Operational Transform Convergence

David Barrett-Lennard
May 2005

Syntactic conventions

O, O_1, O_2 etc are *primitive* (ie indivisible) operations.

L, L_1, L_2 etc are lists of operations.
$L = [O_1 , O_2 , O_3 , \ldots , O_n]$ is the list of operations O_1 then O_2 ... then O_n.
$L = [L_1, L_2, \ldots, L_n]$ is the concatenation of lists $L_1, \ldots, L_n$
$L = [H \mid T]$ is the list with head operation H and tail T

Xi is either an Oi or Li.
$L = [X_1\ X_2 \ldots X_n]$ is the concatenation of lists or primitive operations $X_1, \ldots, X_n$

Definitions

Definition: We write $X_1 \sim X_2$ if X_1 and X_2 are executed on the same initial document state and produce the same final document state.

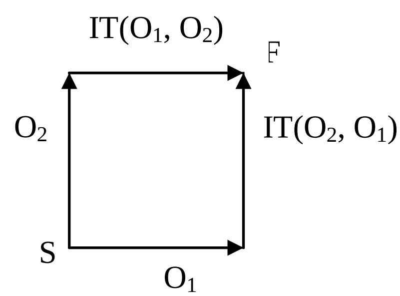
Clearly: $\forall i, X_i \sim Y_i \Rightarrow [X_1 \ldots X_n] \sim [Y_1 \ldots Y_n]$

Definition: $IT(O_2, O_1)$ is the *inclusion transform* of O_2 against O_1. O_1,O_2 are assumed to be concurrent operations on the same initial document state. $IT(O_2, O_1)$ is a transformed version of O_2 that can be executed after O_1.

Definition: Let $O_{1,2} = [O_1 , IT(O_2, O_1)]$

Definition: Convergence condition C_1 : $\forall\ O_1,O_2,\ O_{1,2} \sim O_{2,1}$

The following diagram depicts the two operations O_1, O_2 that apply to initial document state S. Condition C_1 states that the path along the left and top edges yields the same final document state F as the path along the bottom and right edges.

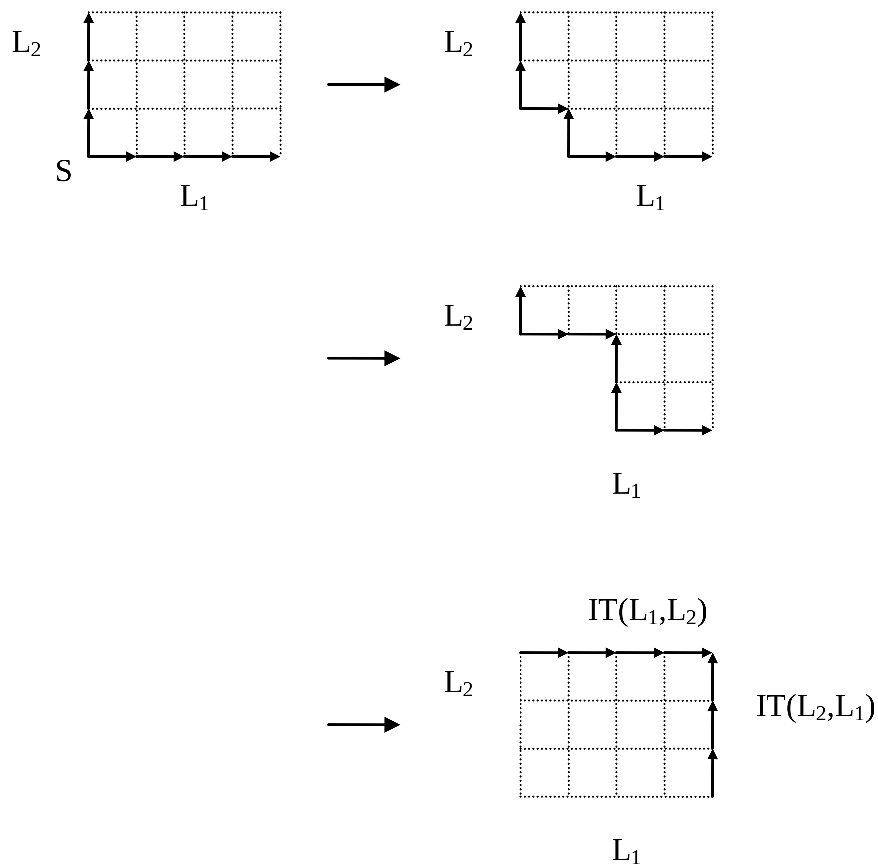


Definition: $IT(L_1, L_2)$ is defined recursively as follows

1. $IT([], []) = []$
2. $IT([L_1, L_2], L_3) = [\ IT(L_1,L_3) , IT(L_2, IT(L_3, L_1))\ ]$
3. $IT(L_1, [L_2 , L_3]) = IT(IT(L_1, L_2), L_3)$

[Note that this definition appears to be ambiguous in the way in which a list is decomposed into smaller sub-lists.   However, it turns out that the final result is independent of the order in which the lists are decomposed.   A more formal approach could begin with an unambiguous definition then prove decomposition rules 2,3 above]

We can depict $IT(L_1,L_2)$ pictorially as follows. L_2 has three primitive operations and L_1 has four primitive operations. The initial document state S is associated with the bottom left corner of the 3 x 4 grid of "cells".

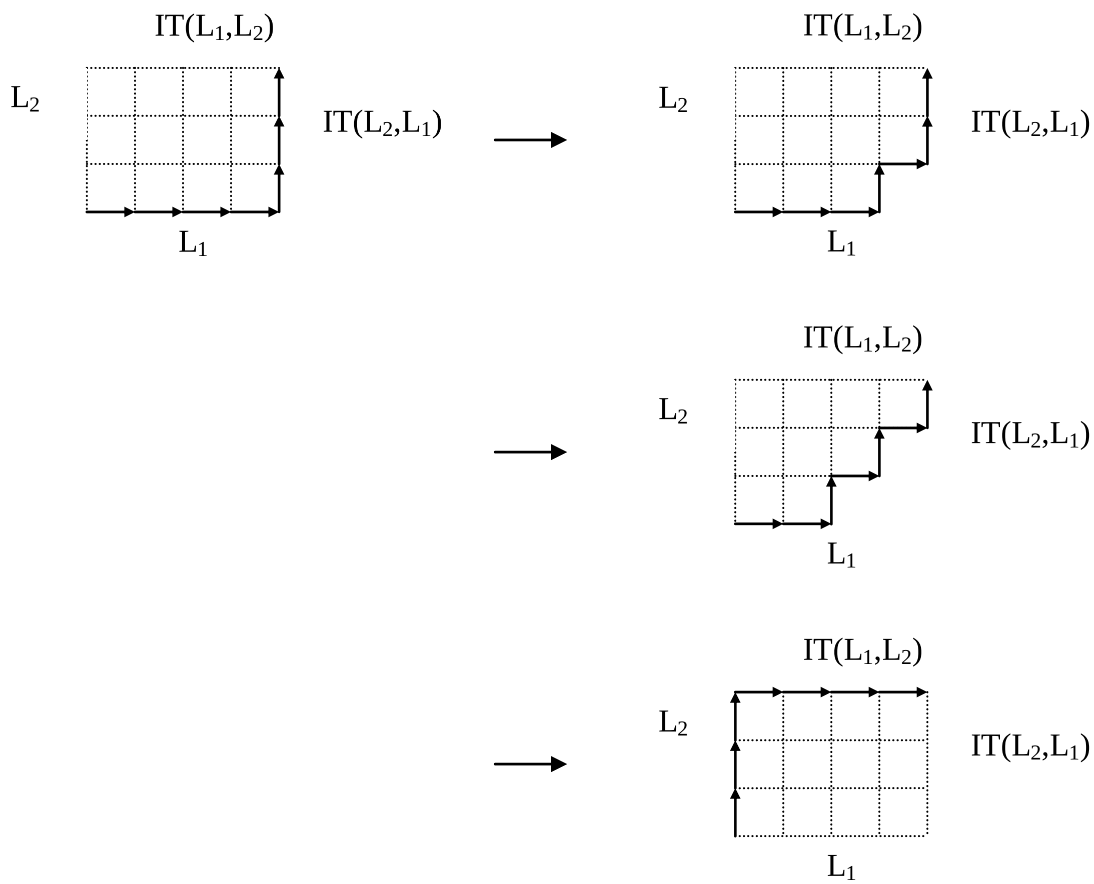


Starting from the bottom left corner we can apply IT on the primitive operations.   This can proceed to adjacent cells until we transform the operations to the top and right boundaries.

Note that IT is quadratic in the size of the lists.

Definition : $L_{1,2} = L_1\ IT(L_2, L_1)$

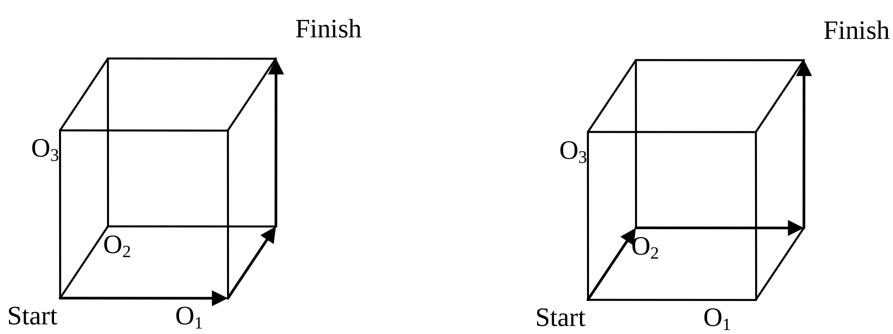
Lemma 1: $C_1 \Rightarrow \forall\ L_1,L_2,\ \ L_{1,2} \sim L_{2,1}$
Proof :
A formal proof can be achieved using induction on the length of L_1,L_2.   Informally, we can prove the result more easily as follows



We begin with $L_{1,2}$, which is associated with the path along the bottom and right edges. Starting from the bottom right corner we can shift the path using the known convergence property on the cell. This is repeated for adjacent cells until the path is transformed to the left and top edges, corresponding to $L_{2,1}$.

Definition: $O_{1,2,3} = O_{(1,2),\,3} = [O_{1,2}\ \ IT(O_3, O_{1,2})\ ]$

This represents one of the six possible paths from "Start" to "Finish" on the following cube.

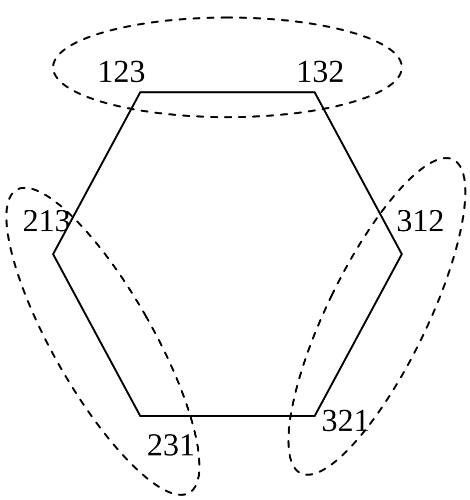


Path $O_{1,2,3}$ Path $O_{2,1,3}$

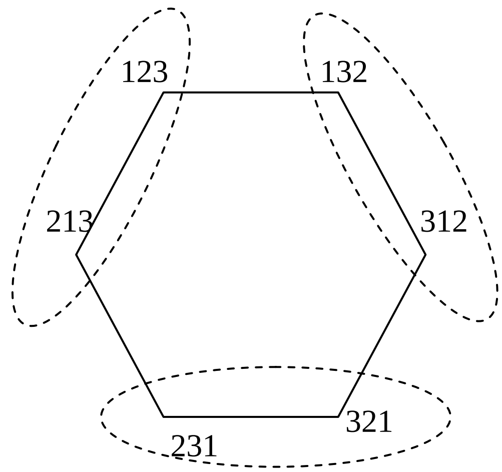
Definition: Condition C_2: $\forall\ O_1,O_2,O_3,\ \ O_{1,2,3} \sim O_{2,1,3}$

Note that it is *not* the case that $C_1 \Rightarrow C_2$. On the contrary, C_2 must be independently assumed in order to achieve convergence when there are more than two sites.

There are 6 possible paths on the cube...



Grouping by C_1



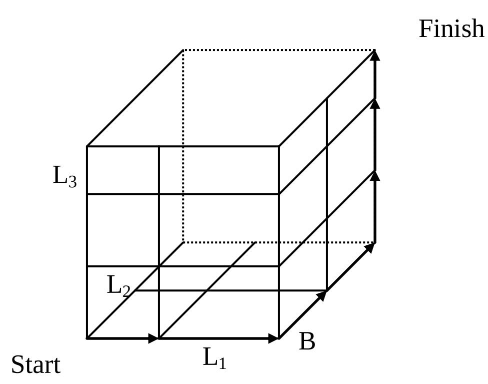
Grouping by C_2

Convergence C_1 implies that the 6 paths can be put into 3 groups, illustrated with the dotted ovals on the left side.
Adding condition C_2 implies that all 6 paths are equivalent.

Lemma : $C_1 \Rightarrow L_{1,2,3} \sim L_{1,3,2}$
Note that it is not the case that $C_1 \Rightarrow L_{1,2,3} \sim L_{2,1,3}$

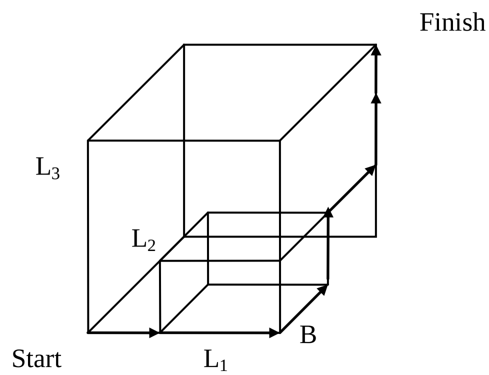
Lemma 2: $C_1, C_2 \Rightarrow \forall L_1, L_2, L_3,\ L_{1,2,3} \sim L_{2,1,3}$
Proof: (Informally)

The following cube depicts two operations in L_1, two in L_2 and three operations in L_3.

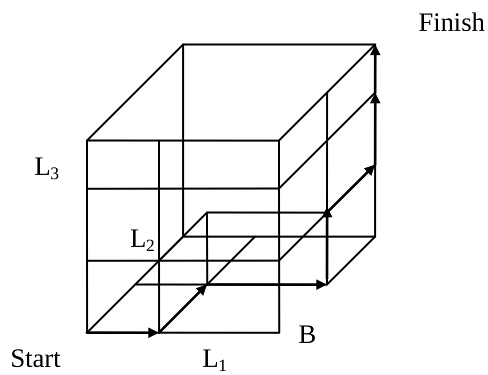


$L_{1,2,3}$ corresponds to the transformation of L_3 along the front face then the right face of the cube. On each of these faces we can depict a regular grid of cells, associated with the simple, two dimensional case. Condition C_1 ensures that convergence is achieved on each face of the cube.

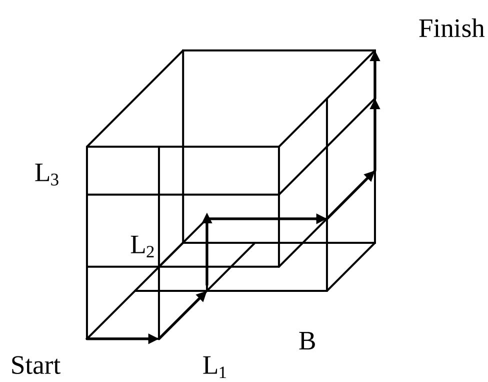
The individual operations allow us to partition the cube into small "cells". In particular the "cell" at the nearest, bottom right corner of the cube (ie labeled B in the diagram) is of interest. Consider that we adjust the path on the right face as follows



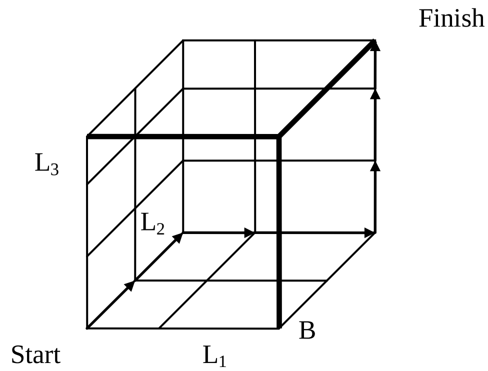
We can now apply condition C_2 to the cell at B to yield an equivalent path as follows



We now adjust the path on the back face of the cell at B



This allows us to proceed to the cell to the left of B. In this manner we can proceed through the cells until we get the following path



which corresponds to $L_{2,1,3}$

This shows informally that $L_{1,2,3} \sim L_{2,1,3}$

RTP. $C_1, C_2 \Rightarrow L_{(1,2),3} \sim L_{1,(2,3)}$ (associativity)
Proof :

$$
\begin{aligned}
L_{(1,2),3} &\sim L_{(2,1),3} && \text{(Lemma 2)} \\
&= L_{2,1}\ \ IT(L_3,\ L_{2,1}) \\
&\sim L_{1,2}\ \ IT(L_3,\ L_{2,1}) && \text{(Lemma 1)} \\
&= L_{1,2}\ \ IT(L_3,\ [L_2\ \ IT(L_1, L_2)]\ ) \\
&= L_{1,2}\ \ IT(IT(L_3,L_2), IT(L_1, L_2)) \\
&= L_1\ \ IT(L_2,L_1)\ \ IT(IT(L_3,L_2), IT(L_1, L_2)) \\
&= L_1\ \ IT([L_2\ \ IT(L_3,L_2)\ ],\ L_1) \\
&= L_1\ \ IT(L_{2,3},\ L_1) \\
&= L_{1,(2,3)}
\end{aligned}
$$

Four sites
There are 4! = 24 paths through the "hypercube". Interestingly it turns out that C_1,C_2 are sufficient for convergence.

The paths can be listed in four groups of 6 as follows

G1	G2	G3	G4
1 2 3 4	2 1 3 4	3 1 2 4	4 1 2 3
1 2 4 3	2 1 4 3	3 1 4 2	4 1 3 2
1 3 2 4	2 3 1 4	3 2 1 4	4 2 1 3
1 3 4 2	2 3 4 1	3 2 4 1	4 2 3 1
1 4 2 3	2 4 1 3	3 4 1 2	4 3 1 2
1 4 3 2	2 4 3 1	3 4 2 1	4 3 2 1

Claim
- Each group contains equivalent paths because they all begin with the same operation which can therefore be disregarded, reducing the problem to the 3-site case which has already been proven
- As an example G1 ~ G2 because $L_{1,2,3,4} \sim L_{2,1,3,4}$. This follows by letting $M = L_{3,4}$. By associativity $L_{1,2,3,4} \sim L_{1,2,M} \sim L_{2,1,M} \sim L_{2,1,3,4}$.

This pattern extends into n sites.