucaspada

Stellenbosch, 14th December 2013.

Back&Forth method

The back&forth method is a method for showing isomorphism between countably infinite structures.

Back&Forth method

The back&forth method is a method for showing isomorphism between countably infinite structures.

a_0

Back&Forth method

The back&forth method is a method for showing isomorphism between countably infinite structures.

Back&Forth method

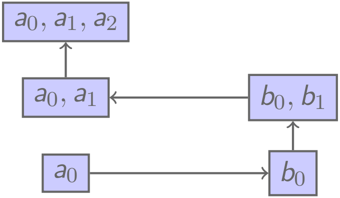
The back&forth method is a method for showing isomorphism between countably infinite structures.

Back&Forth method

The back&forth method is a method for showing isomorphism between countably infinite structures.

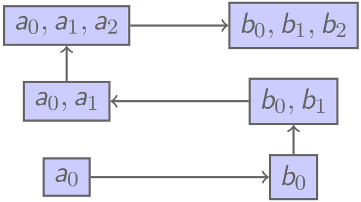
Back&Forth method

The back&forth method is a method for showing isomorphism between countably infinite structures.



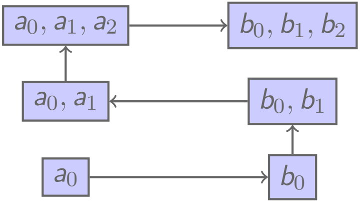
Back&Forth method

The back&forth method is a method for showing isomorphism between countably infinite structures.



Back&Forth method

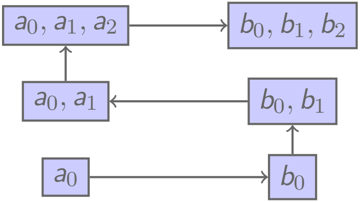
The back&forth method is a method for showing isomorphism between countably infinite structures.



- Any two countably infinite densely ordered sets without endpoints are isomorphic.

Back&Forth method

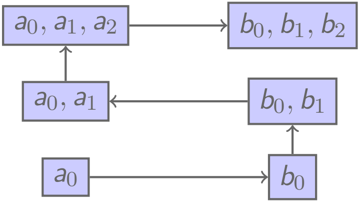
The back&forth method is a method for showing isomorphism between countably infinite structures.



- Any two countably infinite densely ordered sets without endpoints are isomorphic.
- Any two countably infinite atomless Boolean algebras are isomorphic.

Back&Forth method

The back&forth method is a method for showing isomorphism between countably infinite structures.



- Any two countably infinite densely ordered sets without endpoints are isomorphic.
- Any two countably infinite atomless Boolean algebras are isomorphic.
- Any two elementary equivalent countable atomic models of a theory are isomorphic.

Ehrenfeucht-Fraïssé games

The Ehrenfeucht–Fraïssé game is a method for determining whether two structures are elementarily equivalent.

Ehrenfeucht-Fraïssé games

The Ehrenfeucht–Fraïssé game is a method for determining whether two structures are elementarily equivalent.

- In the game there are two structures and two players (Spoiler and Duplicator).

Ehrenfeucht-Fraïssé games

The Ehrenfeucht–Fraïssé game is a method for determining whether two structures are elementarily equivalent.

- In the game there are two structures and two players (Spoiler and Duplicator).
- A round proceeds as follows: Spoiler chooses any element from one of the structures, and Duplicator chooses an element from the other structure.

Ehrenfeucht-Fraïssé games

The Ehrenfeucht–Fraïssé game is a method for determining whether two structures are elementarily equivalent.

- In the game there are two structures and two players (Spoiler and Duplicator).
- A round proceeds as follows: Spoiler chooses any element from one of the structures, and Duplicator chooses an element from the other structure.
- Duplicator wins if, at any round, there exists an isomorphism between the substructures generated by the elements chosen that connects exactly each pair found in each round.

Ehrenfeucht-Fraïssé games

The Ehrenfeucht–Fraïssé game is a method for determining whether two structures are elementarily equivalent.

- In the game there are two structures and two players (Spoiler and Duplicator).
- A round proceeds as follows: Spoiler chooses any element from one of the structures, and Duplicator chooses an element from the other structure.
- Duplicator wins if, at any round, there exists an isomorphism between the substructures generated by the elements chosen that connects exactly each pair found in each round.

Theorem

Suppose the language is finite. Duplicator has a winning strategy iff the two structures are elementarily equivalent

Directed diagrams

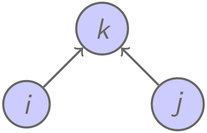
- A set I partially ordered by $\leqslant$ is (upward) directed if for any $i, j \in I$ there exists $k \in I$ with $i, j \leqslant k$.

Directed diagrams

- A set I partially ordered by $\leqslant$ is (upward) directed if for any $i, j \in I$ there exists $k \in I$ with $i, j \leqslant k$.

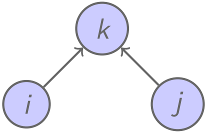
Directed diagrams

- A set I partially ordered by $\leqslant$ is (upward) directed if for any $i, j \in I$ there exists $k \in I$ with $i, j \leqslant k$.



Directed diagrams

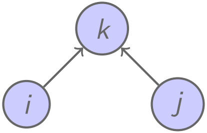
- A set I partially ordered by $\leqslant$ is (upward) directed if for any $i, j \in I$ there exists $k \in I$ with $i, j \leqslant k$.



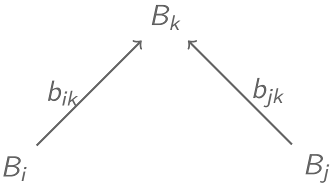
- A directed diagram in C is a pair (B_i, b_{ij}), where
  - I is a directed set, and $i, j \in I$,
  - B_i's are C-object for each $i \in I$,
  - $b_{ij} \colon B_i \to B_j$ is a C-arrow for each $i \leqslant j$.

Directed diagrams

- A set I partially ordered by $\leqslant$ is (upward) directed if for any $i, j \in I$ there exists $k \in I$ with $i, j \leqslant k$.

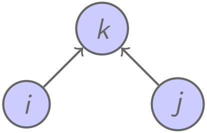


- A directed diagram in C is a pair (B_i, b_{ij}), where
  - I is a directed set, and $i, j \in I$,
  - B_i's are C-object for each $i \in I$,
  - $b_{ij} \colon B_i \to B_j$ is a C-arrow for each $i \leqslant j$.

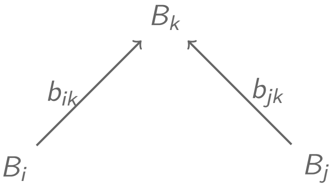


Directed diagrams

- A set I partially ordered by $\leqslant$ is (upward) directed if for any $i, j \in I$ there exists $k \in I$ with $i, j \leqslant k$.



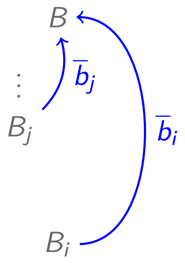
- A directed diagram in C is a pair (B_i, b_{ij}), where
  - I is a directed set, and $i, j \in I$,
  - B_i's are C-object for each $i \in I$,
  - $b_{ij} \colon B_i \to B_j$ is a C-arrow for each $i \leqslant j$.



We call the b_{ij}'s the transition morphisms (or maps).

Cocones

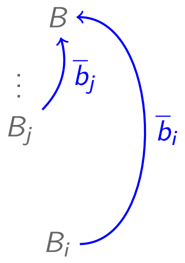
A cocone for (B_i, b_{ij}) is a C-object B equipped with C-arrows $\overline{b}_i \colon B_i \to B$ such that



Cocones

A cocone for (B_i, b_{ij}) is a C-object B equipped with C-arrows $\overline{b}_i \colon B_i \to B$ such that

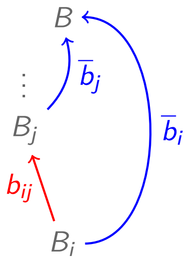
$$\overline{b}_i = \overline{b}_j \circ b_{ij}, \text{ for each } i, j \in I \text{ with } i \leqslant j \ .$$



Cocones

A cocone for (B_i, b_{ij}) is a C-object B equipped with C-arrows $\overline{b}_i \colon B_i \to B$ such that

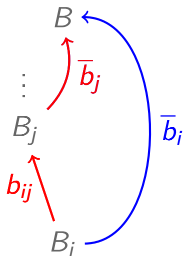
$$\overline{b}_i = \overline{b}_j \circ b_{ij}, \text{ for each } i, j \in I \text{ with } i \leqslant j \, .$$



Cocones

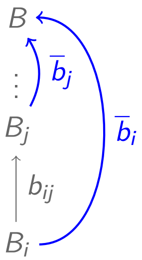
A cocone for (B_i, b_{ij}) is a C-object B equipped with C-arrows $\overline{b}_i \colon B_i \to B$ such that

$$\overline{b}_i = \overline{b}_j \circ b_{ij}, \text{ for each } i, j \in I \text{ with } i \leqslant j \;.$$



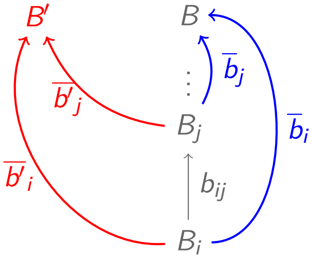
Co-limits

A colimit in C of the diagram (B_i, b_{ij}) is a universal cocone $(B, \overline{b}_i)$,



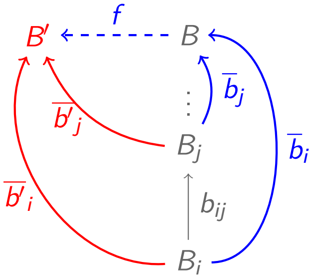
Co-limits

A colimit in C of the diagram (B_i, b_{ij}) is a universal cocone $(B, \overline{b}_i)$, i.e. for any other cocone $(B',' {}_i)$, there is a unique C-arrow f: $B \to B'$ satisfying $\overline{b}'_i = f \circ \overline{b}_i$ for each $i \in I$.



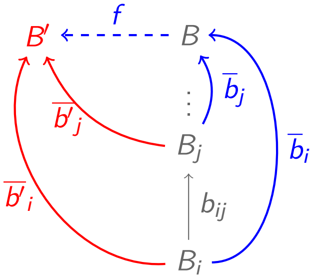
Co-limits

A colimit in C of the diagram (B_i, b_{ij}) is a universal cocone $(B, \overline{b}_i)$, i.e. for any other cocone $(B',' _i)$, there is a unique C-arrow $f\colon B \to B'$ satisfying $\overline{b}'_i = f \circ \overline{b}_i$ for each $i \in I$.



Co-limits

A colimit in C of the diagram (B_i, b_{ij}) is a universal cocone $(B, \overline{b}_i)$, i.e. for any other cocone $(B',' _i)$, there is a unique C-arrow $f\colon B \to B'$ satisfying $\overline{b}'_i = f \circ \overline{b}_i$ for each $i \in I$.



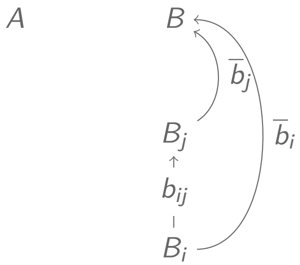
The C-arrows $\overline{b}_i$ are called the colimit morphisms, and B is the colimit object.

Finitely presented object

Definition

An object A in C is finitely presentable if for any arrow f into a colimit C of a diagram $(B, \overline{b}_i)$:

(E)

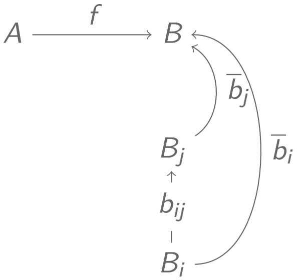


Finitely presented object

Definition

An object A in C is finitely presentable if for any arrow f into a colimit C of a diagram $(B, \overline{b}_i)$:

(E)



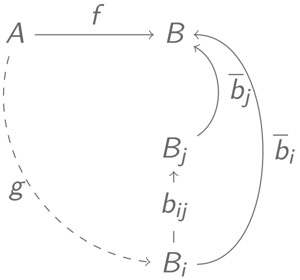
Finitely presented object

Definition

An object A in C is finitely presentable if for any arrow f into a colimit C of a diagram $(B, \overline{b}_i)$:

$$\text{There is } g\colon A \to B_i \text{ such that } f = \overline{b}_i \circ g. \qquad \text{(F)}$$

(E)



Finitely presented object

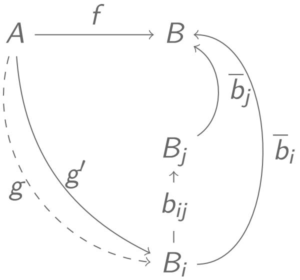
Definition

An object A in C is finitely presentable if for any arrow f into a colimit C of a diagram $(B, \overline{b}_i)$:

There is $g \colon A \to B_i$ such that $f = \overline{b}_i \circ g$. (F)

If $g' \colon A \to B_i$ is such that $f = \overline{b}_i \circ g'$, then

(E)



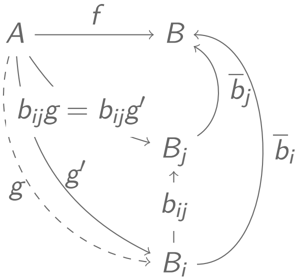
Finitely presented object

Definition

An object A in C is finitely presentable if for any arrow f into a colimit C of a diagram $(B, \overline{b}_i)$:

$$\text{There is } g\colon A \to B_i \text{ such that } f = \overline{b}_i \circ g. \qquad \text{(F)}$$

$$\text{If } g'\colon A \to B_i \text{ is such that } f = \overline{b}_i \circ g', \text{ then there is } j \geqslant i \text{ such that } b_{ij} \circ g = b_{ij} \circ g'. \qquad \text{(E)}$$



Finitely generated objects

Remark

Finitely presented objects are a special case of finitely generated objects.

Finitely generated objects

Remark

Finitely presented objects are a special case of finitely ***generated*** *objects. They can be characterised as the ones that enjoy properties* (E) *and* (F) *above with regard to colimits in which all arrows are* ***mono****.*

Ind-completions

- Ind-completions are essentially a categorical generalisation of ideal completions.

Ind-completions

- Ind-completions are essentially a categorical generalisation of ideal completions.
- An ind-object of a category C is a formal directed colimit of objects of C. The category of ind-objects of C is written ind-C.

Ind-completions

- Ind-completions are essentially a categorical generalisation of ideal completions.
- An ind-object of a category C is a formal directed colimit of objects of C. The category of ind-objects of C is written ind-C.
- Directed diagrams of objects in C are regarded to converge to an object in ind-C, even if that object does not exist in C itself.

Ind-completions

- Ind-completions are essentially a categorical generalisation of ideal completions.
- An ind-object of a category C is a formal directed colimit of objects of C. The category of ind-objects of C is written ind-C.
- Directed diagrams of objects in C are regarded to converge to an object in ind-C, even if that object does not exist in C itself.
- Ind-categories allow one to handle "big things in terms of small things".

Ind-completions

- Ind-completions are essentially a categorical generalisation of ideal completions.
- An ind-object of a category C is a formal directed colimit of objects of C. The category of ind-objects of C is written ind-C.
- Directed diagrams of objects in C are regarded to converge to an object in ind-C, even if that object does not exist in C itself.
- Ind-categories allow one to handle "big things in terms of small things".
- Arrows between ind-objects are defined as

$$\text{ind-C}(F, G) := \lim_{d \in D} \operatorname{colim}_{e \in E} \mathrm{C}(Fd, Ge).$$

Confluence

Definition (Confluent sequences)

Two sequences $(A_i, a_i)_{i\in\mathbb{N}}$ and $(B_k, b_k)_{k\in\mathbb{N}}$ are confluent if there exist integers

$$0 < i_1 < i_2 < \cdots \qquad 0 < k_1 < k_2 < \cdots \ ,$$

Confluence

Definition (Confluent sequences)

Two sequences $(A_i, a_i)_{i\in\mathbb{N}}$ and $(B_k, b_k)_{k\in\mathbb{N}}$ are confluent if there exist integers

$$0 < i_1 < i_2 < \cdots \qquad 0 < k_1 < k_2 < \cdots \ ,$$

and arrows

$$f_n\colon A_{i_n} \to B_{k_n} \text{ and } g_n\colon B_{k_n} \to A_{i_{n+1}}$$

for each $n \in \mathbb{N}$,

Confluence

Definition (Confluent sequences)

Two sequences $(A_i, a_i)_{i\in\mathbb{N}}$ and $(B_k, b_k)_{k\in\mathbb{N}}$ are confluent if there exist integers

$$0 < i_1 < i_2 < \cdots \qquad 0 < k_1 < k_2 < \cdots \ ,$$

and arrows

$$f_n \colon A_{i_n} \to B_{k_n} \text{ and } g_n \colon B_{k_n} \to A_{i_{n+1}}$$

for each $n \in \mathbb{N}$, such that the following commutativity relations hold:

$$a_{i_n i_{n+1}} = g_n \circ f_n \tag{1}$$

$$b_{k_n k_{n+1}} = f_{n+1} \circ g_n \tag{2}$$

Confluence

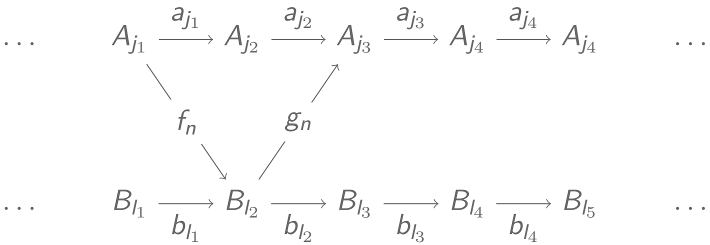
$$\cdots \qquad A_{j_1} \xrightarrow{a_{j_1}} A_{j_2} \xrightarrow{a_{j_2}} A_{j_3} \xrightarrow{a_{j_3}} A_{j_4} \xrightarrow{a_{j_4}} A_{j_4} \qquad \cdots$$

$$\cdots \qquad B_{l_1} \xrightarrow[b_{l_1}]{} B_{l_2} \xrightarrow[b_{l_2}]{} B_{l_3} \xrightarrow[b_{l_3}]{} B_{l_4} \xrightarrow[b_{l_4}]{} B_{l_5} \qquad \cdots$$

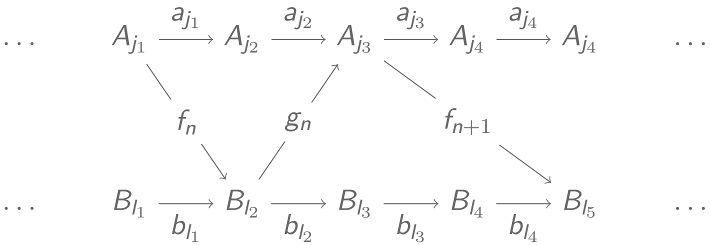
Confluence

$$
\begin{array}{ccccccccccccc}
\cdots & & A_{j_1} & \xrightarrow{a_{j_1}} & A_{j_2} & \xrightarrow{a_{j_2}} & A_{j_3} & \xrightarrow{a_{j_3}} & A_{j_4} & \xrightarrow{a_{j_4}} & A_{j_4} & & \cdots \\
 & & & \searrow^{f_n} & & & & & & & & & \\
\cdots & & B_{l_1} & \xrightarrow[b_{l_1}]{} & B_{l_2} & \xrightarrow[b_{l_2}]{} & B_{l_3} & \xrightarrow[b_{l_3}]{} & B_{l_4} & \xrightarrow[b_{l_4}]{} & B_{l_5} & & \cdots
\end{array}
$$

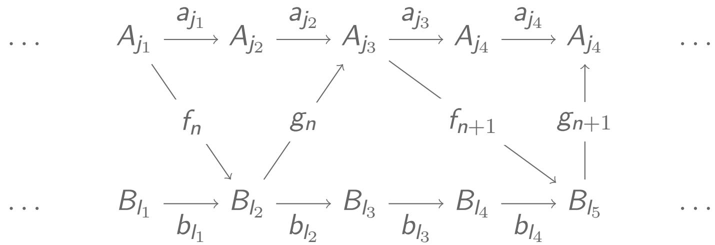
Confluence


$\cdots \quad A_{j_1} \xrightarrow{a_{j_1}} A_{j_2} \xrightarrow{a_{j_2}} A_{j_3} \xrightarrow{a_{j_3}} A_{j_4} \xrightarrow{a_{j_4}} A_{j_4} \quad \cdots$
f_n
g_n
$\cdots \quad B_{l_1} \xrightarrow{b_{l_1}} B_{l_2} \xrightarrow{b_{l_2}} B_{l_3} \xrightarrow{b_{l_3}} B_{l_4} \xrightarrow{b_{l_4}} B_{l_5} \quad \cdots$

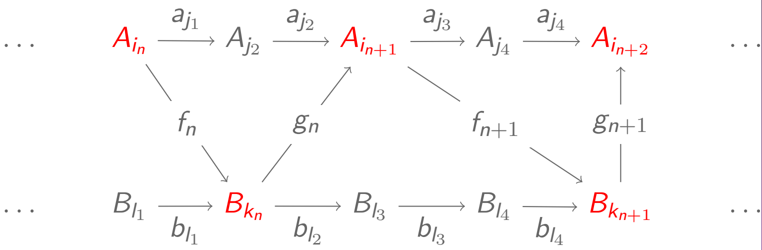
Confluence


$\cdots \quad A_{j_1} \xrightarrow{a_{j_1}} A_{j_2} \xrightarrow{a_{j_2}} A_{j_3} \xrightarrow{a_{j_3}} A_{j_4} \xrightarrow{a_{j_4}} A_{j_4} \quad \cdots$
f_n
g_n
f_{n+1}
$\cdots \quad B_{l_1} \xrightarrow[b_{l_1}]{} B_{l_2} \xrightarrow[b_{l_2}]{} B_{l_3} \xrightarrow[b_{l_3}]{} B_{l_4} \xrightarrow[b_{l_4}]{} B_{l_5} \quad \cdots$

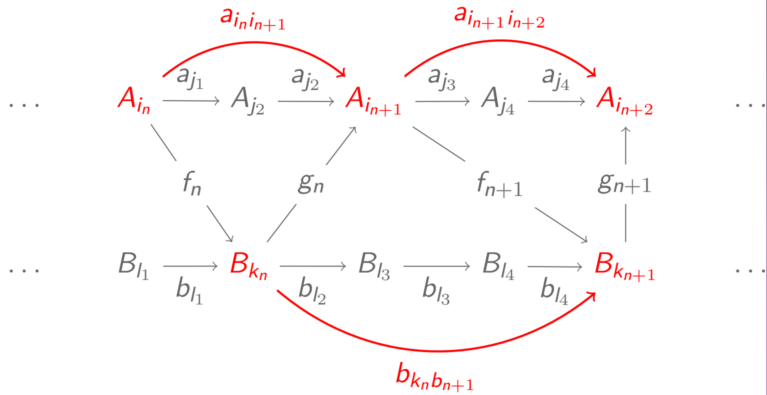
Confluence


. . .
A_{j_1}
a_{j_1}
A_{j_2}
a_{j_2}
A_{j_3}
a_{j_3}
A_{j_4}
a_{j_4}
A_{j_4}
. . .
f_n
g_n
f_{n+1}
g_{n+1}
. . .
B_{l_1}
b_{l_1}
B_{l_2}
b_{l_2}
B_{l_3}
b_{l_3}
B_{l_4}
b_{l_4}
B_{l_5}
. . .

Confluence

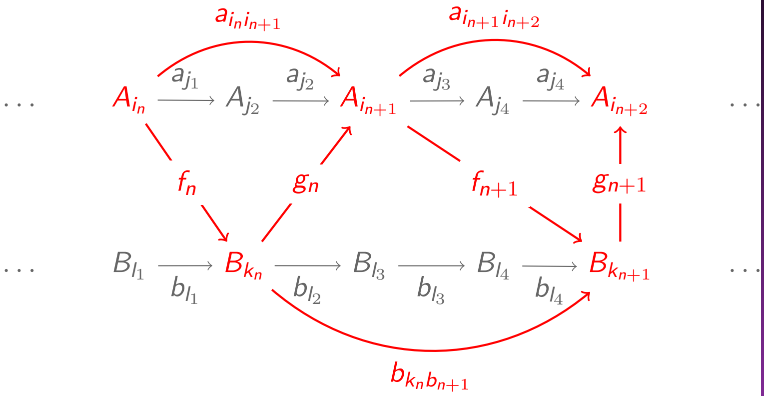

$\dots$ A_{i_n} $\xrightarrow{a_{j_1}}$ A_{j_2} $\xrightarrow{a_{j_2}}$ $A_{i_{n+1}}$ $\xrightarrow{a_{j_3}}$ A_{j_4} $\xrightarrow{a_{j_4}}$ $A_{i_{n+2}}$ $\dots$
f_n g_n f_{n+1} g_{n+1}
$\dots$ B_{l_1} $\xrightarrow{b_{l_1}}$ B_{k_n} $\xrightarrow{b_{l_2}}$ B_{l_3} $\xrightarrow{b_{l_3}}$ B_{l_4} $\xrightarrow{b_{l_4}}$ $B_{k_{n+1}}$ $\dots$

Confluence



$a_{i_n i_{n+1}}$
$a_{i_{n+1} i_{n+2}}$
$\dots$ A_{i_n} a_{j_1} A_{j_2} a_{j_2} $A_{i_{n+1}}$ a_{j_3} A_{j_4} a_{j_4} $A_{i_{n+2}}$ $\dots$
f_n g_n f_{n+1} g_{n+1}
$\dots$ B_{l_1} b_{l_1} B_{k_n} b_{l_2} B_{l_3} b_{l_3} B_{l_4} b_{l_4} $B_{k_{n+1}}$ $\dots$
$b_{k_n b_{n+1}}$

Confluence



$a_{i_n i_{n+1}}$
$a_{i_{n+1} i_{n+2}}$
a_{j_1}
a_{j_2}
a_{j_3}
a_{j_4}
$\dots$
A_{i_n}
A_{j_2}
$A_{i_{n+1}}$
A_{j_4}
$A_{i_{n+2}}$
$\dots$
f_n
g_n
f_{n+1}
g_{n+1}
$\dots$
B_{l_1}
B_{k_n}
B_{l_3}
B_{l_4}
$B_{k_{n+1}}$
$\dots$
b_{l_1}
b_{l_2}
b_{l_3}
b_{l_4}
$b_{k_n b_{n+1}}$

The main theorem

Theorem

Let C *be any locally small category. Suppose two sequences* $(A_i, a_i)_{i\in\mathbb{N}}$ *and* $(B_k, b_k)_{k\in\mathbb{N}}$ *in* C *admit colimit objects* A *and* B *in* C*, respectively.*

The main theorem

Theorem

Let C be any locally small category. Suppose two sequences $(A_i, a_i)_{i\in\mathbb{N}}$ and $(B_k, b_k)_{k\in\mathbb{N}}$ in C admit colimit objects A and B in C, respectively.

If one of the following applies

The main theorem

Theorem

Let C be any locally small category. Suppose two sequences $(A_i, a_i)_{i\in\mathbb{N}}$ and $(B_k, b_k)_{k\in\mathbb{N}}$ in C admit colimit objects A and B in C, respectively.

If one of the following applies

1. *All A_i and B_k are finitely presentable.*

The main theorem

Theorem

Let C be any locally small category. Suppose two sequences $(A_i, a_i)_{i \in \mathbb{N}}$ and $(B_k, b_k)_{k \in \mathbb{N}}$ in C admit colimit objects A and B in C, respectively.

If one of the following applies

1. *All A_i and B_k are finitely presentable.*
2. *All A_i and B_k are finitely generated, and all a_i and b_k are monomorphisms.*

The main theorem

Theorem

Let C be any locally small category. Suppose two sequences $(A_i, a_i)_{i \in \mathbb{N}}$ and $(B_k, b_k)_{k \in \mathbb{N}}$ in C admit colimit objects A and B in C, respectively.

If one of the following applies

1. *All A_i and B_k are finitely presentable.*
2. *All A_i and B_k are finitely generated, and all a_i and b_k are monomorphisms.*

Then A and B are isomorphic if, and only if, the two sequences are confluent.

A non example

In Set, write $\mathbb{N}$ as a colimit of the sequence

$$\emptyset \hookrightarrow \{1\} \hookrightarrow \{1, 2, \ldots\} \hookrightarrow \cdots \ \mathbb{N}.$$

A non example

In Set, write $\mathbb{N}$ as a colimit of the sequence

$$\emptyset \hookrightarrow \{1\} \hookrightarrow \{1, 2, \dots\} \hookrightarrow \cdots \quad \mathbb{N}.$$

Further write $\mathbb{N}$ as the colimit of the constant sequence

$$\mathbb{N} \to \mathbb{N} \to \cdots \quad \mathbb{N},$$

where each arrow is the identicaltity function.

A non example

In Set, write $\mathbb{N}$ as a colimit of the sequence

$$\emptyset \hookrightarrow \{1\} \hookrightarrow \{1,2,\ldots\} \hookrightarrow \cdots \;\; \mathbb{N}.$$

Further write $\mathbb{N}$ as the colimit of the constant sequence

$$\mathbb{N} \to \mathbb{N} \to \cdots \;\; \mathbb{N},$$

where each arrow is the identicaltity function.
The two sequences are not confluent.

A non example

In Set, write $\mathbb{N}$ as a colimit of the sequence

$$\emptyset \hookrightarrow \{1\} \hookrightarrow \{1, 2, \ldots\} \hookrightarrow \cdots \ \mathbb{N}.$$

Further write $\mathbb{N}$ as the colimit of the constant sequence

$$\mathbb{N} \to \mathbb{N} \to \cdots \ \mathbb{N},$$

where each arrow is the identicaltity function.
The two sequences are not confluent. The second sequence does not consist of finitely presentable or finitely generated objects.

Proof

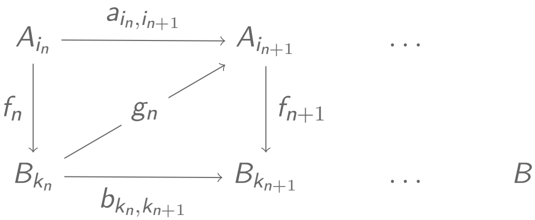
The right-to-left direction (confluence $\Rightarrow$ iso) holds in general i.e. no need for finitely presented objects.

Proof

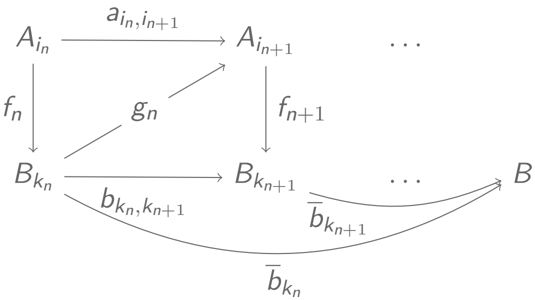
The right-to-left direction (confluence $\Rightarrow$ iso) holds in general i.e. no need for finitely presented objects.

The left-to-right direction (iso $\Rightarrow$ confluence) crucially uses the finite nature of the objects (as seen in the previous example).

Proof of "confluence $\Rightarrow$ iso"



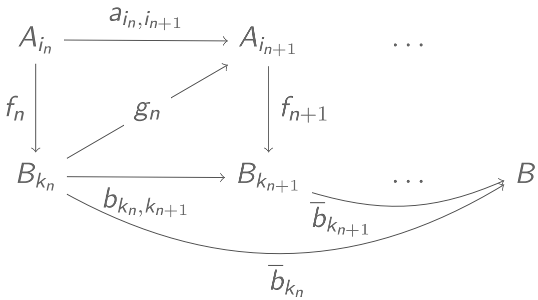
Proof of "confluence ⇒ iso"



Proof of "confluence ⇒ iso"

Claim

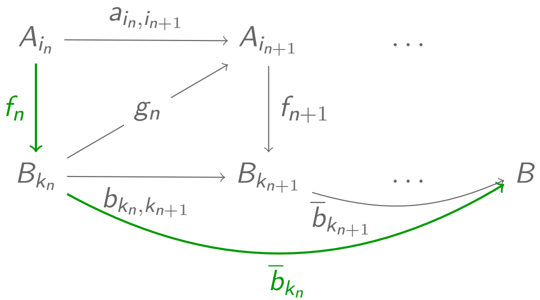
B is a cocone for $(A_i, a_i)_{i\in\mathbb{N}}$



Proof of "confluence ⇒ iso"

Claim

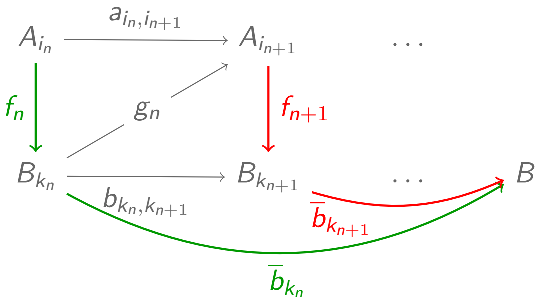
B is a cocone for $(A_i, a_i)_{i\in\mathbb{N}}$



Proof of "confluence ⇒ iso"

Claim

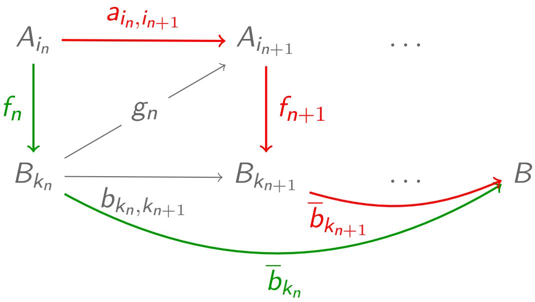
B is a cocone for $(A_i, a_i)_{i\in\mathbb{N}}$



Proof of "confluence ⇒ iso"

Claim

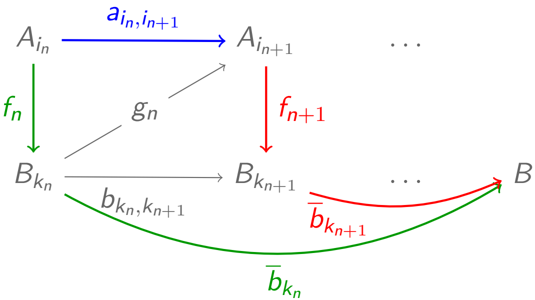
B is a cocone for $(A_i, a_i)_{i \in \mathbb{N}}$



Proof of "confluence ⇒ iso"

Claim

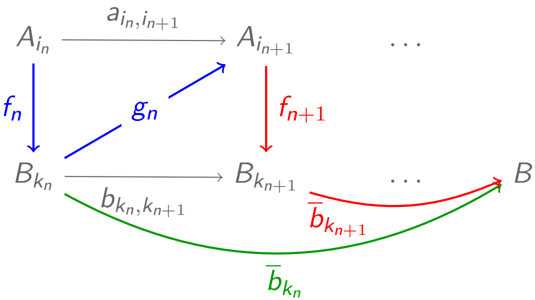
B is a cocone for $(A_i, a_i)_{i \in \mathbb{N}}$



Proof of "confluence ⇒ iso"

Claim

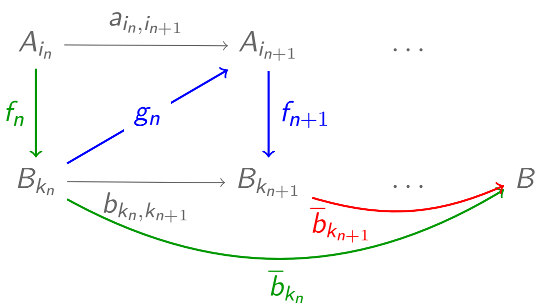
B is a cocone for $(A_i, a_i)_{i \in \mathbb{N}}$



Proof of "confluence ⇒ iso"

Claim

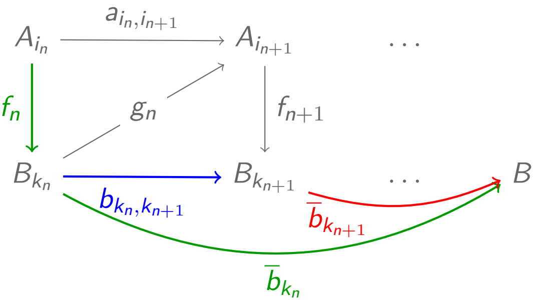
B is a cocone for $(A_i, a_i)_{i\in\mathbb{N}}$



Proof of "confluence ⇒ iso"

Claim

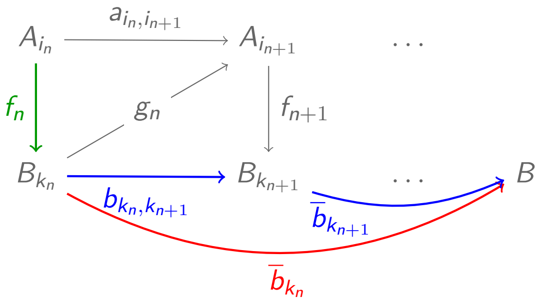
B is a cocone for $(A_i, a_i)_{i\in\mathbb{N}}$



Proof of "confluence ⇒ iso"

Claim

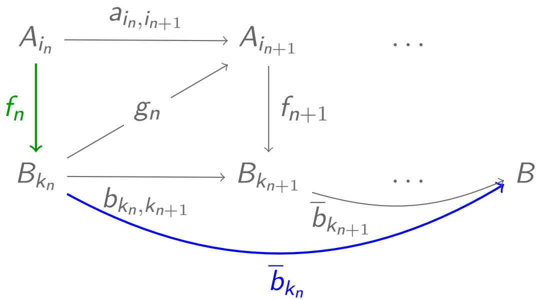
B is a cocone for $(A_i, a_i)_{i \in \mathbb{N}}$



Proof of "confluence ⇒ iso"

Claim

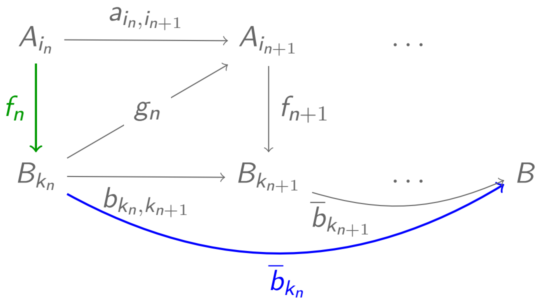
B is a cocone for $(A_i, a_i)_{i\in\mathbb{N}}$



Proof of "confluence ⇒ iso"

Claim

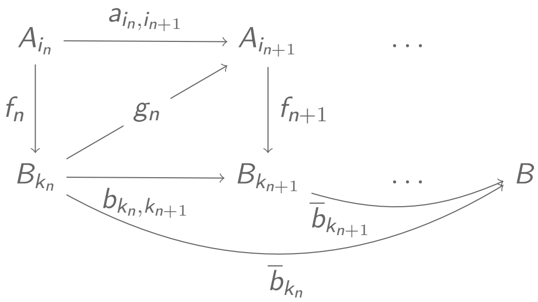
B is a cocone for $(A_i, a_i)_{i\in\mathbb{N}}$ ✓



Proof of "confluence ⇒ iso"

Claim

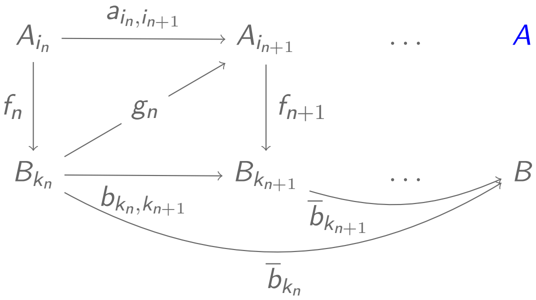
There is an arrow from A to B



Proof of "confluence ⇒ iso"

Claim

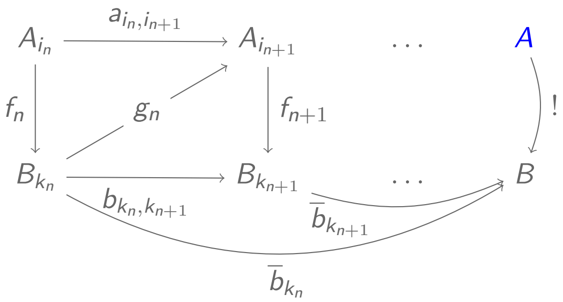
There is an arrow from A to B



Proof of "confluence ⇒ iso"

Claim

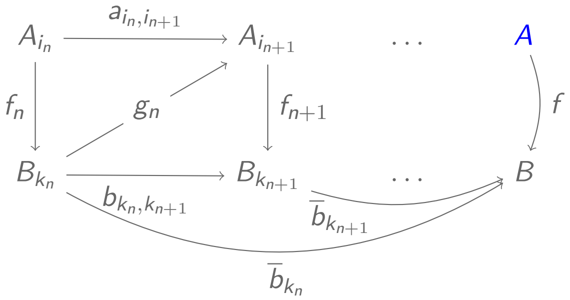
There is an arrow from A to B



Proof of "confluence ⇒ iso"

Claim

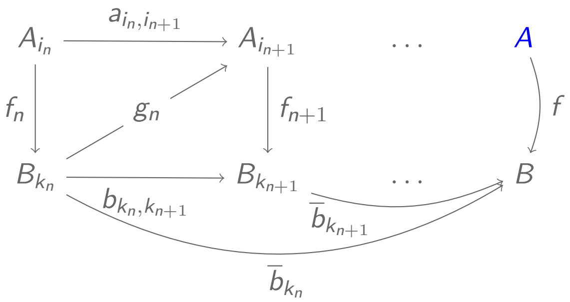
There is an arrow from A to B



Proof of "confluence ⇒ iso"

Claim

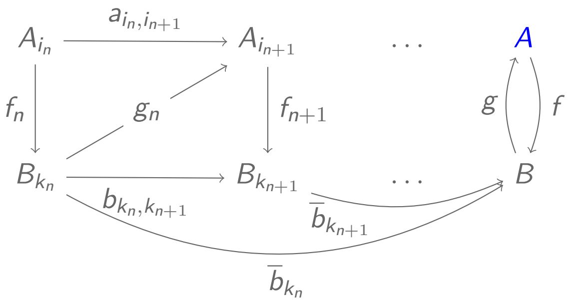
There is an arrow from A to B ✓



Proof of "confluence ⇒ iso"

Claim

There is an arrow from A to B ✓

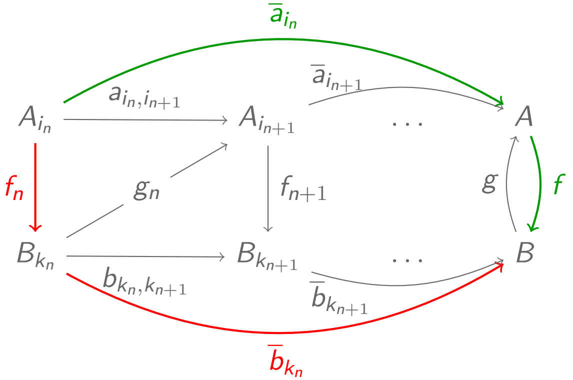


Proof of "confluence ⇒ iso"

Remark

Notice that

1. All cocone arrows factor through f i.e., $\overline{b}_{k_n} \circ f_n = f \circ \overline{a}_{i_n}$

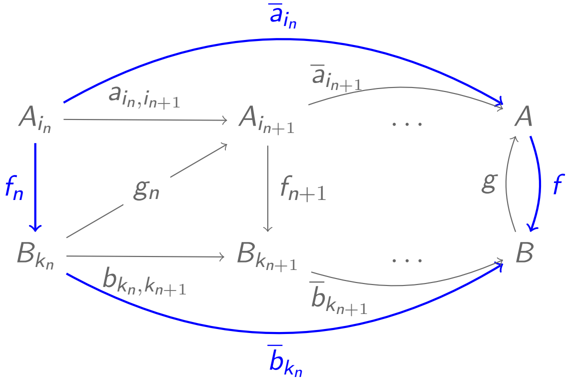


Proof of "confluence ⇒ iso"

Remark

Notice that

1. All cocone arrows factor through f i.e., $\overline{b}_{k_n} \circ f_n = f \circ \overline{a}_{i_n}$

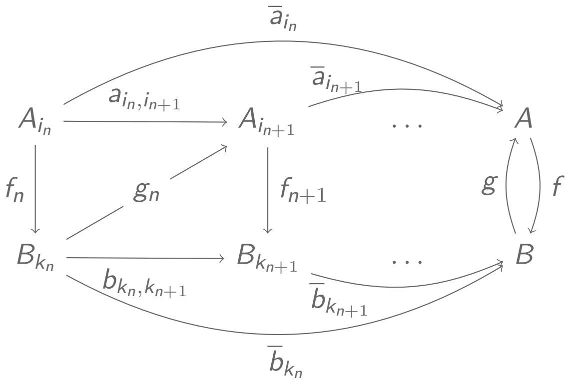


Proof of "confluence ⇒ iso"

Remark

Notice that

1. All cocone arrows factor through f i.e., $\overline{b}_{k_n} \circ f_n = f \circ \overline{a}_{i_n}$
2. All cocone arrows factor through g i.e., $\overline{a}_{i_n} \circ g_n = g \circ \overline{b}_{k_n}$

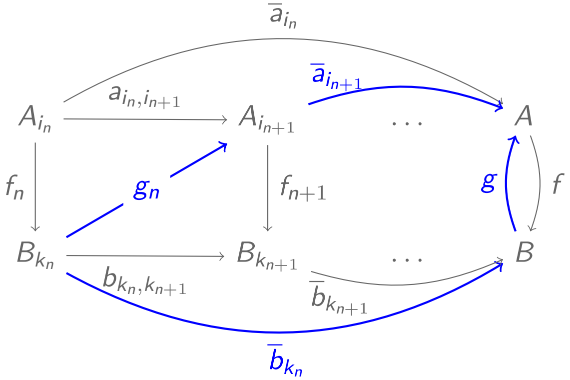


Proof of "confluence $\Rightarrow$ iso"

Remark

Notice that

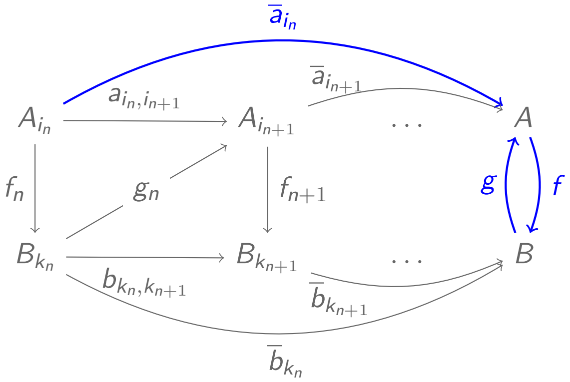
1. All cocone arrows factor through f i.e., $\overline{b}_{k_n} \circ f_n = f \circ \overline{a}_{i_n}$
2. All cocone arrows factor through g i.e., $\overline{a}_{i_n} \circ g_n = g \circ \overline{b}_{k_n}$



Proof of "confluence ⇒ iso"

Claim C

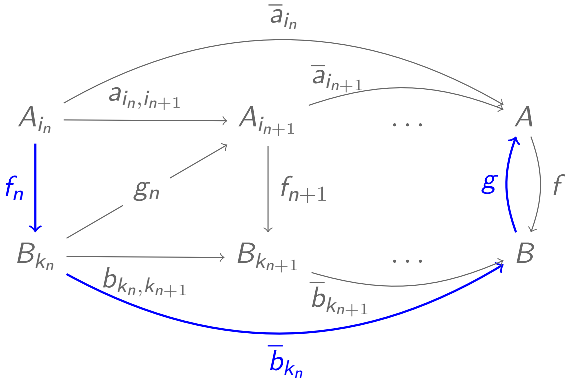
The composition $g \circ f$ is the identity



Proof of "confluence ⇒ iso"

Claim C

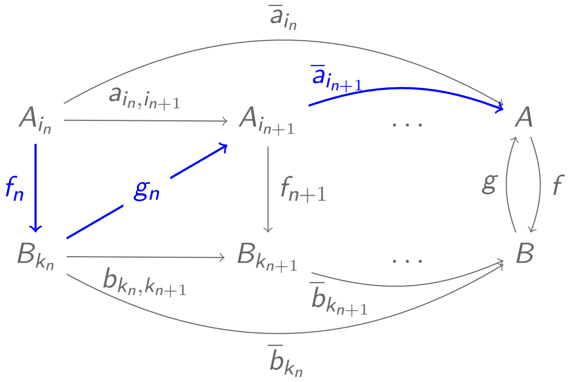
The composition $g \circ f$ is the identity



Proof of "confluence ⇒ iso"

Claim C

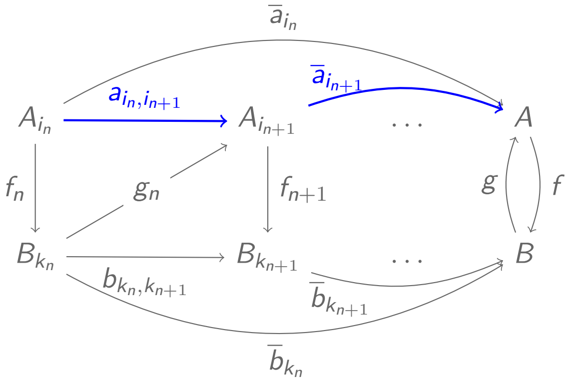
The composition $g \circ f$ is the identity



Proof of "confluence ⇒ iso"

Claim C

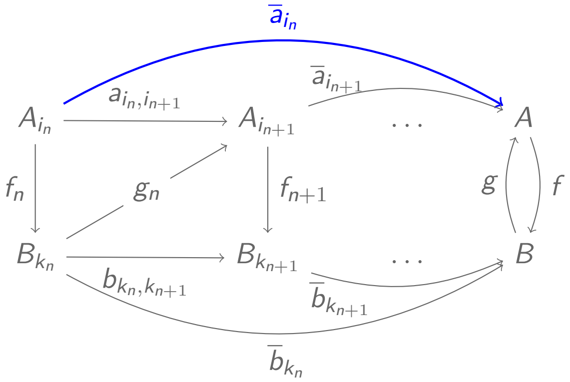
The composition $g \circ f$ is the identity



Proof of "confluence ⇒ iso"

Claim C

The composition $g \circ f$ is the identity



Proof of "confluence ⇒ iso"

Claim C

The composition $g \circ f$ is the identity

1. So $g \circ f$ factors all $\overline{a}_{i_n}$, i.e. $\overline{a}_{i_n} \circ (g \circ f) = \overline{a}_{i_n}$.

Proof of "confluence ⇒ iso"

Claim C

The composition $g \circ f$ is the identity

1. So $g \circ f$ factors all $\bar{a}_{i_n}$, i.e. $\bar{a}_{i_n} \circ (g \circ f) = \bar{a}_{i_n}$.
2. But also id_A does, for $\bar{a}_{i_n} \circ \mathrm{id}_A = \bar{a}_{i_n}$

Proof of "confluence ⇒ iso"

Claim C

The composition $g \circ f$ is the identity

1. So $g \circ f$ factors all $\bar{a}_{i_n}$, i.e. $\bar{a}_{i_n} \circ (g \circ f) = \bar{a}_{i_n}$.
2. But also id_A does, for $\bar{a}_{i_n} \circ \mathrm{id}_A = \bar{a}_{i_n}$
3. But, by the universal property, there can be only one arrow with this property, so $g \circ f = \mathrm{id}_A$.

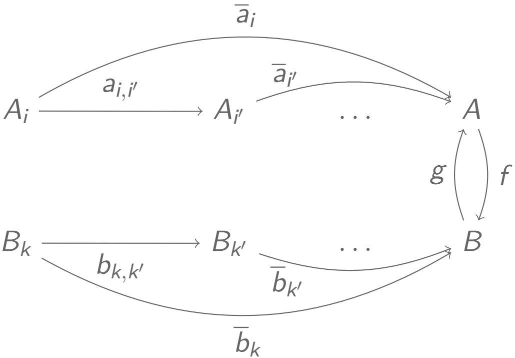
Proof of "confluence ⇒ iso"

Claim C

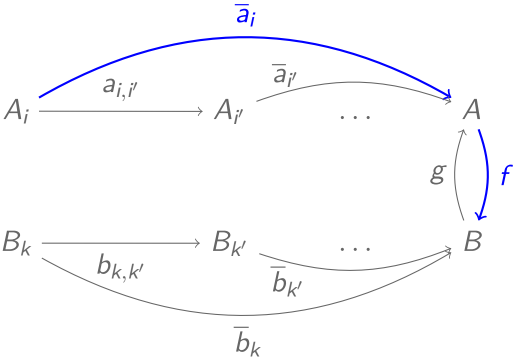
The composition $g \circ f$ is the identity ✓

1. So $g \circ f$ factors all $\bar{a}_{i_n}$, i.e. $\bar{a}_{i_n} \circ (g \circ f) = \bar{a}_{i_n}$.
2. But also id_A does, for $\bar{a}_{i_n} \circ \mathrm{id}_A = \bar{a}_{i_n}$
3. But, by the universal property, there can be only one arrow with this property, so $g \circ f = \mathrm{id}_A$.

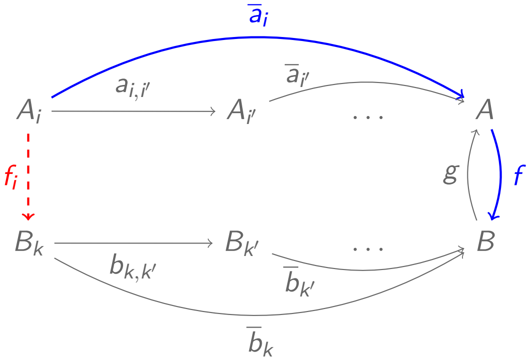
Proof of "iso ⇒ confluence"



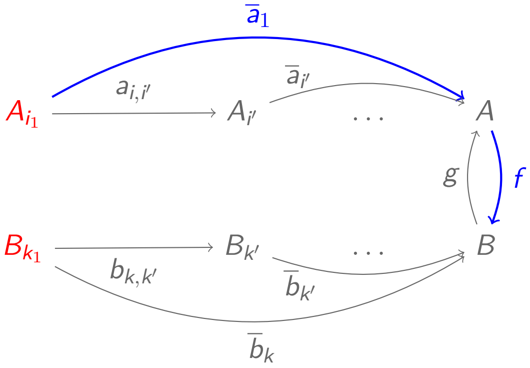
Proof of "iso ⇒ confluence"



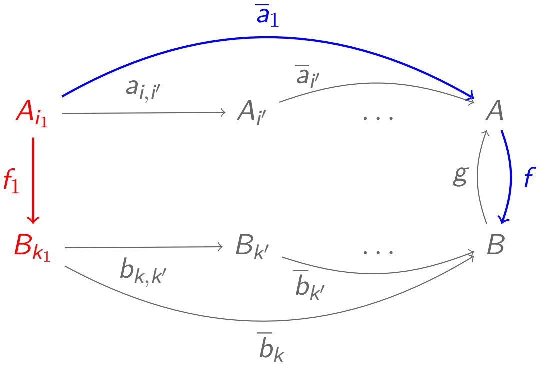
Proof of "iso ⇒ confluence"



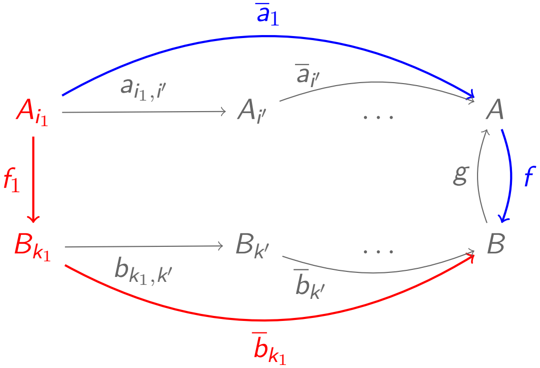
Proof of "iso ⇒ confluence"



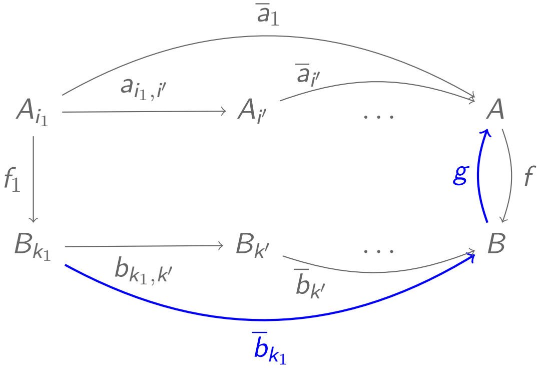
Proof of "iso ⇒ confluence"



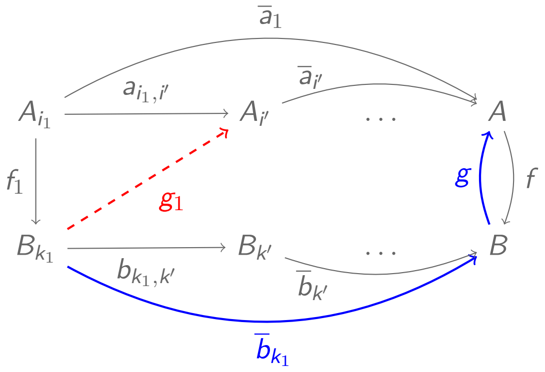
Proof of "iso ⇒ confluence"



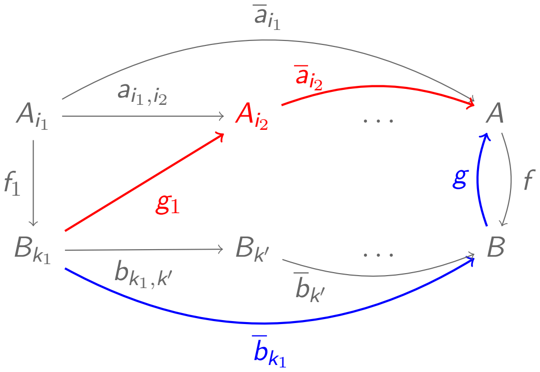
Proof of "iso ⇒ confluence"



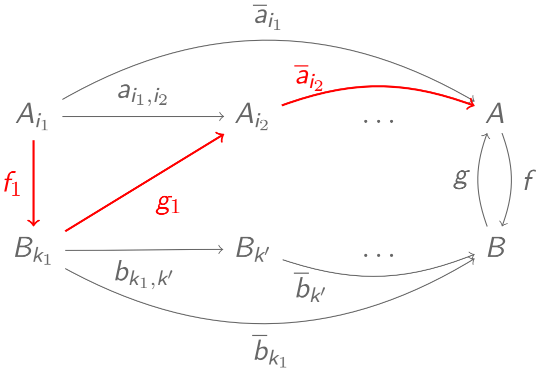
Proof of "iso ⇒ confluence"



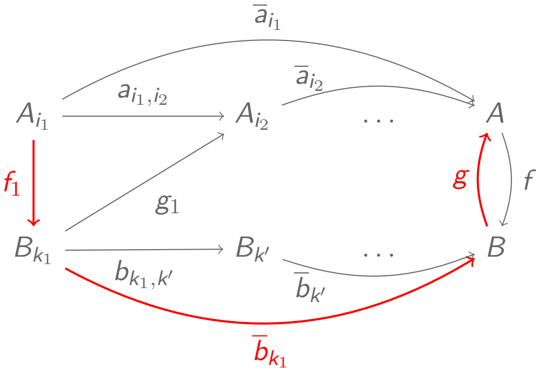
Proof of "iso ⇒ confluence"



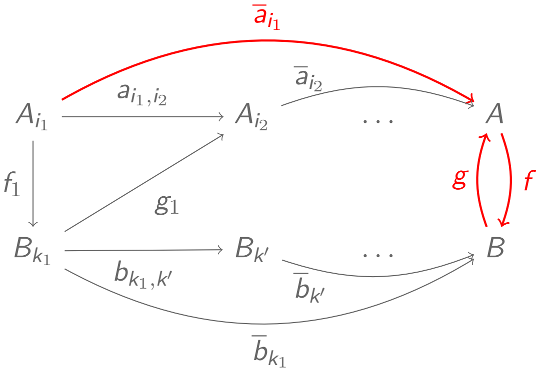
Proof of "iso ⇒ confluence"



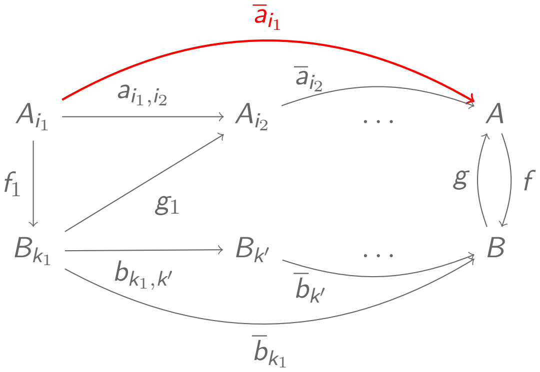
Proof of "iso ⇒ confluence"



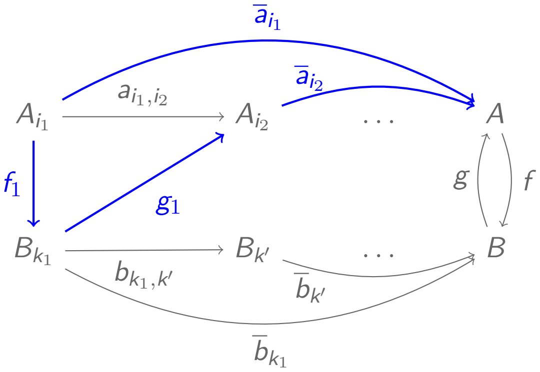
Proof of "iso ⇒ confluence"



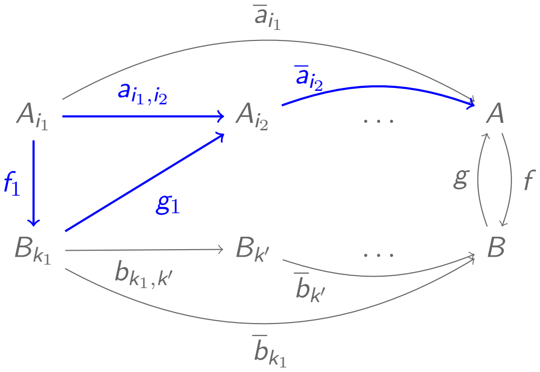
Proof of "iso ⇒ confluence"



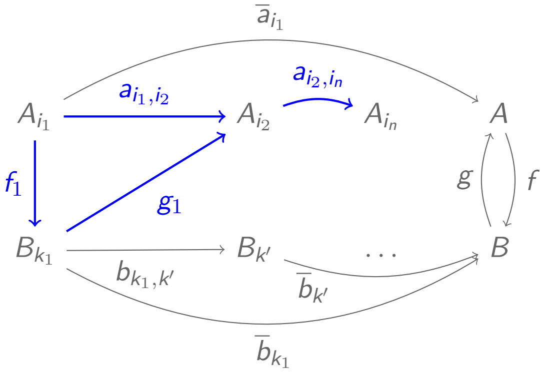
Proof of "iso ⇒ confluence"



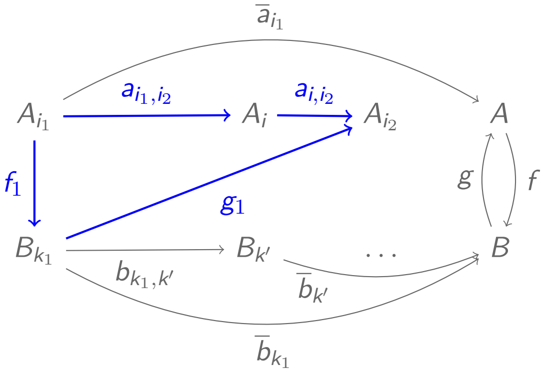
Proof of "iso ⇒ confluence"



Proof of "iso ⇒ confluence"



Proof of "iso ⇒ confluence"



$u\ell$-groups

The main result of

- M.Busaniche, L. Cabrer, and D. Mundici. Confluence and combinatorics in finitely generated unital lattice-ordered abelian groups, *Forum Mathematicum*, 24 (2012), 253-271.

is as follows:

Theorem 3.3

Given direct systems S and T of finitely presented unital ℓ-groups with surjective connecting unital ℓ-homomorphisms let (G, u) and (H, v) denote their respective direct limits. Then the following conditions are equivalent:

1. $(G, u) \cong (H, v)$
2. S and T are confluent.

MV-algebras

Aslo in

- D. Mundici, Advanced Łukasiewicz calculus and MV-algebras, Trends in Logic, Vol. 35 Springer, New York, (2011).

Corollary 8.8

Let $D = A_0 \twoheadrightarrow A_1 \twoheadrightarrow A_2 ...$ and $E = B_0 \twoheadrightarrow B_1 \twoheadrightarrow B_2 ...$, be direct systems of finitely presented MV-algebras with surjective homomorphisms. Let A and B denote their respective direct limits. Then the following conditions are equivalent:

1. $A \cong B$.
2. D and E are confluent.

Torsion-free Abelian groups

Derry, Kurosh, and Mal'cev's isomorphisms invariants for countable torsion-free Abelian groups.

Torsion-free Abelian groups

Derry, Kurosh, and Mal'cev's isomorphisms invariants for countable torsion-free Abelian groups.

- Represent each countable torsion-free Abelian group as the directed colimit of some sequence

$$\mathbb{Z}^{r_1} \xrightarrow{\iota_1} \mathbb{Z}^{r_2} \xrightarrow{\iota_2} \dots \mathbb{Z}^{r_i} \xrightarrow{\iota_i} \dots$$

Torsion-free Abelian groups

Derry, Kurosh, and Mal'cev's isomorphisms invariants for countable torsion-free Abelian groups.

- Represent each countable torsion-free Abelian group as the directed colimit of some sequence

$$\mathbb{Z}^{r_1} \xrightarrow{\iota_1} \mathbb{Z}^{r_2} \xrightarrow{\iota_2} \dots \mathbb{Z}^{r_i} \xrightarrow{\iota_i} \dots$$

- Choose a $\mathbb{Z}$-module bases and represent each group homomorphism in the confluence as a matrix with integer entries

Torsion-free Abelian groups

Derry, Kurosh, and Mal'cev's isomorphisms invariants for countable torsion-free Abelian groups.

- Represent each countable torsion-free Abelian group as the directed colimit of some sequence

$$\mathbb{Z}^{r_1} \xrightarrow{\iota_1} \mathbb{Z}^{r_2} \xrightarrow{\iota_2} \dots \mathbb{Z}^{r_i} \xrightarrow{\iota_i} \dots$$

- Choose a $\mathbb{Z}$-module bases and represent each group homomorphism in the confluence as a matrix with integer entries
- Write down the commutativity conditions in the confluence in terms of products of such matrices,

Torsion-free Abelian groups

Derry, Kurosh, and Mal'cev's isomorphisms invariants for countable torsion-free Abelian groups.

- Represent each countable torsion-free Abelian group as the directed colimit of some sequence

$$\mathbb{Z}^{r_1} \xrightarrow{\iota_1} \mathbb{Z}^{r_2} \xrightarrow{\iota_2} \dots \mathbb{Z}^{r_i} \xrightarrow{\iota_i} \dots$$

- Choose a $\mathbb{Z}$-module bases and represent each group homomorphism in the confluence as a matrix with integer entries
- Write down the commutativity conditions in the confluence in terms of products of such matrices,
- Obtain an equivalence relation on sequences of matrices.

Dimension groups

The Bratteli-Elliott Isomorphism Criterion

Two sequences of simplicial groups and order-preserving group homomorphisms have isomorphic colimit dimension groups if, and only if, they are confluent.