The lattice of database states

David Barrett-Lennard
16 Oct 2016

Introduction

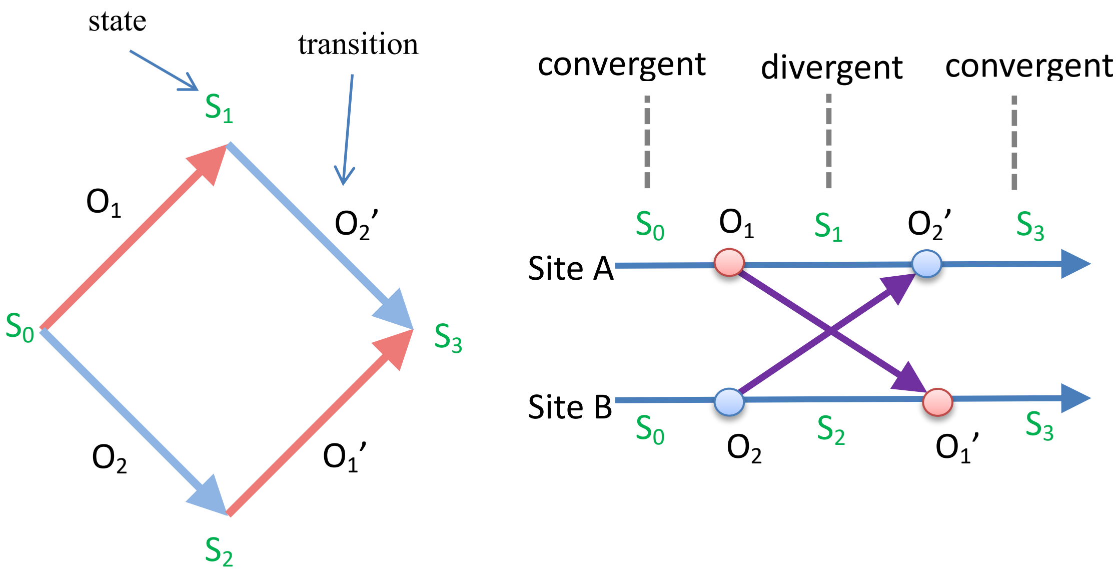
This document describes some of the mathematical aspects of the CEDA replication technology. It is only provided for information purposes for those interested in knowing more about how it works, it does not need to be understood by software developers using CEDA to build applications.

For convenience some mathematical definitions are provided in an appendix.

State transition diagram

Given a scenario of some sites exchanging operations there is an associated ***state transition diagram***. In the case of two sites exchanging two concurrent operations the state transition diagram is a diamond.

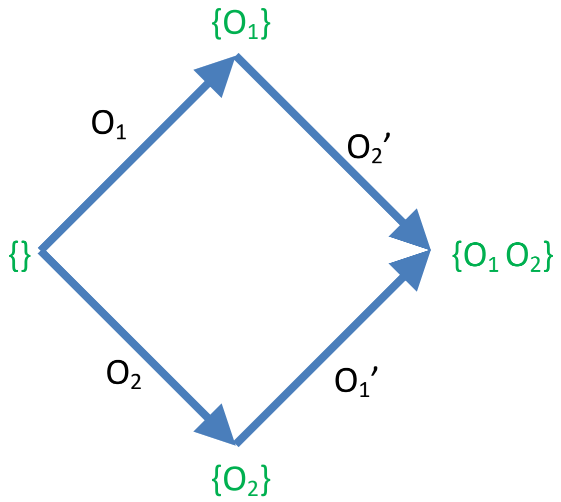
In the following diagram there are two sites A and B that are initially in the same state S_0. Site A generates and applies operation O_1 and transitions to state S_1. Site B concurrently generates and applies operation O_2 and transitions to state S_2. Each site sends its operation to the other site. The received operation is transformed and applied and both sites enter state S_3.



5

Correspondence between states and sets of operations

Each *state* can be labelled with the *set of operations* producing that state, so there is a one to one correspondence between states and sets of operations in the state transition diagram.

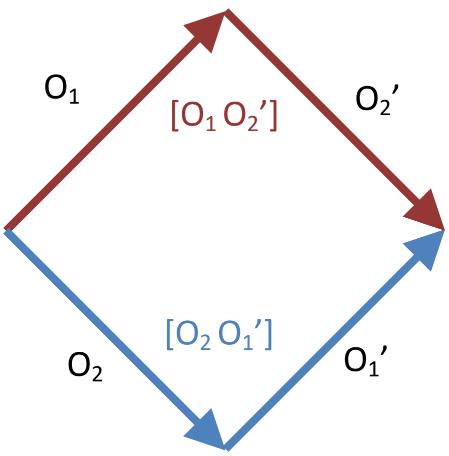


With n concurrent operations
there are 2^n states

$2^0 = 1$
$2^1 = 2$
$2^2 = 4$
$2^3 = 8$
$2^4 = 16$
$2^5 = 32$

Correspondence between paths and lists of operations

A *list of operations* corresponds to a *path* through the state transaction diagram. In the diamond there are two paths (along the top and along the bottom), corresponding to the two permutations of a list with two elements.

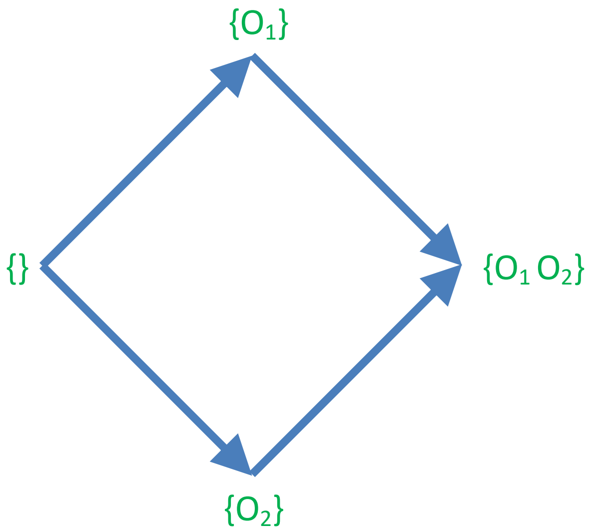


With n concurrent operations
there are n! paths

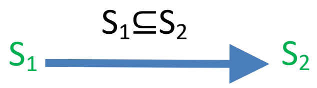
$0! = 1$
$1! = 1$
$2! = 2$
$3! = 6$
$4! = 24$
$5! = 120$

State transition diagram is the Hasse diagram of subset

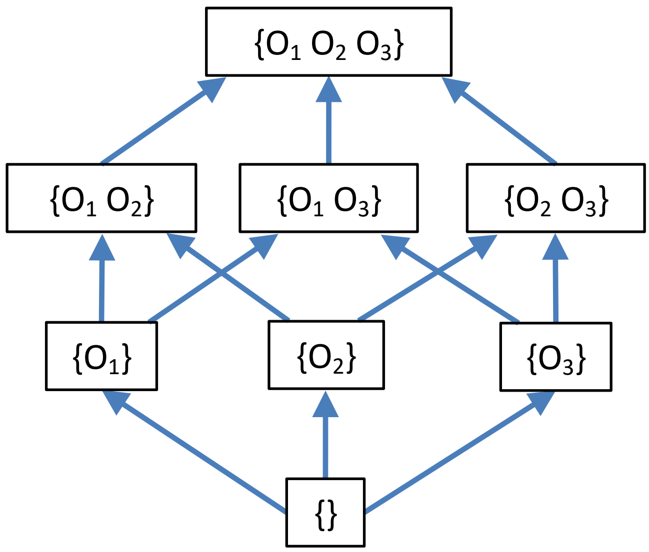
It has already been pointed out that there is a one to one correspondence between states and sets of operations. The subset operation on the sets of operations gives a partial order on the states. The Hasse diagram (see the appendix for a definition) of the subset relation corresponds to what we have previously described as the state transition diagram.



$\subseteq$ is a partial order on the states



When there are three concurrent operations we get a cube.



Poset $(P,\subseteq)$ is a bounded lattice

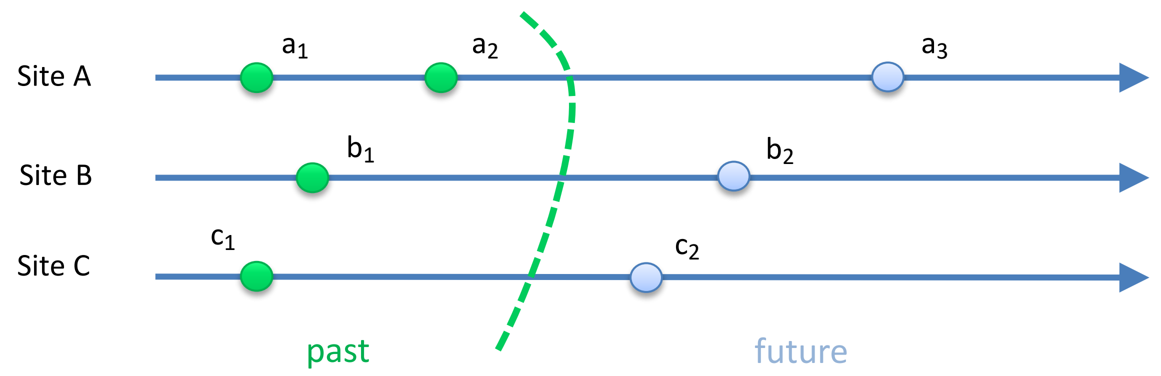
$|P| = 2^3 = 8$

Note that the powerset (set of all subsets) of $\{O_1\ O_2\ O_3\}$ has 8 elements:

$$P = \{\ \{\}\ \{O_1\}\ \{O_2\}\ \{O_3\}\ \{O_1\ O_2\}\ \{O_1\ O_3\}\ \{O_2\ O_3\}\ \{O_1\ O_2\ O_3\}\ \}$$

Vector time

A ***vector time*** identifies a database state by identifying the ***set of operations*** producing that state. In the following diagram the vector time (2,1,1) identifies the set of operations $\{ a_1\ a_2\ b_1\ c_1 \}$. If we think of the operations as events then a vector time is like a time stamp in the way it separates past and future events



A vector time is a multi-dimensional notion of time and is only partially ordered. Let i be a site identifier.

v[i]   = ith component of vector time v
            = number of operations generated by site i in the state associated with v

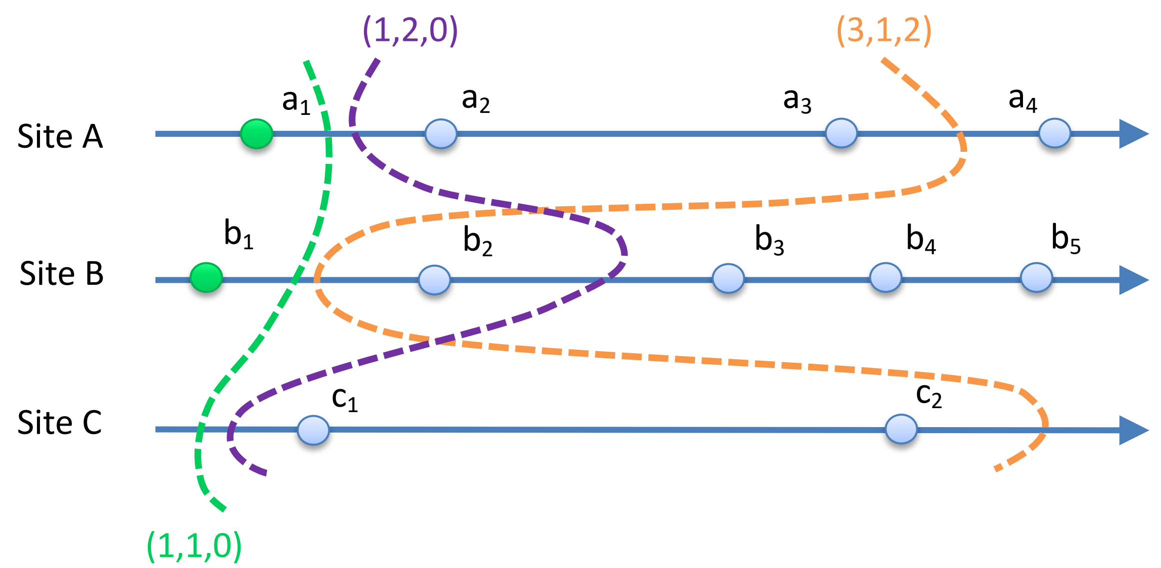
Let v_1, v_2 be vector times. We define a partial order (≤) on vector times as follows:

$v_1 \leq v_2 \quad \Leftrightarrow \quad \forall i\ v_1[i] \leq v_2[i]$

$\Leftrightarrow \quad$ (set of operations of v_1) $\subseteq$ (set of operations of v_2)

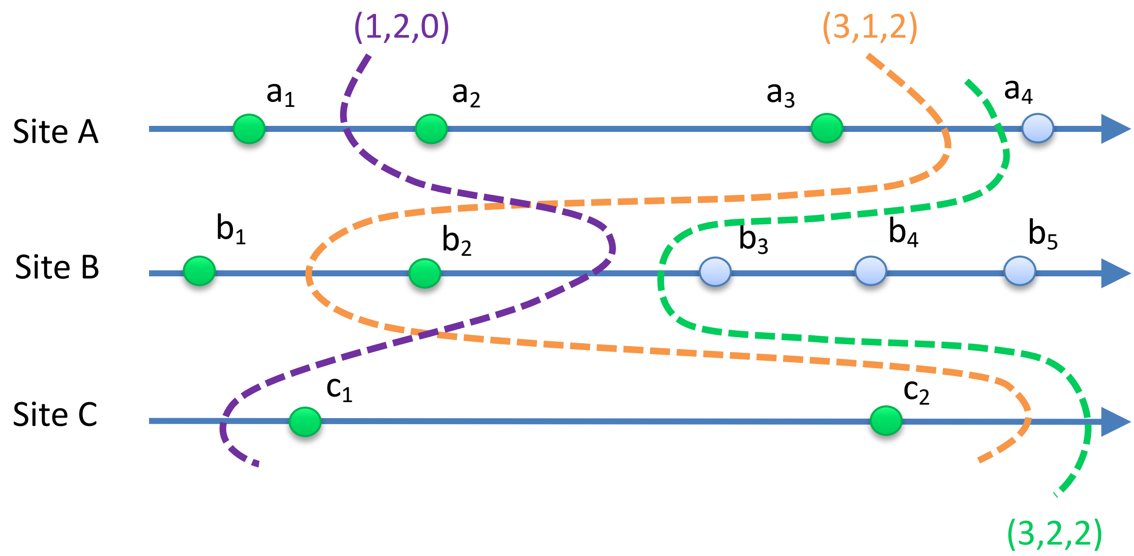
Intersection of two states

The intersection of two states means the operations in common (shown here in green) and corresponds to a green dotted line which "cuts" down through the left of the "cuts" of the two given states.



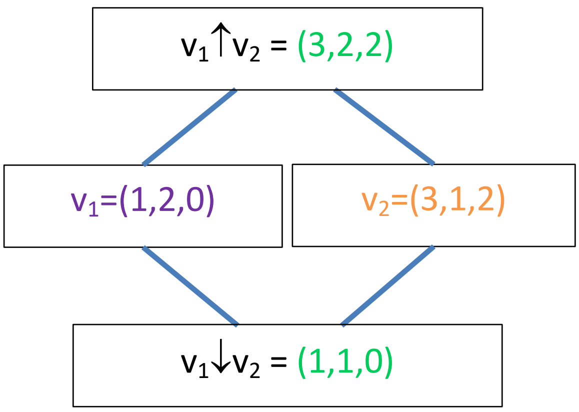
Union of two states

Similarly the union of two states corresponds to a cut down through the right of the cuts of the two states.



Meet and join of vector times

There are corresponding operations on vector times involving taking the minimum/maximum over the components.



join of v_1 and v_2

$(v_1 \uparrow v_2)[i] = \mathbf{max}(v_1[i], v_2[i])$

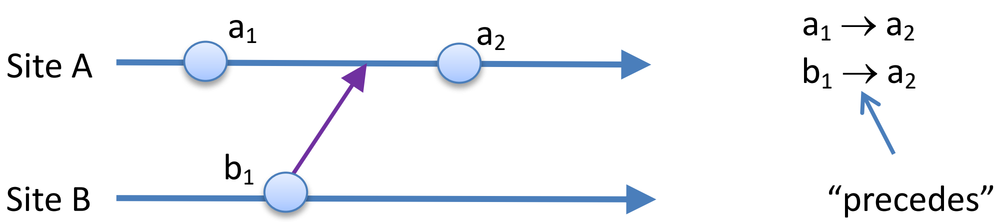
meet of v_1 and v_2

$(v_1 \downarrow v_2)[i] = \mathbf{min}(v_1[i], v_2[i])$

Precedes relation

There is a concept of an operation happening before another operation. We say O_1 ***precedes*** O_2 (and write this as $O_1 \rightarrow O_2$) if O_1 had been applied on the site where O_2 was generated.

In the following diagram operations a_1 and b_1 *precede* a_2 because both a_1 and b_1 had been applied on site A at the time a_2 was generated on site A.



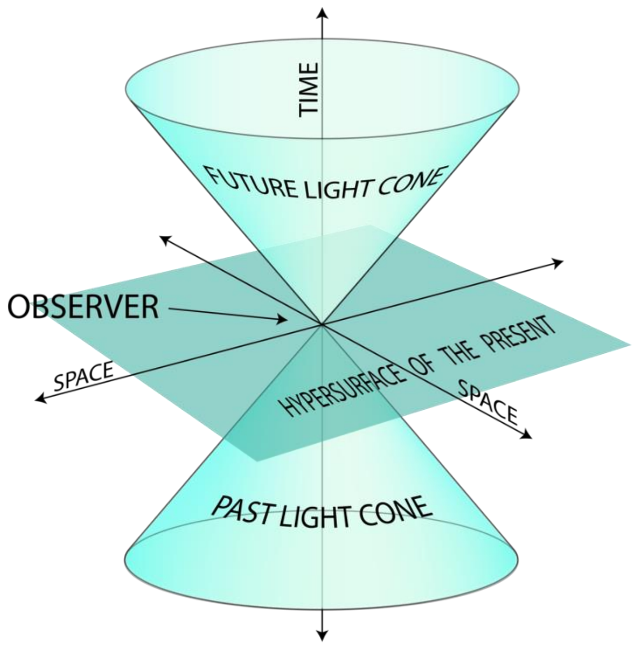
The precedes relation is a strict partial order on the operations:

- $\rightarrow$ is irreflexive
- $\rightarrow$ is asymmetric
- Causality preservation implies $\rightarrow$ is transitive

The precedes relation is not a total order, in fact we write $O_1 || O_2$ when O_1 and O_2 are ***concurrent***, meaning neither $O_1 \rightarrow O_2$ nor $O_2 \rightarrow O_1$.

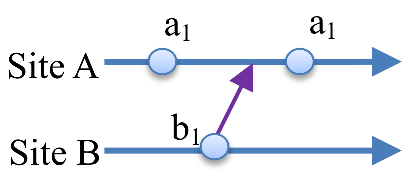
Analogy to special relativity

If one considers operations to be events that occur in space and time, then the precedes relation is analogous to the partial order on events in special relativity. In relativity events can only affect events on or inside its future light cone. Events that are not in each other's light cone are "space like" and are not comparable.



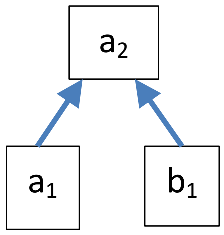
Don't confuse the two posets

Consider the following example where there are two sites:



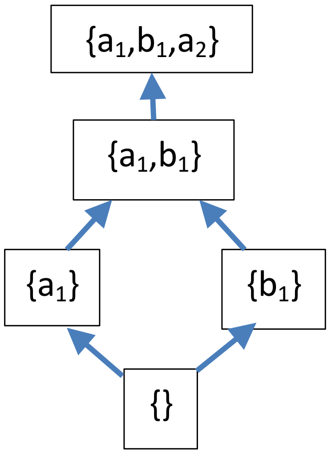
There are two different posets which are being discussed. On the left are the operations partially ordered by the precedes relation and on the right are the states/vector times/sets of operations partially ordered by the subset relation. The Hasse diagram on the right is the state transition diagram.

(operations,$\rightarrow$)



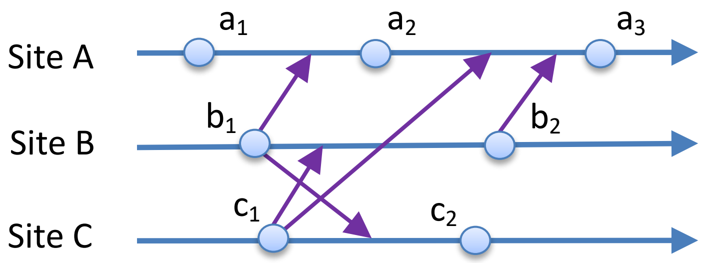
$a_1 \rightarrow a_2$
$b_1 \rightarrow a_2$

(states,$\subseteq$)



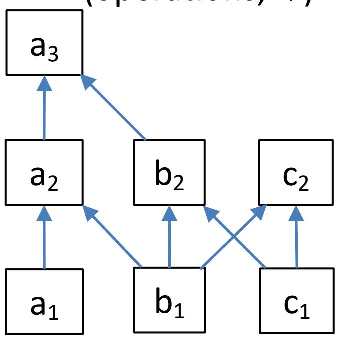
More complicated example

This is a more complicated example. Later in this document the state transition diagram will be shown.



precedes	*concurrent*
$a_1 \rightarrow a_2$	$a_1 \mid\mid b_1$
$b_1 \rightarrow a_2$	$a_1 \mid\mid b_2$
$b_1 \rightarrow b_2$	$a_1 \mid\mid c_1$
$c_1 \rightarrow b_2$	$a_1 \mid\mid c_2$
$b_1 \rightarrow c_2$	$b_1 \mid\mid c_1$
$c_1 \rightarrow c_2$	$c_1 \mid\mid a_2$
$a_1 \rightarrow a_3$	$a_2 \mid\mid b_2$
$a_2 \rightarrow a_3$	$a_2 \mid\mid c_2$
$b_1 \rightarrow a_3$	$a_3 \mid\mid c_2$
$b_2 \rightarrow a_3$	
$c_1 \rightarrow a_3$	

Hasse diagram of (operations,$\rightarrow$)

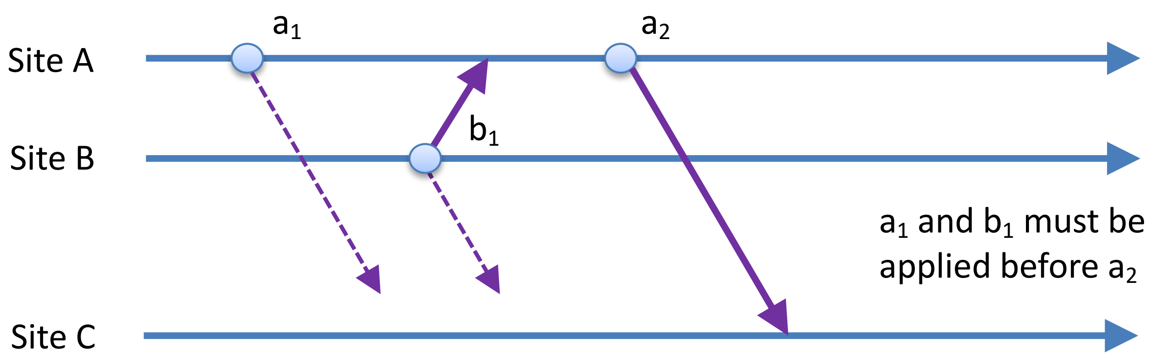


Causality Preservation

Causality preservation requires the following condition to be met:

If $O_1 \rightarrow O_2$ then O_1 is applied before O_2 at all sites

In this diagram a_1 and b_1 both precede a_2. Causality preservation requires that both a_1 and b_1 be applied before a_2 at all sites. So there is no point sending a_2 to site C before sending a_1 and b_1.

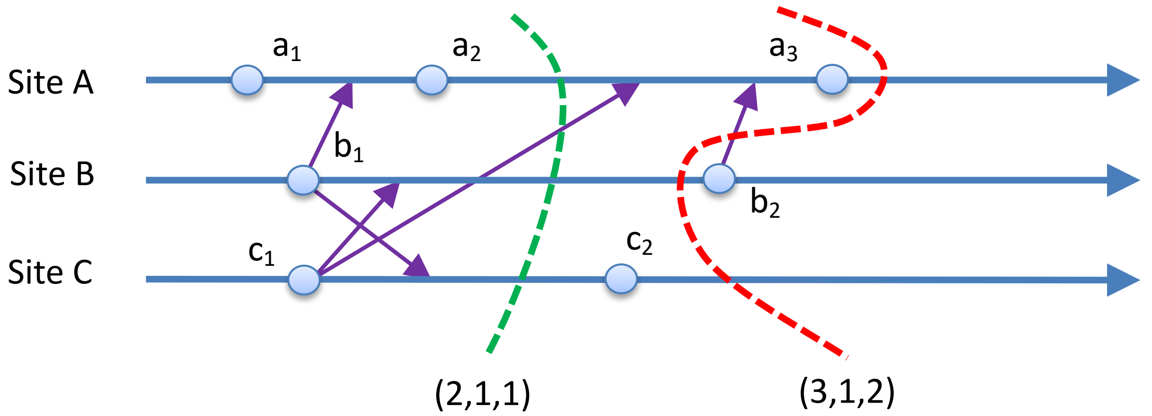


States that preserve causality

There's a corresponding definition of the states which preserve causality:

$$S \text{ preserves causality} \quad \Leftrightarrow \quad (O_2 \in S \ \wedge \ (O_1 \rightarrow O_2 \ \Rightarrow \ O_1 \in S))$$

The vector time shown in red doesn't preserve causality because there's an operation in its future b_2 which precedes an operation a_3 in its past

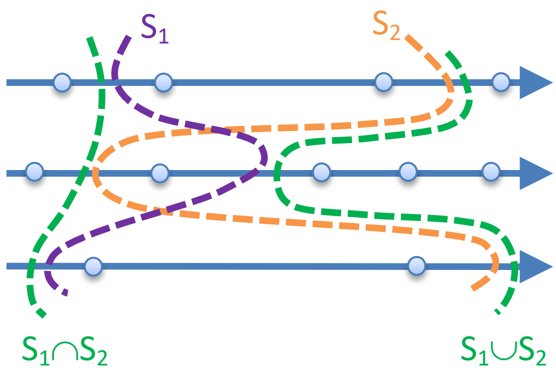


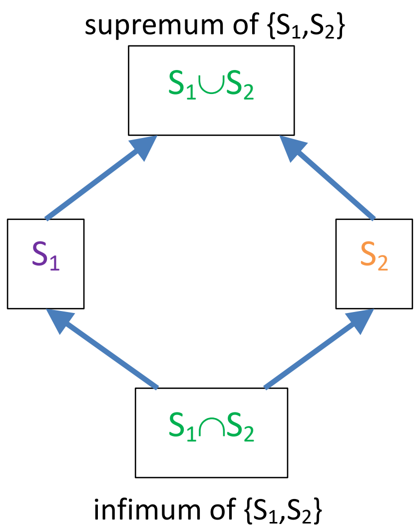
The states that preserve causality form a bounded lattice

It can be proven that the states which preserve causality are closed over intersection and union. In other words, let S_1 and S_2 be states then

$$S_1 \text{ and } S_2 \text{ preserve causality} \quad \Rightarrow \quad S_1 \cap S_2 \text{ and } S_1 \cup S_2 \text{ preserve causality}$$

$S_1 \cup S_2$ is the smallest set which contains both S_1 and S_2 so it's the supremum. $S_1 \cap S_2$ is the largest set contained by both S_1 and S_2 so it's the infimum.

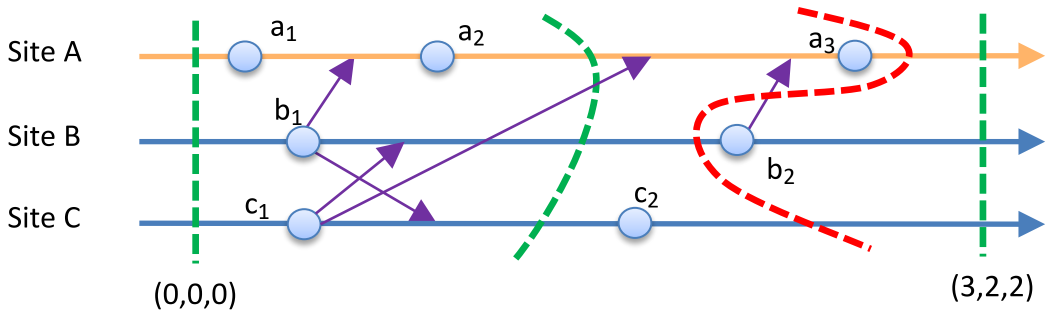


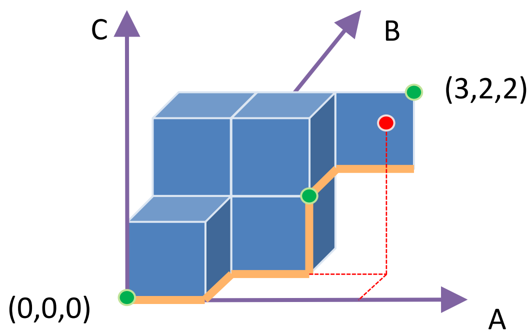


The existence of the infimum and supremum means the states that preserve causality form a mathematical structure called a *lattice.* This is shown in the lower part of the picture below having 5 cubes plus a flat face for this particular scenario with three sites.

This lattice has 24 vertices (i.e. vector times or states or sets of operations) which correspond to 24 different cuts in the top picture (only three are shown with green dotted lines).   The red vector time doesn't preserve causality and therefore doesn't correspond to a point on the lattice in the picture below.

Site A is applying operations in a particular order and this corresponds to one of the paths (the one drawn in orange) which is zig-zagging its ways though the lattice, starting at (0,0,0).





Paths through the lattice

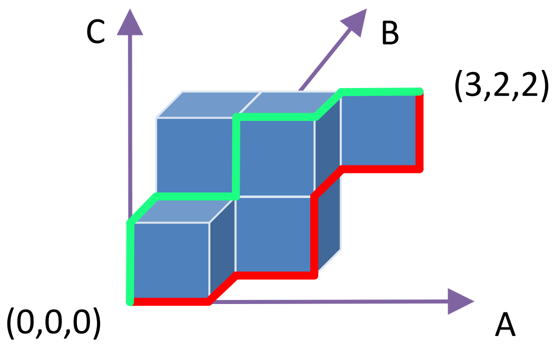
A path through the lattice is defined by an ordered list of operations. In the following diagram two alternative paths are shown in red and green.

$c_1\ b_1\ a_1\ c_2\ a_2\ b_2\ a_3$

$a_1\ b_1\ a_2\ c_1\ b_2\ a_3\ c_2$

constrains order ←

$a_1 \to a_2$
$b_1 \to a_2$
$b_1 \to b_2$
$c_1 \to b_2$
$b_1 \to c_2$
$c_1 \to c_2$
$a_1 \to a_3$
$a_2 \to a_3$
$b_1 \to a_3$



Causality preservation ⇒ total order respects the partial order defined by the precedes relation.

i.e. $O_1 \to O_2 \Rightarrow O_1$ appears before O_2 in every list

Even though there are 7 operations in this example, there are not 7!=5040 paths through the lattice because the great majority of them break causality. There are 12+40+40+6+12+4 = 114 paths which can be taken by a site, depending on the order in which it receives the operations. These are the permutations of the 7 operations which respect the partial order given by the precedes relation.

Note as well that there are not 2^7 = 128 states, there are only 24 states.

Appendix: Mathematical background

Partially ordered set (mathematics)

(P,≤) is a ***poset*** if binary relation ≤ on P is a ***partial order***:

- x≤x (reflexive)
- x≤y & y≤x ⇒ x=y (antisymmetric)
- x≤y & y≤z ⇒ x≤z (transitive)

≤ is a ***total order*** if also:

- x≤y or y≤x

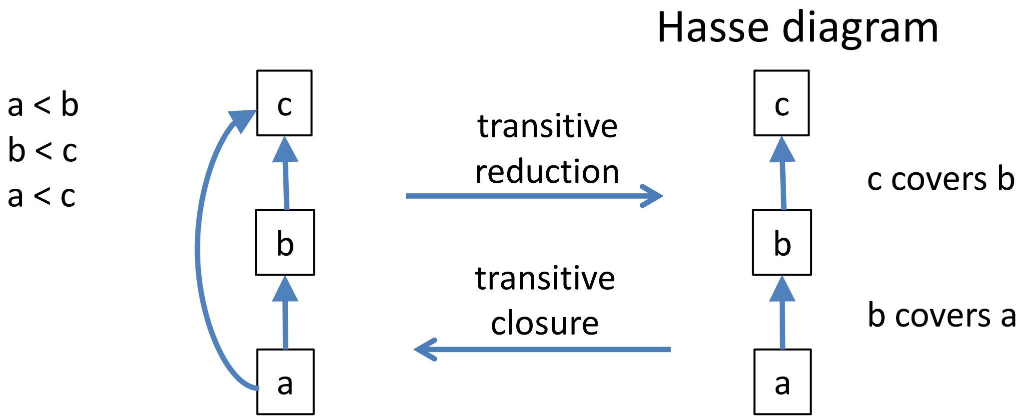
< is a ***strict partial order*** if

- not x<x (irreflexive)

- $x<y \Rightarrow$ not $y<x$ (asymmetric)
- $x<y$ & $y<z \Rightarrow x<z$ (transitive)

Hasse diagram of a finite poset

The ***Hasse diagram*** of a finite poset involves taking the transitive reduction which eliminates "redundant" edges in the graph. We say y covers x if $x<y$ and $\nexists z\ x<z<y$



Infimum and supremum

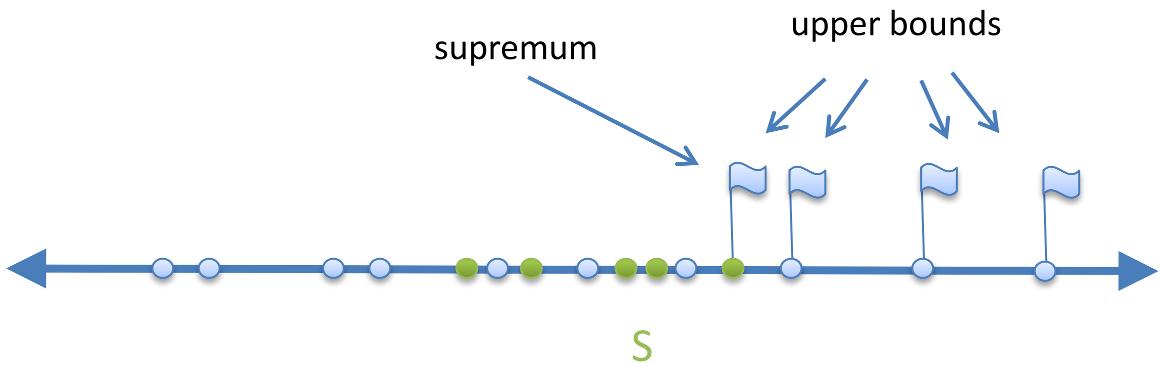
The infimum and supremum of a given subset of a poset are unique if they exist. The supremum is the lowest of all the upper bounds and the infimum is the greatest of all the lower bounds.

Let $(P,\leq)$ be a poset. Let $S\subseteq P$

u is an ***upper bound*** of S if $x\in S \Rightarrow x\leq u$

s is the ***supremum*** of S if

- s is upper bound of S and
- u is upper bound of $S \Rightarrow s\leq u$



Lattice (mathematics)

A ***lattice*** is a poset in which every two elements have a supremum and an infimum

s = sup {x, y} means:

- $x\leq s$ & $y\leq s$; and (s is an upper bound)

- $\exists u \; x \leq u \;\&\; y \leq u \;\Rightarrow\; s \leq u$ (s is the least upper bound)

A ***bounded lattice*** has least and greatest elements