Science vectorielle

"La science c'est le plaisir de discuter pour comprendre"

Gianni Mocellin

Straco
www.straco.ch
18.02.2024, 05h00

Introduction

Une science universelle.

Symboles

Conjoindre

+ : adjoindre

- : subjoindre

Conjecter

^ : éjecter

• : cojecter

→ : enjecter

← : enjecter

Composer

**** : imposer***

/ : opposer

⇘ : commuter

⇖ : anticommuter

Adapter

× : moduler

÷ : inverser

|q| : magnifier

∼ : renverser

≁ : involuer

Isoler

⊳ : extraire

⊙ : complémenter

⊚ : implémenter

Sélectionner

u a (ab) b u : croiser

u (a ab b) u : unir

u (a) ab (b) u : désunir

(u) a (ab) b (u) : étendre

Formules

L'originalité

Les conjections

Magnitude d'une mono 67

magnitude = mono • mono

cojecter[mono, mono]

Angle de deux monos

Cosinus (angle)
=
mono1 • mono2 / magnitude1 • magnitude2

opposer[injecte[mono1, mono2], injecter[magnitude1, magnitude2]]

Magnitude d'une exo

magnitude = cojecter[exo, exo-renversée]

exo • exo-renversée

Angle de deux exos

cosinus (angle)

=

*exo1 → exo2-renversée / magnitude1 * magnitude2*

exo1 INJECTER exo2-renversée OPPOSER magnitude1 IMPOSER magnitude2

OPPOSER[INJECTER[exo1, exo2-reversée], IMPOSER[magnitude1, magnitude2]]

Enjection 71

(exoX ⊕ exo1) ⊖ exo2

(exoX EJECTE exo1) INJECTE exo2
=
exoX INJECTE (exo1 $INJECTE_{GD}$ exo2)

INJECTE[exoX, $INJECTE_{GD}$[exo1, exo2]]

Complexité de l'injection de deux exos

(exo1 $INJECTER_{GD}$ exo2)$_k$

$INJECTER_{GD}$[exo1, exo2]$_k$

=

$k_{exo1} - k_{exo2}$

Injection

exo2 INJECTER (exo1 EJECTER exoX)
=
(exo1 $INJECTER_{DG}$ exo2) INJECTER exoX

INJECTER[$INJECTER_{DG}$[exo1,exo2], exoX]

Les oppositions

Opposée d'une mono

1 ↘ *mono*

1 OPPOSE[mono]

=

mono ⊖ (mono ⊗ mono)

mono OPPOSE (mono IMPOSE mono)

OPPOSE[mono, IMPOSE[mono, mono]]

=

Opposée d'une exo

1 ⨸ *exo*

OPPOSER (exo)
=

exo-renversée ⨸ (exo ⊖ exo)

exo-renversée OPPOSER (exo INJECTER exo)
=
(-1) k (k–1)/2 IMPOSER exo OPPOSER magnitude IMPOSER magnitude

Opposé de l'omnio

OPPOSER (omnio)
=
RENVERSER (omnio)

Complémentarités

Complémentation

Co-exo
=
exo INJECTER$_{GD}$ *OPPOSER omnio*

Dualité 82

(exo1 EJECTER exo)
=
exo1 INJECTER$_{GD}$ *RENVERSER exo2*

RENVERSER exo1 INJECTER$_{GD}$ *exo2)*
=
exo1 EJECTER INVERSER exo2

Projections

Projection d'une mono dans une exo

PROJETTE (mono, exo)
=
(mono INJECTE exo) INJECTE OPPOSE exo

Projection d'une exo dans une exo

PROJETTE (exo1, exo2)
=
(exo1 INJECTE exo2) INJECTE OPPOSE exo2

Rendu plus explicite d'un élément d'exo plutôt qu'un élément de 1/exo

PROJETTE (exo1, exo2) =

(exo1 INJECTE OPPOSE exo2) INJECTE exo2

L'injection d'une exo1 dans une exo2 est la sous-exo de exo2 de complexité $k_{exo2} - k_{exo1}$ qui est le complément par exo2 de la projection de exo1 dans exo2

Générations variées 99

Intersections et unions 125

Composition 141

Imposition d'une mono à une mono

mono1 IMPOSER mono2
=
mono1 INJECTER mono2 + mono1 EJECTER mono2

Projection d'une mono dans une mono

PROJETER (mono1, mono2)
=

(mono1 INJECTER mono2) OPPOSER mono2

Alterojection d'une mono hors d'une mono

ALTEROJECTER (mono1, mono2) =

(mono1 EJETER mono2) OPPOSER mono2

Projection d'une mono dans une exo

PROJETER (mono, exo) =

(mono INJECTER exo) OPPOSER exo

Alterojection d'une mono hors d'une exo

ALTEROJECTER (mono, exo)
=
(mono EJECTER mono) OPPOSER exo

Projection d'une exo dans une exo

PROJETER (exo1, exo2)
=
(exo1 INJECTER exo2) OPPOSER exo2

Alterojection d'une exo hors d'une exo

ALTEROJECTER (exo1, exo2)
=
(exo1 INJECTER exo2) OPPOSER exo2

Transjections

Transjection d'une mono à travers une mono

TRANSJECTER (mono1, mono2) =

OPPOSER mono1 IMPOSER mono2 IMPOSER mono1

Transjection d'une exo à travers une mono

TRANSJECTER (exo, mono) =

mono IMPOSER exo OPPOSER mono

Transjection d'une mono à travers une exo

TRANSJECTER (mono, exo)
=
IMPOSER (mono EJECTER exo) OPPOSER exo
=
– INVERSER exo IMPOSER mono OPPOSER exo

Transjection d'une exo à travers une exo

TRANSJECTER (exo1, exo2)
=
*(-1) * (e+1) IMPOSER exo2 IMPOSER exo1 OPPOSER exo2*

Transjection d'une exo à travers une co-exo

TRANSJECTER (exo, co-exo)
=
*(-1) * xxx IMPOSER co-exo IMPOSER exo OPPOSER co-exo*

Interjection d'une exo entre une exo

INTERJECTER (exo1, exo2) =

*(-1) * (e+1) IMPOSER exo2 IMPOSER exo1 OPPOSER exo*
2

Roteur

Un roteur est une imposition d'un nombre pair de monos tels que:

roteur IMPOSER INVOLUER roteur

=

1

Rotation d'une mono

mono1 ROTER (mono2, mono3)
=
mono3 OPPOSER mono2 IMPOSER mono2 OPP mono3
=
orienteur IMPOSER mono1 OPPOSER orienteur

Rotation d'une exo

ROTER (exo)
=
roteur IMPOSER exo OPPOSER roteur

Composition de rotations

roteur-total
=
roteur1 IMPOSER roteur2 IMPOSER roteur3

Adaptateur

Adaptation d'une mono 192

adaptateur
=
mono IMPOSER mono IMPOSER .. IMPOSER mono

Adaptation d'une exo

ADAPTER (exo)
=
*(-1) * t IMPOSER adaptateur IMPOSER exo OPPOSER adaptateur*

Adaptateurs pairs

adaptateur IMPOSER exo OPPOSER adaptateur

Adatptateurs impairs

adaptateur IMPOSER RENVERSER exo OPPOSER adaptateur

Différentiation 213

Commutateur

Différentiation numérique

Différentiation numérique d'une fonction d'hétéro

Différentiation variationnelle 230

* =

Différenciation hétérovariationnelle 235

La latéralité

Les rectités

La versalité

Opposée de l'omnio

OPPOSER omnio

=

INVOLUER omnio

Les pointos 360

Les pointos sont des radios nulles, de rayon nul

Les radios nulles représentent des pointos finies et infinies

radio

=

Substance IMPOSER (originité ADJOINDRE mono ADJOINDRE ½ mono IMPOSER mono IMPOSER infinité)

Les uni-pointos sont les pointos dont la magnitude de la substance vaut 1

Position relative à une autre position 454

di-pointance =

pointo1 – pointo2 =

pointo1 – pointo2 + ½ (pointo1 INJECTER pointo2) OPPOSER infinité

Les rectos

co-recto
=
*omnio + di-pointance-recto * infinité*

Les radios

radio-tangible

=

Substance IMPOSER (mono + ½ IMPOSER rayon IMPOSER rayon IMPOSER infinité)

co-radio-intangible

=

Substance IMPOSER (mono - ½ IMPOSER rayon IMPOSER rayon IMPOSER infinité

Les rectos et les orientations 370

k-recto

=

originité EJECTER k-exo EJECTER infinité

Les évolution

Les évolutions de l'originalité sont représentables par des évolueurs dans l'universalité

Toutes les évolutions de l'originalité peuvent être représentées par des transjections multiples à travers des bi-mono

Les repositions

positionneur

=

1 – di-pointance IMPOSER originité OPP

Les réorientations à l'origine

roteur

=

Cosinus (ambivalence/2) – Sinus (ambivalence/2) IMPOSER bi-exo

roteur

=

Exponentielle (– ambivalence/2 IMPOSER bi-exo)

Les évolutions générales peuvent être construites en faisant d'abord

- une rotation à l'origine, puis

-un déplacement

ce qui donne un roteur dans un plan de rotation

évolueur

=

*déplaceur * roteur*

=

*(1 – di-pointance IMPOSER originité OPPOSER 2) * Exponentielle (– ambivalence/2 IMPOSER bi-exo)*

Les pointos, les rectos et les orientations

Les pointos

pointo

=

Substance IMPOSER (pointo EJECTER … EJECTER pointo)

Les rectos

recto

=

Substance IMPOSER (pointo EJECTER … EJECTER pointo EJ infinité)

k-recto
=
Substance (recto EJECTER … EJECTER recto)

Les rectos originées

recto-originée
=
originité EJECTER k-exo EJECTER infinité

Les rectos positées

recto-positée

=

pointo EJECTER k-exo EJECTER infinité   372

L'omnio

uni-omnio

=

originité EJECTER ori-omnio EJECTER infinité

Co-recto

k-co-recto

=

– pointo INJECTER (OPPOSER k-exo IMPOSER infinité)

Orientation

k-orientation

=

k-exo EJECTER infinité

Co-orientation

k-co-orientation

=

– 1 k-endo EJEJECTER infini

opposée uni-positée

=

infinité EJECTER OPPOSER uni-recto EJECTER infinité

Les radios

Une k–1 radio est contenue dans une k-radio, un conteneur pour elle

Une radio en 3D est une 2-variété courbe, une 2-radio

- un cercle est une 1-radio sur une 2-radio

- une bi-pointo est une 0-radio sur une 1-radio et elle est unique en ce sens qu'elle est n'est composée que de deux pointos séparées

Radios à l'origine 398

co-radio

=

originité – ½ IMPOSER rayon IMPOSER rtayon IMPOSER infinié

Co-radio à l'origine

co-radio

=

co-posito EJETER co-recto

(originité – ½ IMPOSER rayon IMPOSER rayon IMPOSER infinité) EJECTER co-bi-recto

Co-bi-pointo à l'origine

cobipointo =

co-radio EJECTER co-recto-positée EJECTER co-recto-originée

Radio originée

radio originée

=

(originité + ½ IMPOSER rayon IMPOSER rayon IMPOSER infinité) IMPOSER k-exo

Radio positées

radio positée

=

(centro + ½ IMPOSER rayon IMPOSER rayon IMPOSER infinité) EJECTER (– centro INJECTER (k-exo IMPOSER infinité))

Co-radio positée

Co-radio positée

=

(centro + ½ IMPOSER rayon IMPOSER rayon IMPOSER infinité) EJECTER (– centro INJECTER (k-exo IMPOSER infinité))

Les transitos 451

Transito libre

Une radio sans pointo

Origo

=

mono EJECTER infinité

Transito alignée influence

ligne
=
originité EJECTER exo EJECTER infinité

Une évolution se fait par tranjection à travers deux rectos parallèles

Elle peut aussi se faire par deux centro parallèles

C'est le produit de deux inversions

Transito normale co-transitée

La co-transito d'une recto à l'origine est une exo

Elle donne une orientation

Pour lui appliquer une réorientation

roteur (exo)

Pour lui appliquer un roteur de translation

positeur (exo)

=

pointo INJECTER (endo EJ infinité)

=

mono + (pointo INJECTER mono) IMPOSER infinité

Transito tangente entransité

Une mono attachée à l'origine

Transito

=

originité EJECTER exo

Dans une réorientation, l'exo est le seul aspect rotationnel

ROTER (*endo EJECTER mono*)

=

orginité EJECTER ROTER (*mono*)

Une translation emporte à la fois la pointo et la mono

PTRANSLATET [*originité EJECTER mono*]

=

PTRANSLATER(*originité* EJECTER (*originité* INJECTER (*mono* EJECTER *infinité*)))
=

pointo EJECTER (*pointo* INJECTER (*mono* EJECTER *infinité*))

Les co-rectos

La co-recto d'une idée est la plus petite recto qui la contient

Une recto est donc sa propre co-recto:

co-recto

=

infini # infini

La co-recto d'une radio, qui peut être une tango est:

contenence

=

radio # infini

Les tando n'ont pas de rectos tangentes bien qu'elles aient une contenence.

Les fluido

Les enfluité passent par une posité de l'idée, infinité ou centrité.

Les fluido d'une infinité

fluido =

pointo $INJECTER_G$ *recto-involuée*

Fluido d'une radio

tangence en une radio =

fluido =

pointo INJECTER$_G$ radio-involuée

Enfluo d'une recto

tangence en une recto =

fluo =

pointo INJECTER$_G$ radio-involuée

Enfluo d'une centrité

tangence en une radio =

fluo =

pointo INJECTER$_G$ radio-involuée

Les enradio

Une radio ou co-radio peut être entourée par la plus petite radio la contenant

en-radio

=

radio OPPOSER (radio EJECTER infinité)

en-centrité

=

(– 1) k IMPOSER co-radio-involuée OPPOSER infinité INJECTER$_G$ co-radio

Une radio peut être décomposée en une imposition de son environnence et de sa contenence.

factorisation

=

radio OPPOSER (radio EJECTER infinité) INJECTER radio EJECTER infinité

Adatations conformes 495

Préservent les angles, donc la forme de l'idée

Elles sont générées par des adaptateurs, c'est-à-dire des impositions de latités

Dilatation

Une translation se fait par tranjection à travers deux tendences parallèles

Elle peut aussi se faire par deux radités parallèles. C'est le produit de deux inversions

Contraction

Résumé

On utilise les verbes à

l'infinitif

lors de la définition d'une opération

*o-per-at-**e***

et non le nom de l'opération

*o-per-at-**ion***"

et on utilise le suffix

-ing

pour marquer le fait que l'opération fait partie d'un work-flow

c'est-à-dire pour parler de l'activité elle-même

*o-per-at-**ing***

En français

adjoindre[q1,q2]

disjoindre[q1,q2]

éjecter[q1,q2]

cojecter[q1,q2]

injecter[q1,q2]

imposer[q1,q2]

opposer[q1,q2]

En anglais

adjoin[q1,q2]

disjoin[q1,q2]

eject[q1,q2]

coject[q1,q2]

inject[q1,q2]

impose[q1,q2]

oppose[q1,q2]

L'origine s'appelle

originité

L'infini s'appelle

infinité

L'univers est représenté dans la pensée par

un certain nombre n de propriétés

Dans ces *propriétés* la pensée peut distinguer

des unités

L'ensemble des propriétés et de leurs unités respectives est rangé dans un Dataset qui s'appelle *originality*

originality = Dataset[partities, unities]

La pensée construit ensuite

une centralité

qui contient

le nombre *1* dont a complexité *k* vaut *0*

permettant de combiner toutes les unités tout avec les nombres

et

toutes les combinaisons possibles des *partities* de *l'originality*

Subset[1, partities, combinaisons de l'originality] = centrality

Si on ajoute *l'originité* derrière le *1* à cette centralité on obtient

une latéralité

Ajouter[originité, centralité] = latéralité

Si on ajoute à la latéralité l'infinité on obtient

une versalité

Ajouter[infinité, relativité] = versalité

On peut multiplier les unités

u

par des magnitudes

m

ce qui lui donne

des oro

m x u = oro

La complexité des oro est de 1

Les quantités sont donc pensées comme

modulables

par une magnitude *m*

On peut conjoindre, c'est-à-dire adjoindre ou disjoindre

des *oro*

pour en faire

des monos

et dont la complexité *k* est de *1*

Les quantités doivent être multipliables par

une unité monétaire monu

pour obtenir

une valeur monétaire

q x monu = monomo

L'éjection consiste à éjecter des quantités les unes des autres

ce qui en augmente la complexité en produisant

des exos

de plus en plus complexes

Il en va de même des *monomo* qui deviennent de plus en plus complexes

On note alors la quantité par une lettre majuscule

Q

parceque sa complexité est supérieure à 1

éjecter[q1,q2] = Q

Dans le cas de valeurs monétaires monomo

Ejecter[monomo1,monomo2] = exomo QM

Complémentairement à l'éjection existe l'injection

Si les quantités considérées ont la même complexité k, l'injection se résume à

une cojection

notée

"↔"

(produit scalaire)

Si la complexité k est 1, par exemple

q1 ↔ q2 = m

cojecter[q1,q2] = m

m est le cosinus de

l'angle

entre les deux monos q, c'est-dire-un angle

Si la complexité k de la quantité est supérieure à 1 mais est identique pour les 2 quantités on a

Q1 ↔ Q2 = m

cojecter[Q1, Q2] = m

qui est de nouveau une magnitude, à savoir celle de l'angle entre les exos

Si les exos sont alignées, l'angle vaut 0

Dans le cas où les quantités n'ont pas la même complexité, la comparaison devient une injection

L'injection par la gauche d'une mono q dans une exo Q, notée

$q \rightarrow Q$

donne la partie de Q

"*indifférente*"

par rapport à q

que l'on qualifie de

neutro"

On peut aussi faire une injection par la droite

$q \leftarrow Q$

sachant qu'elle peut toujours se ramener à une injection par la gauche

L'injection permet à la pensée de trouver

l'inverse

d'une mono ou d'une exo

Il ne faut pas confondre

l'inverse

avec

la renversée

qui est elle-même simplement une exo

$Q^{\sim}$

dont les monos q ont été mises dans l'ordre inverse de celui qu'elles ont dans

Q

en notant $^{\sim}$ pour *renverser*

ni avec

l'opposée

qui est le résultat de l'opération

opposition

notée $\nwarrow$

L'inverse d'une mono est

$$q^{-1} = q \nwarrow m^2$$

$$q\text{-}1 = q$$

$$inverser[q] = q \nwarrow m^2$$

Si la complexité est supérieure à 1 et

$$Q^{-1} = Q^{\sim} \nwarrow m^2 = (-1)^{k\,(k-1)}\, Q \nwarrow m^2$$

Le terme $(-1)^{k\,(k-1)}$ sert à calculer la binarité de l'exo

.

$$Q^{-1} = renverser[Q]\ mM^2$$

$$Q \nwarrow Q^{-1} = 1$$

L'injection permet de trouver

la pro-jection

d'une éventualité dans une autre

la pro

Si la pensée considère l'injection d'une mono dans une bi-exo, par exemple, elle trouve la pro-quantité (projection)

$$(q \rightarrow Q) \rightarrow 1 \nwarrow Q$$

$(q \rightarrow Q) \rightarrow Q^{-1}$

injecter[injecter[q,Q], inverser[Q]] = pro-quantité

L'injection permet de trouver directement

la neutro

de la première dans la seconde, c'est à dire la partie indifférente de la seconde à la première

$q \rightarrow Q = neutro$

injecter[p,Q] = neutro

La partie de la quantité *q* différente de la quantité Q est

l'altero-quantité

L'imposition quant à elle est une opération consistant en l'adjonction d'une injection et d'une éjection

imposition = injection + éjection

imposer [q1,q2] = injecter[q1,q2] + éjecter[q1,q2]

$a \searrow b = a \rightarrow b + a \diamond b$

L'imposition est inversible

Outre *l'imposition* il existe donc *l'opposition*

Ces deux opérations d'imposition et d'opposition constituent à elles deux une classe d'opérations que l'on peut nommer

composition