

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

لا حول ولا قوة إلا بالله

اللهم انا مغلوبون فانتصر

اللهم لا سهل إلا ما جعلته سهلا وانت تجعل الحزن «سهل»

Research area¹ : **Algebra**, **Geometry** and **Optimal Transport**
<**KatMerien Theory**>

I. Algebiac Geometry :

- i. **Connection with Symplectic^{-^}/_> \mathbb{C}^n et CP^n**
- ii. **Tropical !! complex\13, Automorphic forms and representations²[NT]];**
- iii. **RT : Lie ; {carquois/amassées}³**

II. Transport optimal

III. **KatMerien Theory**

1 **Mathématique**

1.1 Algebiac Geometry⁴

Mes recherches se situent à l'intersection de la théorie des nombres et de la géométrie algébrique et traitent des questions fondamentales et des structures qui sous-tendent notre compréhension moderne des équations diophantiennes. Cela concerne d'une part le programme Langlands, qui relie des objets de géométrie arithmétique, tels que des courbes elliptiques sur des champs de nombres, à des objets analytiques ou topologiques, tels que des formes modulaires ou la cohomologie des variétés hyperboliques. En revanche, cela concerne les invariants cohomologiques liés aux variétés arithmétiques elles-mêmes, qui font l'objet de la théorie de Hodge, ou plus particulièrement de la théorie de Hodge p-adique.

¹ Je m'intéresse principalement à Transport optimal d'un côté et de la géométrie algébrique, la **théorie Katemrien** et ses relations avec la théorie des nombres, géométrie symplectique, et la théorie de l'homotopie. Je m'intéresse également à la théorie de la représentation, **geometric group theory** et **carquois/amassées**/Representation Theory .

² <https://sites.google.com/site/ngbaymath/home/publications?authuser=0>

³Representations of p-adic groups and automorphic representations

⁴ <https://atricolfi.github.io/Papers.html>

Mes propres contributions à ces questions incluent la construction de représentations de Galois associées par exemple à des classes de torsion sur des variétés 3 hyperboliques et le développement d'un cadre géométrique sous-tendant une grande partie de la théorie de Hodge p -adique, sous la forme d'espaces perfectoides. À l'avenir, j'ai l'intention de poursuivre ces questions. En particulier, en m'appuyant sur des espaces perfectoides, je souhaite transporter des machines du programme géométrique Langlands au cas des champs p -adiques, et j'espère les utiliser pour obtenir de nouveaux résultats sur la conjecture locale de Langlands pour les groupes généraux sur p -adique. des champs. D'autre part, dans les travaux avec Bhatt et Morrow, nous avons trouvé de nouveaux invariants cohomologiques de variétés arithmétiques, qui conduisent à une surprenante déformation q de la cohomologie De Rham, que je compte étudier plus avant.

– theory of algebraic groups, which is a branch of mathematics closely related to algebraic geometry, group theory and Lie theory. More precisely, I study the structure of isotropic reductive algebraic groups and their connection to non-associative algebras (such as Jordan algebras and structuralist algebras) and algebraic K -theory.

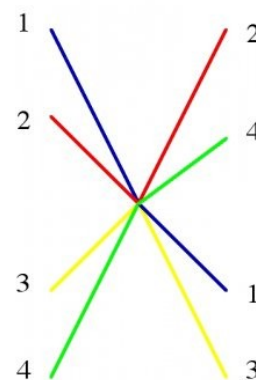
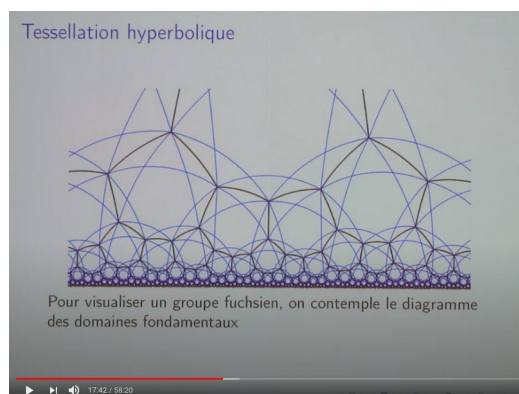
<https://kerodon.net/>

<http://wwwf.imperial.ac.uk/~cshao/geohypde/abstracts.php>

un morphisme étale ? C'est l'équivalent algébrique de morphisme de variétés qui est un difféomorphisme local en tout point.

1.1.1 Théorie de Gromov-Witten des variétés projectives

1.1.2 Une version tropicale de l'observation de Kontsevich⁵⁶ Étienne Ghys Polynôme d'une variable réel $\mathbb{R}[X]$ (Version Tropical)



idée de démonstration ou une réécriture dans un contexte tout à fait nouveau, la figure à droite peut être facilement transmissible dans la figure à gauche (tiré de la conférence donnée par Bao Chau Ngo au collège de France sur La fonctorialité de Langlands et l'équation fonctionnelle des fonctions L automorphes) mais dans le cas de \mathbb{R}

clea) si en prendre l'intersection du polygone principale avec coloration en trouve le théorème des polynômes pour \mathbb{R} ainsi du groupe d'isométries

⁵ <http://perso.ens-lyon.fr/ghys/articles/Promenade.pdf>

⁶ <https://www.youtube.com/watch?v=91OBnOXXqq8>

$$PSL_2(\mathbb{R}) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \mid a, b, c, d, ad - bc = 1 \right\} / \{\pm 1\} \quad (1)$$

1.1.3 Prismatique Cohomologie planaire

Cette idée semble un peu farfelue est fausse, mais quand même. L'idée est la suivante des travaux de Bhargav Bhatt et Scholze, prendre en considération dans leur hypothèse (<http://www.kurims.kyoto-u.ac.jp/~motizuki/papers-english.html>) <https://www.quantamagazine.org/titans-of-mathematics-clash-over-epic-proof-of-abc-conjecture-20180920/>

1.1.4 AG: Riemann Surface

1.1.4.1 Sur l'expression de dérivation de la fonction de Tao le cas ⁷²⁴/ MISC

Algébrisation et tropicalisation de la formule obtenue par Tao^{825 926} en dérivant successivement à n-ième dérivée de la fonction , quel lien avec les surfaces de Riemann?

1.1.5 Théorème de Pick dans le cas Tropical

Rappelons l'énoncé de ce théorème, Pour tout polygone P , i les points intérieurs et b le bord du polygone alors la surface est

$$A = i + \frac{1}{2}b - 1 \quad (2)$$

1.1.6 Tropicalisation de la géométrie de Hilbert¹⁰

Vision plutôt naïve Pourquoi avoir pensé à la géométrie de Hilbert sachant que c'est pas mon domaine de recherche, je rappelle la distance de Hilbert pour la droite

$$d_c(x, y) = \ln() \quad (3)$$



7 <http://lille1tv.univ-lille1.fr/tags/video.aspx?id=344ac96f-4c18-48fe-8450-fe5b405afa89>

8 <https://mathoverflow.net/questions/208341/is-this-differential-identity-known>

9 <https://terrytao.wordpress.com/2015/05/30/a-differentiation-identity/>

10 <https://images.math.cnrs.fr/Geometrie-de-Hilbert.html>

1.1.6.1 L'objectif est de faire une métrique p-adique de Hilbert

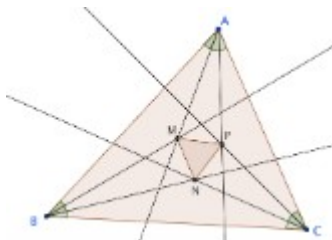
1.1.7 Autour théorème de Morley ¹¹:

Proposer une nouvelle démonstration à base tropicale : nouvelle démonstration; tropicale du théorème; correspondance entre géométrie et combinatoire; passage au dessin d'enfants et extension Galoisienne (publications : <http://annals.math.princeton.edu/>)

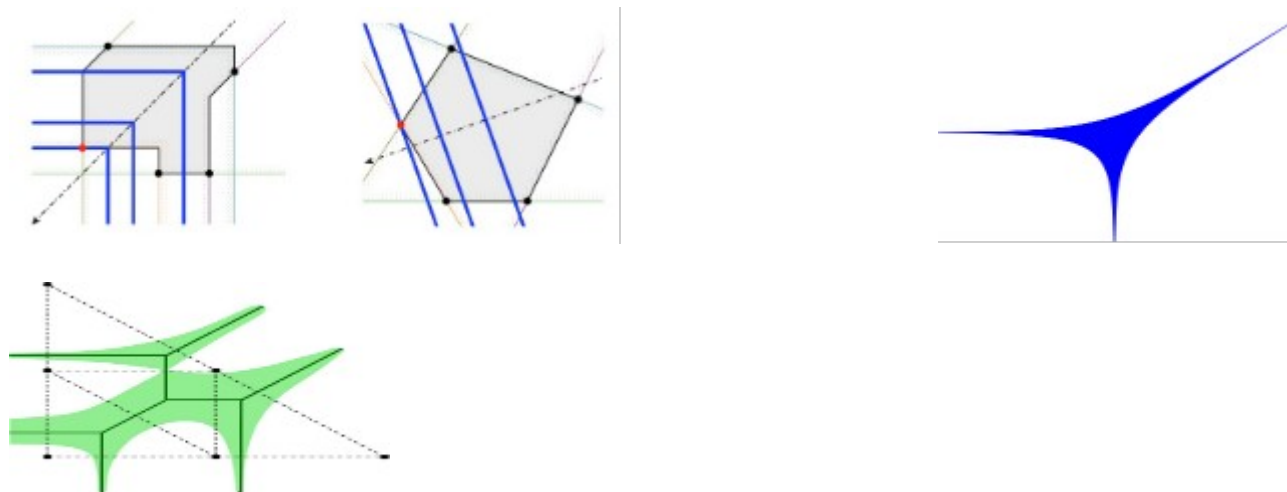
Inventé un théorème de Morley tropical

Inventé un nouvel objet à base de ce triangle

Nous pensons que le théorème de Morley est très intéressant et bon pour la géométrie tropicale pour deux raisons : 1 – une batterie pour le démontré et l'usage de la combinatoire au-quel ce théorème fait appel dans divers recherches, et du coup en pourrait moderniser les démonstrations et notamment celle donnée par Alain Connes¹²²³ à la base de l'isométrie dans \mathbb{C} (nombres complexes), 2 – Une version tropicale de e théorème, qui se base sur l'idée elle-même de la démonstration Amibes basé sur le théorème de Gelfand-Kapranov-Zelevinski.

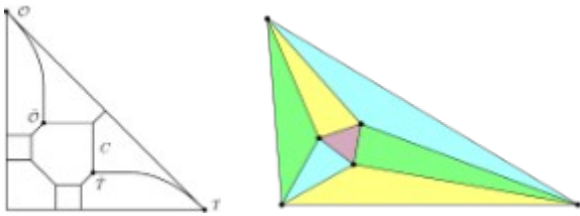


L'observation des deux figures montres, le lien entre les deux formes, qui peuvent être exprimés et démontrés par un procédé tropical. En pourrait même dire que le semi-anneaux tropical démontre l'existence d'un triangle isocèle. L'évolution dans le temps dans la figure à droite du triangle



¹¹ https://fr.wikipedia.org/wiki/Th%C3%A9or%C3%A8me_de_Morley

¹² Alain Connes, « A new proof of Morley's theorem », *Publ. Math. IHES*, S88, , 43-46



généralisation de la fonction, c-a-d au produit de sa démonstration obtenu

1.2 **Transport optimal**¹³ et Soliton

« Je considère la portabilité en génie logiciel comme une théorie de transport optimale d'information ou grain d'information qui se voit un horizon dans le cas quantique, il ne s'agit pas d'application mais bel et bien d'une théorie à part entière, la portabilité en soit un champ de recherche très vaste en génie logiciel qui touche presque à divers domaines, l'objectif général est de porter le portage d'application, de code source ou binaire, de système d'exploitation et toute la nature du problème est informatique plus exactement le génie logiciel, mais trouve une excellente formulation en mathématique, physique mathématique, physique statistique, mécanique quantique et économie »¹⁴

1.2.1 Thermodynamics, and Weil-Petersson metric geometry, Teichmüller metric

Merci à Curtis T. McMullen et pour son article « *Thermodynamics, dimension and the Weil-Petersson metric* » qui m'a mis l'eau à la bouche et le lien avec le transport optimal en effet je cherchais depuis longtemps, un cadre géométrique pour le transport optimal pour la portabilité et sa mesure, un autre cadre consiste à travailler sur l'espace de Teichmüller sublime, donc un nouveau choix, mais pourquoi ça tout simplement ce qui a attiré est c'est le mot thermodynamique !, est bien oui

1.2.2 Soliton, séisme, information et logiciel

1.2.2.1 Preamble

Il y a quelques années j'ai découverte un passage entre la représentation séismique d'un logiciel en l'occurrence en 3D ayant pour objectif la visualisation de bug, évolution surtout dans le cadre de la maintenance¹⁵, à cette époque j'ai pu comprendre et par là déduire que tel phénomène peut être une forme de soliton formuler en EDP, mon objectif est d'écrire une EDP pour la propagation séismique de logiciel, mais comme ce dernier est continu et le logiciel et discret le mariage semble être très difficile, du coup

1.2.1 Introduction

En 1990, Takahashi-Satsuma introduisit un système à soliton discret appelé système de *box and ball* (BBS). C'est une sorte d'automate cellulaire obtenu par l'équation ultra-discrète de Lotka-Volterra [2]. Un état du BBS consiste en une séquence infinie de cases, chaque case

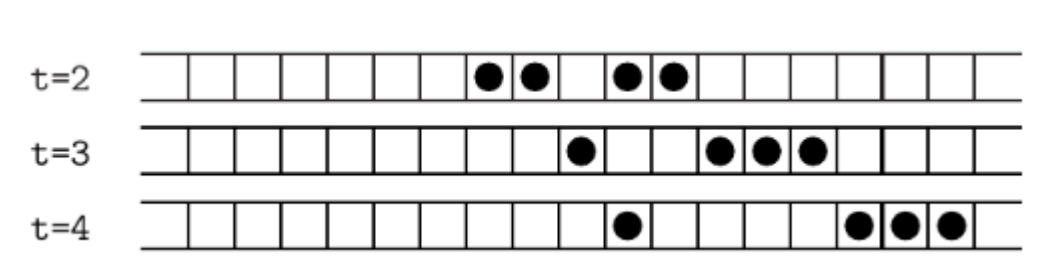
¹³ <https://www.math.purdue.edu/~torresm/geometric-measure-theory.html>

¹⁴En computer science plus spécifiquement en génie logiciel le portage logiciel joue un rôle phare dans divers branches de cette disciplines et des applications majeurs en informatiques, mathématiquement l'idée est intéressante le point de départ s'agit

¹⁵Soliton decomposition of the Box-Ball System

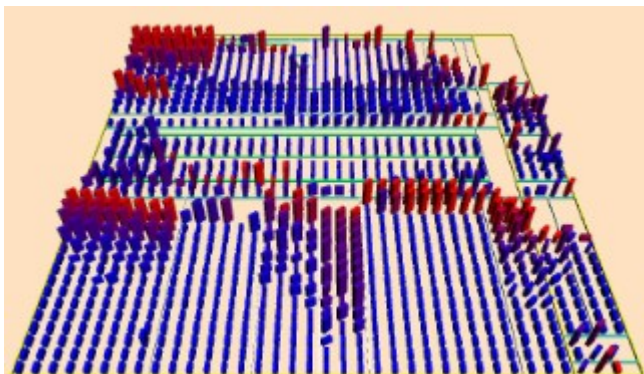
pouvant contenir une balle au maximum. Le BBS a été étudié et généralisé sous divers aspects. ultra discrétisation de l'équation du soliton [9], de la base cristalline [1] [3], de la méthode de diffusion inverse [4], etc.

Le système box-ball est un automate cellulaire intégrable sur un réseau dimensionnel. Il provient de systèmes intégrables quantiques ou classiques par les procédures appelées cristallisations et ultra-discrétisation, respectivement. La double origine de l'intégrabilité a conféré au système box-ball une variété d'aspects. Liées aux modèles intégrables de Yang-Baxter en mécanique statistique, théorie de la base cristalline dans les groupes quantiques, Bethe ansatz combinatoire, cristaux géométriques, théorie classique des solitons, fonctions tau, méthode de diffusion inverse, variables d'angle d'action et tores invariants dans des systèmes entièrement intégrables, spectrale courbes,

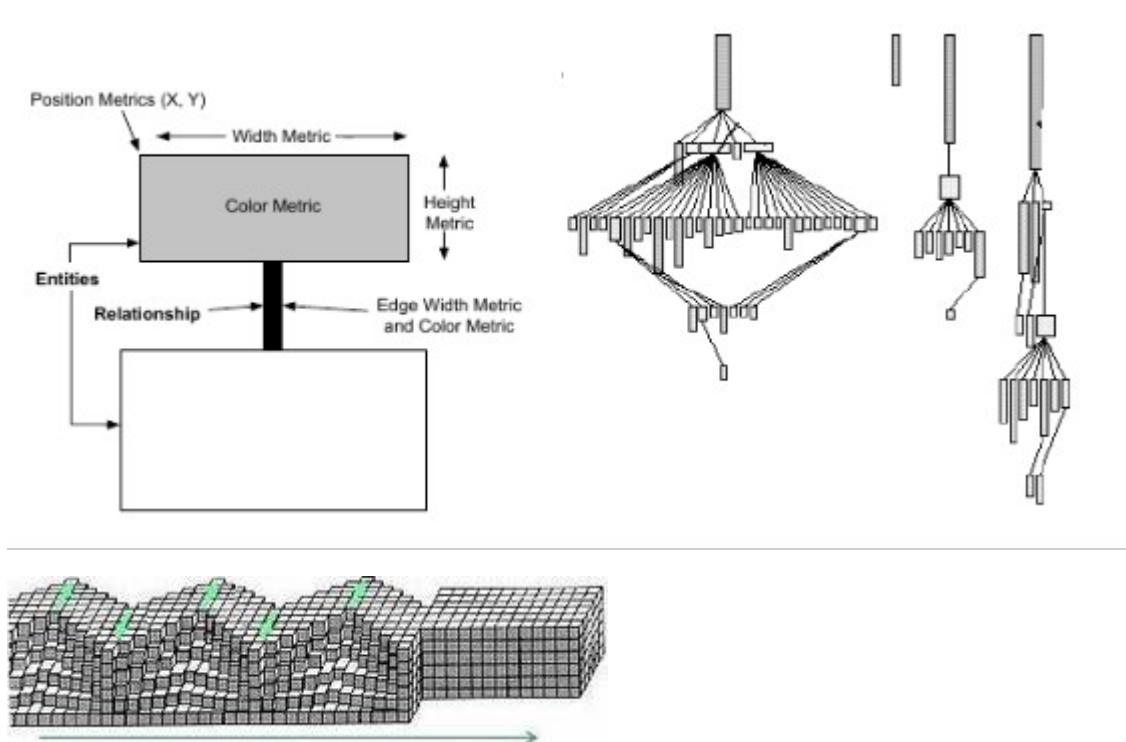


- Mami Okiyoshi, Generating functions of Box and Ball System

1.2.1 Case Study : Software wave



Une modélisation récente entrepris par un groupe de chercheur canadien exploitant l'analogie entre séisme est modification-changements dans le code source du logiciel[Hassine] les méthodes utilisées sont purement informatiques, bien que ces recherches parait intéressant, néanmoins il manque de rigueur est des explications fondé et axiomatique, l'objectif consiste à développé une équation qui interprète ces modifications en suivant l'hypothèse de l'analogie par le séisme



<https://diro.umontreal.ca/english/departement-directory/professors/professor/in/in15076/sg/Houari%20Sahraoui/>

<http://musee-sismologie.unistra.fr/comprendre-les-seismes/notions-pour-petits-et-grands/notions-de-base/ondes-sismiques/>

1.2.2 Broken windows phenoma^{16 17 18}

Le thème très intéressant du « **Broken windows phenoma** » une petite dégradation dans une fenêtre la dégradation se propage dans tout le système ; modèle est analyse éliminer le mauvais code, le risque de la propagation en pourrait faire un lien avec la visualisation logicielle, dans un contexte purement logiciel avec l'analyse et l'étude d'empirisme, et d'un autre côté en peut analyser en utilisant notre approche purement mathématique, c-a-d en employant une EDP qui modélise les séismes¹⁹ comment une idée pour la cinétique de la dette technique.

¹⁶ https://en.wikipedia.org/wiki/Broken_windows_theory

¹⁷ https://fr.wikipedia.org/wiki/Hypoth%C3%A8se_de_la_vitre_bris%C3%A9e

¹⁸ <http://patrickmorvan.over-blog.com/article-la-theorie-de-la-fenetre-brisee-broken-window-theory-64718475.html>

¹⁹ <http://ieeexplore.ieee.org/document/6080772/> (A seismology-inspired approach to study change propagation)

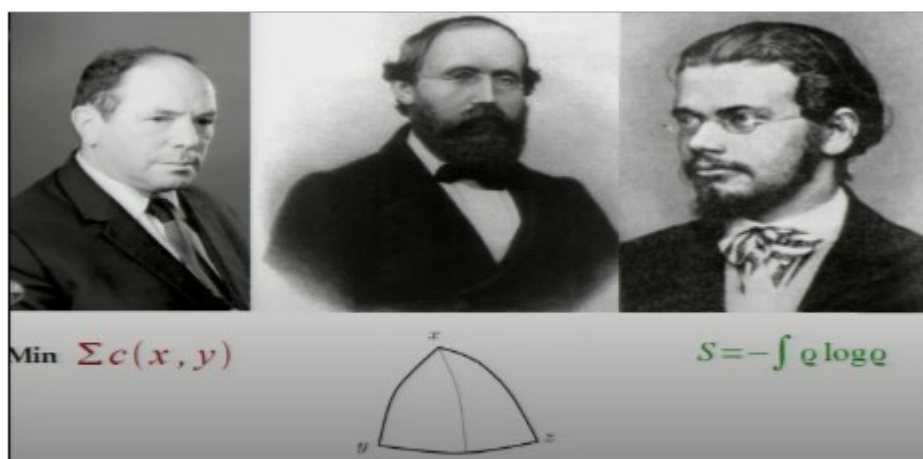


1.2.3 Portabilité

La portabilité entre logiciel ou système d'exploitation et entre logiciel et système jouent un rôle fondamental et très important en informatique dans divers domaine de recherche et surtout industrielle voir de l'entreprise

1.2.4 **Transport optimal²⁰ et technical debt**

Le travail de Cedric Villani est fondamentale en effet par l'axiome que la dette technique est une entropie initiale, il existe une sorte de correspondance entre. *Pourquoi affirmer qu'il y a une analogie entre gaz et dette technique ?* Pour répondre à cette question, je me suis appuyée sur le fait qu'un ordinateur est un gaz qu'un logiciel aussi une mesure de densité à l'intérieur de cette grande boîte de gaz. Je vais parler d'une dette technique paresseuse en exploitant l'analogie avec le gaz d'après Villani et d'autres auteurs, mon objectif et de faire une approche géométrique et topologique de la dette technique, liée au transport optimal, mais dit-ton est ce que le transport optimal à une relation avec la dette technique . Mais avant toute chose en doit supposer que $debt_0 = 0$

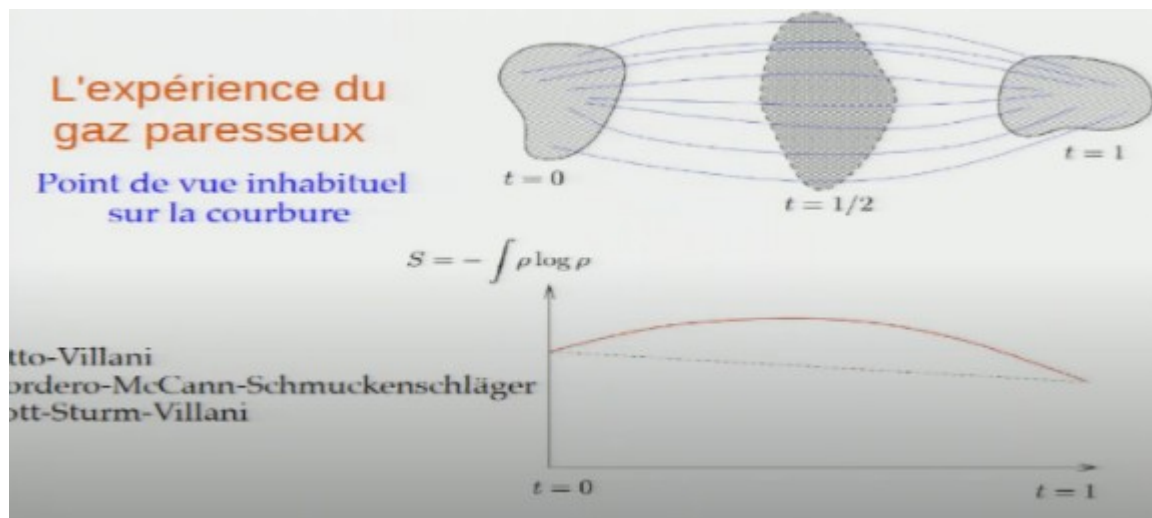


§

Figure13

²⁰<https://www.futura-sciences.com/sciences/dossiers/mathematiques-maths-transport-moindre-cout-849/>

Maintenant et d'après la figure l'idée et de relier les trois images donc trois formules, avec une petite précision à savoir changer la formule de l'entropie de Boltzmann. Une question de partition²¹ pour le logiciel ou l'information en entrée ?



1.2.4.1 Quelques observations

Mark Burgess dans son livre sur la théorie administration système, évoque la symétrie des configurations, cette dernière est applicable pour la portabilité et du coup pour le transport optimal, donc le transport optimal possède une propriété symétrique illustré par l'exemple : Si une relabélise chaque fichier du système, dans toutes les références, elle continuera à fonctionner comme auparavant.

1.2.5 Hypothèse de portage

Théorème fondamental : le coût du portage est t-il invariant ?

Comme une théorie de transport optimal (formulation)

Il est important de constater que les articles qui touchent à la durée de vie du logiciel peuvent être vue avec façon différente, la première sur le coût de développement d'un point de vue purement économique développé dans un papier ancien de Michel Volle²²³⁵ n suivant le même raisonnement que volle mais avec une hypothèse d'une variable aléatoire dans un milieu mul-

²¹**Version mathématique de la dette technique:** La métaphore de la porte et la fenêtre dans une maison est leur dépendance une réalité qui peut être interprété mathématiquement. Durant les conditions initiales la dette technique est toujours existante le démarrage, dans un projet libre ou open source à partir qu'un autre développeur contribue la dette technique est en mouvement elle se déplace durant le processus de développement qui plaque aussi le modèle économique liée au open source, donc la dette technique géré par exemple. Un résultat fondamentale est celui de dire que la dette technique à une **propriété de transport** ou de **conductivité**, en doit penser que chaque partie de la dette technique à t-elle une mesure d'entropie cela rappelle la théorie de Bowen entropie sofique ?

²²<http://www.volle.com/rapports/duree.htm>

ti-choc, en prend en considération de rechercher des analogies. $c_0 = kf(n_0)$ avec c_0 le coût initial, n les lignes d'un logiciel, je prends pas en considération ça ramification reporté sur un graphe, mais plutôt à l'état brut, assimilé à une PME/PMI !!; des questions : La durée de vie d'une dette technique ?, comprendre analytiquement la pérennité. Est ce que la pérennité et le pire ennemi de la durée de vie !. Si je reprends $c_0 = kf(n_0)$ dans le contexte d'un programme ou bien l'hypothèse de celle de Hassen en considère que c_0 le coût qui sort de l'usine donc la valeur, si on suppose que la valeur quand suppose positif du coup k (constante de Il y a ici une grande problématique à savoir la nuance qui existe entre théorie de l'information et thermodynamique si on part du principe du fait que ces deux choses sont complémentaires, car là je bloque totalement et je mélange beaucoup de chose, mais si je veux calculer, la durée de vie d'un logiciel ou d'un composant avec quel aspect je peux le faire dans quelle contexte, si conclus qu'il y a trois sorte d'approche.

Une théorie profonde de transport optimal pour la portabilité finie et infini, la relation avec l'entropie se fera d'une manière très naturelle

1.3 Symplectic Geometry

1.3.1 Lecteur de l'article « The symplectic egg in classical and quantum mechanics »

1.3.2 On s'intéresse quelques aspects des surfaces de Kummer dans le cas symplectique

1.3.3 "Certains essayent aussi de mieux comprendre les « espaces de modules », c'est-à-dire les espaces décrivant toutes les formes possibles d'objets d'un type géométrique donné, comme les courbes algébriques"

1.3.4 Groupe de Burgess

Definition 68. A group \mathcal{G} of transformations is a symmetry of the high level configuration $q(\vec{x}, t)$, if for some x and time t , the transformation of the configuration domain

$$q(\vec{x}, t) = q(g(x), t), \quad (16.6)$$

is an identity, and $g \in \mathcal{G}$.

1.3.4.1 Problème de la politique de configuration

Dans son article Mark Burgess, introduit la formule suivante :

Cette formule est extraordinaire et magique elle indique que la politique de configuration est la division de la configuration $C(t)$ sur le produit tensoriel du groupe de Burgess est Pour l'instant cette expression regroupe trois chose: théorie des groupes, théorie de la représentation, la **géométrie symplectique** et **algébrique**

- 1.3.4.2 Question d'action
- 1.3.4.3 Quel lien avec le groupe symplectique
- 1.3.4.4 Quel lien avec le groupe de Lie G ?

1.4 Retour aux tresses, Catégorie, Galois, Géométrie de groupe, c^* -patches

Point de départ : Théorie d'administration système

1.4.1 Braided semigroup diagram

Dans cette section, notre objectif est de motiver le formalisme des diagrammes de **semi-groupe tressés** étudiés dans les sections suivantes en présentant les groupes de Thompson F , T et V d'informations sur ces groupes peuvent être trouvées dans. Nous commençons par Thompson's groupe²³

1.5 Katmerien Theory²⁴

La jeunesse de la théorie. Ma première réflexion sur le sujet il y a de cela 12 ans plus exactement en 2003-2004 juste après avoir obtenu mon diplôme est lu beaucoup de livres de système d'exploitation de se demander s'il en peut écrire une équation dynamique des processus qui s'exécutent dans la mémoire de l'ordinateur, c'est le commencement de mes recherches et réflexions sur la théorie Katmerienne, à cette époque je m'intéressais guère à faire le pont avec la physique, ayant aucune connaissance de l'entropie, de l'espace-temps dans sa forme actuelle ni des recherches académiques sur le logiciel libre et l'open source, j'ai eu ma première réponse dans un livre de Sacha Krakowiac sur les systèmes d'exploitations,

La question du temps dans tout ça.

Le lien avec l'entropie.

D'autres approches. Je viens d'avoir connaissances de publications de deux physiciens de l'université de Michigan Gabriele Carcassi et Christine A. Aidala²⁵, sur un thème qui me préoccupe depuis un beau moment à savoir la conservation de l'information en utilisant un Hamiltonien \hat{H} je fais référence ici à leur article « *Hamiltonian mechanics is conservation of information entropy* » ou il montre l'équivalence entre la mécanique hamiltonienne et la conservation de l'entropie de l'information avec la condition que l'information doit être conservée durant l'évolution du système une question intéressante est de rechercher un invariant ici ? Mon idée est de prendre **Donaldson-Thomas invariants**²⁶ comme hypothèse puis le déformé

1.5.1 Mouvement Brownien et intégrale de chemin

Pour bien voir les choses, il est clair que la relation entre Mouvement Brownien et intégrale de chemin est bien établie donne une idée sur la représentation évolutive du logiciel qui esas-

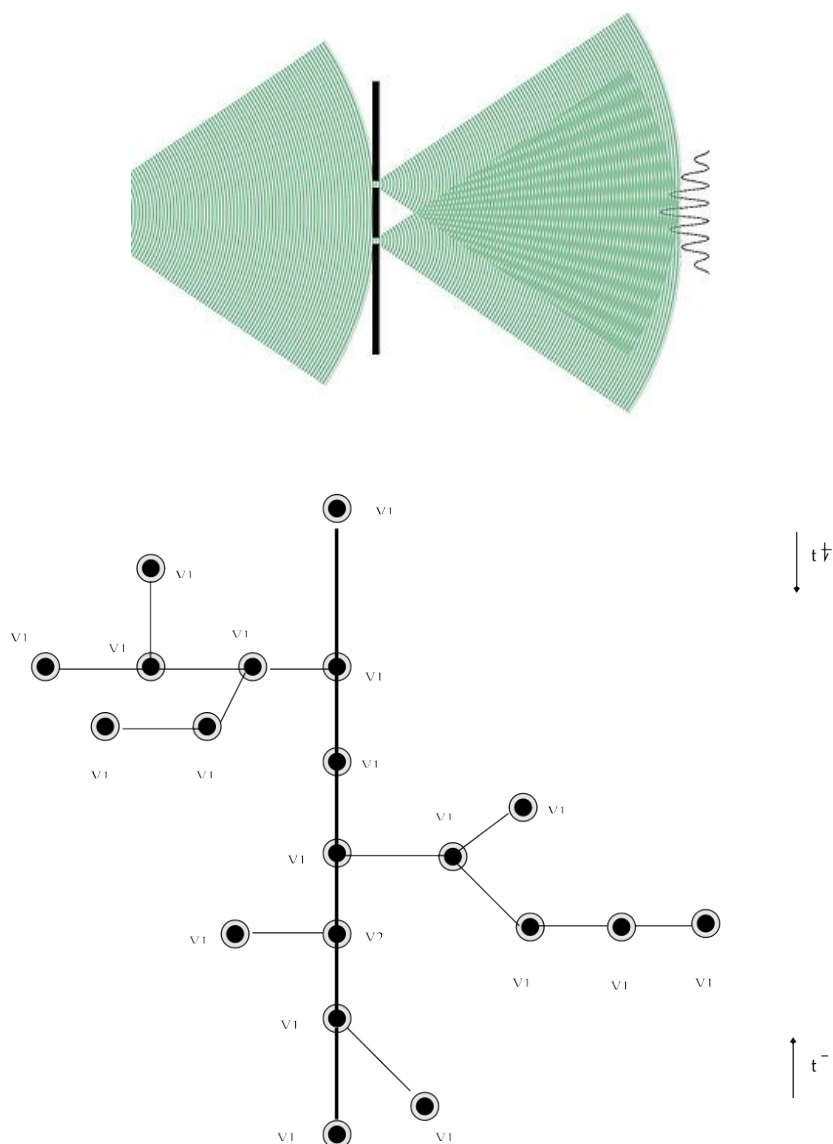
²³ <https://arxiv.org/pdf/1709.03888.pdf>

²⁴ **L'histoire des sciences, épistémologie et originalité scientifique.** Je pense que ce sujet n'a pas été totalement traité dans la sphère pensante, ou bien à effet quelque apparition dans des écrits, bribes et insinuation telle le traitement de l'esthétique chez Poincaré, ou les remarques de Hadamard, Wiles, Muford et tant d'autres de mathématiciens allant au grand savant musulman **Ibn Khaldoun**, **Ibn Taymiyya**, ce que j'ai envie de développer ici c'est le fait de jugé un travail original, comment ? Sur quelle base ?, est t-il soumis aux lois du marché et critère subjectif de jugement collective sur une évaluation personnelle de mathématicien est bien d'autres aspects, prenons l'exemple de Grothendiek, Arnold et Perelman pour ouvrir le débat

²⁵ <http://assumptionsofphysics.org/>

²⁶ <https://www.ihes.fr/~maxim/TEXTS/DTinv-AT2007.pdf>

sez semblable à un intégral de chemin on a une source est en somme sur tous les chemins en terme dynamique, pour rappel l'évolution du logiciel ressemble à l'expérience de Fentes de Young, d'un côté de l'autre une ressemblance avec les graphes de Feynman.



Les chemins de la source à l'écran sont les multiples méthodes ou algorithmes pour écrire le logiciels, donc si en somme tout les méthodes ou algorithmes qui nous décrit. En remarque et j'insiste longtemps sur ce point que l'intégrale de chemin existe belle et bien en informatique et notamment en génie logiciel, est ce que on pourrait exprimer l'évolution du logiciel par une intégrale de chemin, sa reste quelques choses à vérifier²⁷. Maintenant le programme est beaucoup plus clair, on suppose que on a une source d'information au sens de la théorie d'information de la théorie de Shannon, c-a-d au lieu de parler d'électron on parlera d'information logiciel

²⁷ Je suis parvenu à ce résultat il y a de ce la 10 ans, mais son donnée un réel développement, ni une continuité mis a part un survol ou une progression vers une algébrisation de ces intégrales, vue le caractères discret de l'informatique et du logiciel

1.5.2 Modèle JANUS et la mécanique héréditaire

Équation héréditaire obtenu par Francis Fer dans son livre L'irréversibilité

$$\frac{d\zeta^i}{dt} = X^i(\zeta^k, t) + \int_{-\infty}^t K^i(t; \zeta^k(\theta), \theta) d\theta \quad (4)$$

1.5.3 Théorème de Noether invariance et conservation énergie-information

Y a-t-il conservation de l'information ?, elle se manifeste dans beaucoup d'occasion. Mon hypothèse en ce qui concerne cette théorie est la suivante : l'unité de mesure est l'entropie donc en a affaire à un système dynamique, bien que l'entropie elle-même ne suffit pas. Du coup je cherche des relations avec **KMS-structure** !. de comprendre oui ou non, le théorème de Noether est applicable dans le cas ou on a la fois irréversibilité et réversibilité de l'énergie-Information-matière-espace_temps La réponse suite à la question d'une approche de la théorie de Noether dans le cadre de l'entropie ne semble pas être convaincante pour manque de variable qui conserve l'état ou la structure de la matière. Allant plus loin et si ce théorème se collait très bien avec la complexité de Kolmogorov et plus particulièrement la complexité organisée et le calcul de Bennet, énergie est tout ce que nous devrions voir comme moment angulaire, quantité de mouvement seront stockés dans l'objet lui-même, d'insuffisance de la pensée physique actuelle de comprendre le monde sans pour autant s'appuyer sur la confusion et la complexité du monde qui nous entoure.

1.5.4 Verlinde ideas/ Axiomes de Wightman

Est-ce qu'on peut passer d'un opérateur à un autre au sens physique point de départ YM? (Hypothèse : Spectral sur un C*-Algebra) d'un point de vue purement physique

Quels sont les types de problèmes posés par l'étude de YM?

Une théorie de *mécanique symplectique complexe* d'après Jean-Pierre Petit. Quel serait l'entropie de densité dans ce cas, toujours n'en est loin de la **Katemi theory**

Pourquoi pas la pression ?. Voilà une idée qui me titille depuis plus de six mois, comme je suis convaincu que la gravité n'a plus de sens notamment dans l'espace, je me demande s'il est intéressant de penser en termes de densité, pression, entropie et électromagnétique, j'ajouterais que l'unification des lois de la physique c-à-d l'exprimer on une seule équation ou comme ils veulent la surnommer par la formule de Dieu ! est une quête éphémère et perte de temps pour arriver à leur fin les physiciens veulent coûte que coûte vouloir torsader et obliger la nature à se soumettre à un processus de névrosé et une sorte d'état psychédélique intense jusqu'à sortir même du cadre simpliste des modèles la gravité est une faute grave dans la pensée et l'hypothèse, allons un peu loin si on suppose de facto l'existence de la gravité g , rappelons tout de même que cette constante dite universelle est une hypothèse posée par Newton dans son « Principia » justifiant l'attraction et la chute des corps, mouvement des particules, quand trouvera dans la mécanique des fluides, etc., je vais sauter la ligne est énoncé la loi de l'énergie potentielle $E = mgh$ pour m la masse, h la hauteur et g^{28} la gravitation

²⁸En va supposer que g est tellement faible que l'on passe à 0 ou nulle ou l'énergie potentielle devient

du milieu qui est une constante, donc j'ai deux inconnus la masse et la hauteur du coup je me retrouve avec une fonction à deux variables que j'exprime par le gradient \overrightarrow{grad} si je regarde les choses d'un point de vue plus abstrait j'ai le droit de jouer avec la constante g est comme c'est une hypothèse nous allons utiliser le symbole ρ . Autre remarque important, en va supposer que le début de toute chose est une **l'information** et en va écarte toute discussion concernant sa nature physique pour le moment est se focaliser sur le commencement, le premier exemple est celui du fœtu humain. A noté que l'information est une dérivation de la Katemrienne, c'est une entité qui nous y parvenu et dont en sont servis, nous pensons que le vrai point de départ est dans notre vraie compréhension **الاسماء و الصفات** ce point doit être pris au sérieux est très bien détaillé, car il recèle en lui d'énorme surprise et de révélation sur la nature même de la physique, nous parlons et Dieu parle, mais sa façon de parler nous y somme très différente est s'identifie avec sa majesté, telle sera nos origines de réflexions.

<https://quantumgravityresearch.org/about-quantum>

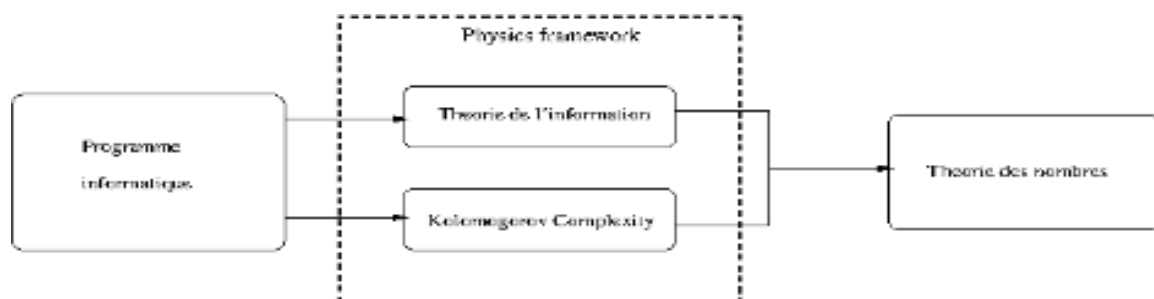
Un livre à découvrir :

https://www.academia.edu/31316572/Mathematical_aspects_of_quantum_field_theory?email_work_card=thumbnail

1.5.5 Un problème de théorie des nombres²⁹

La théorie des nombres est une branche très ancienne des

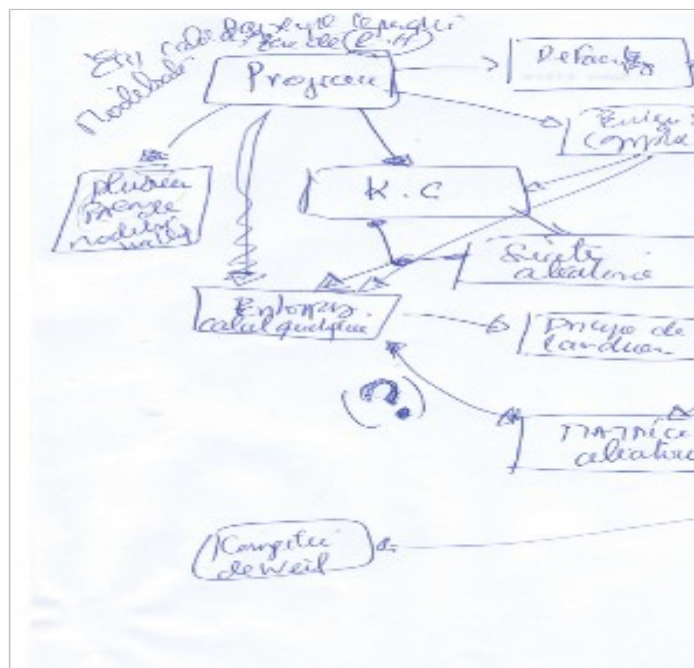
1.5.5.1 Introduction



1.5.5.2 La question principale

La question principale est de comprendre le lien entre un logiciel/programme et les nombres ?. Y a-t-il une relation entre le principe de Landauer et la théorie des nombres ?. Quel type de relation nous espérons trouver entre un programme informatique et les nombres entiers, rationnels ou algébriques ?. Pourquoi pas cette relation conduit t-elle à la fonction zêta de Riemann est la répartition des nombres premier avec les programmes au sens du génie logiciel ou bien avec les nombres entiers ?. **Ce rapprochement peut-t-il nous aider à démontrer la conjecture de Riemann ?. Ou autre conjecture ou problème.**

Une question de modélisation oui ou non?. Un programme informatique peut être modélisé ou représenter de différente manière il y a ceux qui existaient auparavant et ceux auxquelles je suis arrivée à les mettre en oeuvre pour résumer un programme est : espace vectoriel, un système dynamique discret dans ou **système dynamique** tout court avec une **théorie ergodique** associé, un **C*-Algebra** qui pourrait déboucher sur une géométrie non commutative, un **graphe** orienté associé à une variété Laplacienne, homotopie stable, **espace métrique ou ultra-métrique**, espace de Banach non linéaire allant à l'**analyse spectral**, une autre piste algébrique consiste à recherche des constructions de la théorie des catégories qui à fait ces preuves en génie logiciel, et **corps fini** algébrique , la complexité de Kolmogorov est une méthode intéressant pour mesurer la complexité d'un programme dans le cas numérique ou autre chose reposant sur des être plus fondamentale qui est la machine de Turing , cette complexité algorithmique nous informe sur beaucoup de chose intéressante exprimer en termes de suite aléatoire qui à la fois est un handicapé et une bonne solution.



1.5.5.3

Mis à jours de l'intérêt et objectif.

Arithmetic dynamic, partition function, Ramanujan, dynamical system, entropy, Kolmogorov Complexity, programming, Cohomology, Principe de Landauer, random matrix, finite Field, suite aléatoire

1.5.5.4 L'objectif de travail

Répondre à la première question ?. La première réponse à la question si un programme quelconque X est un nombre entier ou peut être vue comme un nombre ? est sans aucun doute une question fondamentale un programme se ramène à une quantité d'information structuré en ligne de code, donc en parle de taille et de nombre fini d'élément, par exemple combien de ligne de code écrire dans un langage donné pour obtenir la valeur de $\zeta\left(\frac{1}{2}\right)$?

$$\zeta\left(\frac{1}{2}\right) = -1.4603545088095868128894991525152980124672293310126....$$

$\zeta\left(\frac{1}{2}\right)$ est-t-il aléatoire ? Quel est le programme le plus court qui générera $\zeta\left(\frac{1}{2}\right)$? ce type de question est purement de l'ordre de l'informatique théorique il nous s'intéresse pas beaucoup.

$$\zeta(-2) = \zeta(-4) = \zeta(-6) = \dots = 0 \quad (5)$$

La conférence³⁰ de Gérald Tenenbaum. L'idée est se libérer des programmes comme état physique et j'en parle pas d'algorithme. En plus si en remarque très bien, en a trouvé jusqu'au présent beaucoup de lien entre mathématique, physique et informatique entre mathématique et informatique en a une transcription du génie logiciel de la topologie, géométrie, mais je n'est pas trouvais un travail qui fait le lien

Un programme informatique s'annule t-il ? : cette question à telle un sens ?

Rappel : Avant de répondre à cette question, en cherche à la fonction au polynôme qui s'annule, l'approche du programme connu qui est issue de l'informatique théorique peu ne peut pas nous intéresser, je vais re poser la question, comment peut-on écrire notre programme il est possible, c'est clair et à chaque fois ; en ajoute une ligne et diminue, donc on obtient une fonction qui apparaît en théorie de la complexité comme suite du nombre de ligne de code en suppose que cette fonction est la fonction **d'une partition d'entier** c'est là le point angulaire très très importante.

Remarque. Pourquoi les digrammes de Young³¹⁴⁰ sont très proches de la partition³² du code source et le calcul de nombre de ligne dans un logiciel, nous allons le montrais en dessous.

A la recherche du plus fondamental ?

Le rôle décisif de la physique : avant de le préciser j'ai posé la question suivants y a t-il une relation entre le principe de Landauer et l'hypothèse de Riemann ? Cette question surgit d'un délire total. Cependant avant de commencer il existe de sorte d'apparition de nombre dans la physique celle quand qualifie de naturelle et l'autre proposable c-a-d écrire une version à titre d'exemple

Et ceux ou la manifestation d'une théorie physique dans un théorème de théorie des nombres apparaissent d'une façon naturelle et surprenante, par exemple les matrices aléatoires et la fonction

³⁰<https://vimeo.com/208448534> (a partir de 50:00)

³¹<http://villemin.gerard.free.fr/Referenc/Prof/APROF/Partitio.htm>

³²<http://villemin.gerard.free.fr/Referenc/Prof/APROF/Partiti1.htm>

de Riemann, notre travail va dans ce sens qui permet de prouver d'un côté que la physique liée à l'informatique subsiste dans beaucoup de théorie des nombres. Et comment les programmes informatiques se réduisent aux opérations algébriques des nombres sur certains corps algébriques et structure arithmétiques.

Hypothèse: Les zéros de la fonction $\zeta(s)$ de la droite $\text{Re}(z)$ correspondante à l'énergie minimale qui permet la manipulation d'un bit d'information décrit par le principe de Landauer ?

1.5.5.5 Chercher n ?^{33 34}

L'approche classique consiste de partir de la combinatoire des graphes dire qu'une représentation d'un programme est un graphe, ou bien poser la question combien en peut partitionner un programme, enfin de compte en veut obtenir la valeur n égale au différent façon de la partitionner, d'une manière analogue à au partitionnement d'un nombre entier, mais en veut aussi faire apparaître le fait que c'est un programme. Sachant aussi que X peut se représenter de divers manières. Il y a une autre chemin beaucoup plus simple pour partitionner un programme, en amont le partitionnement c'est une suite convergente donc un nombre fini, et des arrangements, combien de manière en fait pour distribuer le code c-a-d les nombres de ligne et leur complexité !.

1.5.5.6 Lien avec le programme de Langlands

<https://mathoverflow.net/questions/119329/the-riemann-hypothesis-and-the-langlands-program>

<https://math.stackexchange.com/questions/71113/consequences-of-the-langlands-program>

1.5.1.1 Clin d'œil

1.5.1.1.1 Thermodynamical formalism and Markoff-Hurwitz equations d'après le travail de Michael Magee

En partant de la simple question «quand est la somme des carrés d'un tuple d'intégrales qual à un multiple de leur produit?», On arrive à une famille d'équations diophantiennes appelées équations de Markoff-Hurwitz. Je vais expliquer comment le problème du comptage de solutions entières à ces équations peut être étudié à l'aide d'opérateurs de transfert et du formalisme thermodynamique. Cela conduit à de nouvelles connexions entre la géométrie diophantienne et certaines fractales, y compris le `` joint Rauzy ": une fractale qui apparaît dans des domaines mathématiques disparates, y compris les surfaces triplement périodiques, la dynamique des cartes sur le cercle, les généralisations dimensionnelles supérieures des fractions continues, la théorie de Teichmüller, et maintenant , en diophantine geometry. Il s'agit d'un travail conjoint avec Alex Gamburd et Ryan Ronan

<https://www.youtube.com/watch?v=DDLe1AqBM-Y>

1.5.1.2 Relation avec la théorie additive des nombres

Une note d'application et de lien qui fait mis en action la théorie additive de nombre. On lit dans Wikipédia « **La théorie additive des nombres** est une branche de la **théorie des nombres** où sont étudiées des parties de l'ensemble des entiers, et leur comportement vis-à-vis de l'addition. Plus

³³Pourquoi chercher ?,

³⁴<http://villemine.gerard.free.fr/Wwwgvm/Addition/PttIntro.htm>

abstraitement, ce domaine inclut l'étude des groupes abéliens et des demi-groupes commutatifs, dont la loi interne est alors notée additivement. Il a des liens étroits avec la combinatoire arithmétique et la géométrie des nombres. Le principal objet d'étude est la somme d'ensembles : somme de deux parties A et B d'un groupe abélien et somme itérée d'une partie A avec elle-même. »

Par exemple la densité de Schnirelmann où A est le nombre de lignes de programme qui est un nombre entier, avec n un nombre entier, $A(n)$ le nombre d'éléments de $A \cap [1, n]$ ou encore $A(n) = \text{Card}(\{k \leq n \mid k \in A\})$ la densité de Schnirelmann³⁵

$$\sigma(A) = \inf \left\{ \frac{A(n)}{n} \mid n \in \mathbb{N}^* \right\} \quad (6)$$

1.5.1.3 Relation avec entre système dynamique et théorie analytique des nombres^{36 37}

Une observation très pertinente consiste à remarquer que à prouver l'aspect de cette théorie avec Christopher Deninger et d'autres, si en prend un programme et sa partition en remarque que cela s'écrit très naturellement dans ce contexte, Arithmetic dynamics^{38 39}

1.5.1.4 De l'intérêt de la fonction Zêta ou ployzeta

1.5.1.5 Besoin d'un polynôme comment le faire sortir ?

1.5.1.6 Chemin plus court pour trouver $K_U(s)$ ⁴⁰

$K_U(s)$ Existe en physique, c'est un concept d'information donc en peut le relier avec la théorie ds nombres car, la physique possède beaucoup de connexion avec la théorie des nombres, de simple déduction en apparence. En affirme que $K_U(s)$ à une bijection avec un programme, en pose $|M| = n$ alors en a que : $p(K_M(s)) \rightarrow p(n)$. Mais ce raisonnement nécessite une chaîne très astucieuse

1.5.1.7 Les idées de Carlos Castro Perleman

Le chercheur de QGR Carlos Castro Perelman parle d'une approche pour résoudre l'hypothèse de Riemann qui est revisitée dans le cadre des propriétés spéciales des fonctions et de la notion d'invariance $\$ \{ \backslash \text{cal } C \} \{ \backslash \text{cal } T \} \$$. L'opération de conjugaison $\$ \{ \backslash \text{cal } C \} \$$ équivaut à des transformations d'échelle complexes, et l'opération $\$ \{ \backslash \text{cal } T \} \$$ revient à

³⁵https://fr.wikipedia.org/wiki/Densit%C3%A9_de_Schnirelmann

³⁶<http://villemin.gerard.free.fr/Wwwgymm/Suite/FoncPart.htm>

³⁷<https://etzetas2018.sciencesconf.org/>

³⁸https://en.wikipedia.org/wiki/Arithmetic_dynamics

³⁹<http://www.math.brown.edu/~jhs/ADSHome.html>

⁴⁰<http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/kolmogorov.html>

l'inversion Un opérateur de type échelle judicieux est construit dont le spectre est à valeur réelle, conduisant à et / ou $s \in \mathbb{R}$. Ces valeurs sont l'emplacement des zéros non triviaux et triviaux, respectivement. Une analyse approfondie de la correspondance biunivoque entre les zêta zéros et des conditions d'orthogonalité entre paires de fonctions propres révèle que les zéros existent en dehors de la ligne critique. Le rôle des transformations $\{ \text{cal C} \}, \{ \text{cal T} \}$ et les propriétés de la transformation de Mellin des fonctions ont été essentiels dans notre construction.

<https://www.youtube.com/watch?v=ipeYMsIIbJk>

Cela nous amène à prendre en considération la généralisation de Riemann-Polya

1.5.1.8 Lien avec P=NP

1.5.1.9 Quels autres liens avec l'hypothèse de Riemann

Avant de partir sur la généralisation de Riemann-Polya, nous rappelons qu'il y a divers chemin à prendre pour en arriver à la généralisation et surtout très important de souligner qu'il faut trouver une relation entre répartition des nombres, **système dynamique discret** dans \mathbb{R}^n ou \mathbb{C}^n (figure-2) et hypothèse de Riemann, $K_M(x)$ peut être identifié avec les groupes moyennables(c'est extraordinaire) non !; Alors quelques questions : Par exemple

interprétation abstraite

Un programme est un *système dynamique discret* dans \mathbb{R}^n : on cherche des ensembles englobant les valeurs atteignables, à chaque ligne, pour toutes les exécutions possibles.
Et on crée un immense système dynamique ensembliste (une équation par ligne)

Pour le calculateur primaire de vol de l'A380 : 1 million de lignes et des dizaines de milliers de variables!

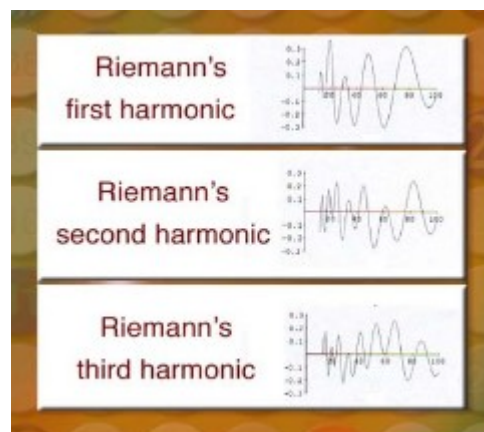


1.5.1.10 Quels autres liens avec l'hypothèse de Riemann

- Y a-t-il une relation entre la fonction de partition $p(n)$ et l'hypothèse de Riemann ou celle de la généralisation de Riemann-Polya, on lit dans Wikipédia le commentaire suivant « **Une autre piste part d'étranges analogies entre la répartition empirique des zéros connus et le spectre de certains opérateurs ; là encore, il n'a pas été possible d'en tirer même un plan d'attaque.** »

- Relation entre la fonction Zêta et la mécanique statistique, une très belle entrée pour l'utilisation d'entropie qui correspond à l'évolution des lignes de codes.
- Comment exploiter la relation entre un système dynamique discret et la complexité de Kolmogorov?, une réécriture permet de passer à une entropie ou un groupe donc les questions sont trop couplées si empreinte cette expression. Donc il faut bien choisir les points à traiter.
- Il faut se mettre à l'évidence que la découverte d'un résultat avec l'hypothèse de Riemann bien que sur le plan hypothétique est fort probable cela induit pas sa démonstration, mais plutôt une piste et l'ouverture vers d'autre horizon en mathématique
- On choisit un autre corps pour le système dynamique discret⁴¹⁵⁰ ? je pense à \mathbb{Q}/\mathbb{Q}_p ou \mathbb{F}_p corps fini ou une variété ou tout simplement à \mathbb{C}^n

1.5.1.11 Le travail de Terence Tao⁴²



1.5.1.11.1 S'appuyant sur des résultats partiels obtenus par le cinquième projet Polymath, Tao annonce en septembre 2015 une preuve de la conjecture de discrepancy d'Erdős utilisant en particulier pour la première fois des calculs d'entropie en théorie analytique des nombres

1.5.1.11.2 $p(n)$ Et $K_M(x)$ ⁴³⁴⁴

Pour commencer on peut affirmer les choses suivantes:

Partition d'un nombre entier, partition d'un graphe...

⁴¹https://www.canal-u.tv/video/inria/numbers_computers_and_dynamical_systems.33667

⁴²<http://villeminegerard.free.fr/aMaths/ThNb/Erdos.htm>

⁴³https://golem.ph.utexas.edu/category/2017/12/entropy_modulo_a_prime.html

⁴⁴ <http://images.math.cnrs.fr/Dynamique-diophantienne.html>

d'existence majeure et non d'unicité, une préoccupation de la communalité du logiciel libre depuis le début, et des langages de programmation, c'est cette problématique qui à permis à inventer des styles, techniques et technologies de programmations et demeure encore une nécessité en utilisant des approches à la fois issues de l'informatique ou bien d'autre disciplines

Le problème existe et il prend beaucoup d'intérêt dans la communauté du logiciel qui à conduit entre autres à la programmation objet, aspect, l'écriture d'une programmation fonctionnelle, architecture, gestion de projet, répartition de tâche entre groupe et développeur, processus de développement, optimisation. Etc. Et demeure jusqu'à présent surtout que d'autre question de recherche se pose pour le cas de la programmation objet, relative au niveau d'abstraction appliquée : méthode, classe, module,

Question1: Comment partitionner un programme ou un logiciel informatique ?

Hypothèse(une première question) : $K_M(x)$ est t-il décomposable en somme plus petite?⁴⁵⁵³

Question2: Quelle est la nature des programmes organisé ou aléatoire?.

Il faut beaucoup d'exemple pour illustrer cette relation, car elle existe une représentation d'un programme de système dynamique discret dans réel forte attirante le plus attirant c'est le cas \mathbb{C}^n c-a-d un programme qui se voit vivre non pas dans \mathbb{R}^n mais dans \mathbb{C}^n , cela permet de ce posé la question de son partitionnement, au passage une réflexion mérite beaucoup d'intérêt.

Un Exemple: il y a un programme très court pour faire écrire $2^{1000000}$ (même si son temps d'exécution, lui, est de l'ordre du million) alors que pour faire écrire un nombre donné mais tiré au hasard avec un million de chiffres, il faudra un programme de taille à peu près un million. Noter, en revanche, que l'écriture d'un million de digits du nombre π (le fameux 3, 14 . . .), se fait par un programme très court, car on connaît des algorithmes simples⁴⁶⁵⁴ pour calculer ces digits.

En déduit : quand s'intéresse au nombre (la taille), ou le contenu pour le programme, même si on divise le programme, cela aucun cas n'influent sur notre travail.

Remarque. Liée $p(n)$ avec le refactoring n'est pas toujours correcte et de bon usage, mais on va garder ceci pour quelques choses de très optimal et nous annonçons le résultat suivant : **Le nombre maximum de ligne qu'un refactoring d'un programme donné peut prendre est $p(n)$** on rappelle la formule de Ramanujan-Hardy

$$p(n) \sim \frac{1}{4n\sqrt{3}} \exp\left(\pi\sqrt{\frac{2n}{3}}\right) \quad (7)$$

car il est possible d'écrire qui est d'origine en ligne à deux voir à trois selon le type du paradigme de programmation utilisée, mais si on cherche par contre l'optimisation et la réduction

⁴⁵Même si je me trompe, comment faire pour cohabiter les partitions dans le cas de la physique, et la théorie des nombres d'un côté et celui de la

⁴⁶Il reste à savoir la méthode et l'approche de leur implémentation qui n'est pas du tout une préoccupation pour moi, mon seul intérêt et de prouver la propriété de partitionnement qui apparaît d'une manière purement empirique dans les travaux et les recherches actuelles en génie logiciel

au maximum alors le nombre devient le plus minimal possible ce qui est correspondant au fait avec $K_M(x)$, l'objectif et de relier au premier étape $p(n)$ avec la fonction $\zeta(s)$

Bibliographie

Fonction ζ et matrices aléatoires, <http://math.univ-bpclermont.fr/~royer/art/rcp75/RCP75.pdf>

<http://villemin.gerard.free.fr/Wwwgymm/Identite/RiemAppr.htm>

Fonction de Mobius: <http://villemin.gerard.free.fr/Wwwgymm/Identite/Mobius.htm>

<http://www.les-mathematiques.net/phorum/read.php?13,431810,433141>

<https://webusers.imj-prg.fr/~antoine.chambert-loir/publications/pdf/bnf.pdf>

<https://ncatlab.org/nlab/show/Riemann+hypothesis+and+physics>

<https://cs.stackexchange.com/questions/29010/applications-in-computer-sciences-of-partition-functions>

<https://cs.stackexchange.com/questions/13030/partition-problem-with-distinct-integers>

<https://cs.stackexchange.com/questions/24552/divide-an-integer-into-the-sum-of-consecutive-positive-numbers>

Alain Connes, Motivic Rhythms, 2018 (important et inintéressante pour le programme informatique comme idée basé sur la visualisation)

Stephan Mertens, A physicist's approach to number partitioning

http://www.scholarpedia.org/article/Applications_of_algorithmic_information_theory

1.5.2 Projet : **Entropy software**, principe de Landauer et Zêta fonction

L'objectif est de répondre à la question suivante quelle causalité entre l'entropie logicielle est le principe de Landauer en d'autres termes la dissipation entre l'entropie logicielle et principe limite de Landauer?, tant de questions pour comprendre le lien entre les deux, en présuppose que l'entropie logicielle détermine une entropie macroscopique calculant la dissipation des structures à travers les bugs et l'autre à une entropie microscopique Par définition un logiciel est un ensemble de programme, classe, composant cela dépend la méthodologie de conception et comment l'observateur le décrit. La rigueur sous entendre qu'un logiciel est un système dynamique discret dans \mathbb{R}^n comment une grandeur telle que l'entropie continue peut calculer un phénomène discret, mais s'en demande s'il y a une ?. Nous postulons que l'entropie software qui est une sorte d'entropie empirique et avec la limite de Landauer⁴⁷⁵⁵, c-a-d entre un très haut niveau et un niveau bas, la transformation de l'un à l'autre, un point culminant ce qui me pousse à dire aussi de voir comment cela se passe dans le cadre de la portabilité logiciel qui est vérifié le démon de maxwell un papier ?.

Une autre remarque très forte je n'ai pas encore trouvé un lien entre entropie logiciel et une autre type d'entropie qui permet tout le mécanisme, il y a des papiers sur l'entropie et Landauer pour le calcul réversible, mais pas avec l'entropie software, ce qui sera le lien entre les deux.

⁴⁷Une note biographique sur Ralph Landauer <https://www.nytimes.com/1999/04/30/nyregion/rolf-landauer-pioneer-in-computer-theory-dies-at-72.html>

Ma construction :

Dans l'interminable univers de complexité engendré par l'entropie, je creuse à la construction d'une entropie logicielle qui se lie dans un graphe est écrite avec un formalisme mathématique plutôt rigoureux: soient les **c*-algebra** et leur géométrie des groupes, et celui encore de mécanique statistique je cherche à la fois un formalisme analytique et synthétique, encore plus que j'envisage de construire une théorie mathématique de l'entropie logiciel pure, des choses que on devrait voir ou bien démontré : classification des groupes par leur entropie de graphe, spectre d'entropie

<http://www.di.uniba.it/~lanubile/>

<http://stackoverflow.com/questions/510412/what-is-the-computer-science-definition-of-entropy>

http://www.researchgate.net/publication/3499716_An_entropy_metric_for_software_maintainability

https://en.wikipedia.org/wiki/Software_entropy

http://www.webopedia.com/TERM/S/software_entropy.html

<https://www.toptal.com/software/software-entropy-explained>

C) [Papier recherche pure]

Il y a eu beaucoup de proposition d'entropie, mais ce qui ma fait remarquer qu'Hassen dans son article il a pris en considération deux aspect, le premier le choix des modifications dans le code source, et la deuxième sont utilisation de l'entropie empirique

$H(w|X) = \min_X \{K(X) + H(X) : |H(X) - \log_2 P(X=w)| \text{ is minimal}\}$ je me demande si je peut écrire un papier ou je justifier ceux choix de formule

Lemme : La formule de Hassen n'est autre que l'entropie empirique⁴⁸⁵⁶

Si J'arrive à produire une preuve, alors je pourrait l'envoyer au journal qui a publier celui de Hassen!, mieux encore si j'arrive à généraliser sont résultat et faire une comparaison avec les autres formules dans un article survey, ou bien lui raccordé à celle de Hartely's, le travail de l'article consiste à choisir la formule de Hassen et la transposée dans le cadre de l'écriture de celle de Hartely's, car dans ce travail Olivier Rioul affirme que cette formule de l'entropie de shannon est n'est autre que celle une façon de notre part de vérifier cela, ou bien mixé le travail de Hassen avec celui de Mark Burgess de faire l'analogie avec bug, changement de code et bruit qui pourrait joué la même chose, en cherche alors la capacité du channels $C = 0.5 \log_2(1 + \frac{P}{N})$ ça représente quoi au juste ?.

<http://csttheory.stackexchange.com/questions/17900/who-coined-the-term-empirical-entropy>

D) Entropy software and physics

Comme on l'a déjà répondu, l'entropie de Shannon et Boltzmann entropie sont la même chose, même si elles sont mesurées dans des unités différentes. Vous avez également demandé s'il y a un lien pratique. Il peut ne pas être pratique encore, mais l'idée du refroidissement algorithmique utilise le lien entre ces deux

concepts, et a en effet été démontré expérimentalement^{4957 5058}. Cet article est très important pour prouver ou réfuter mes recherches concernant la relation qui pourrait y avoir entre logiciel (état) et thermodynamique, il y a eu beaucoup de discussion sur le sujet :

48<http://arxiv.org/abs/1103.5985>

49<http://www.nature.com/nature/journal/v438/n7067/full/nature04272.html>

50<http://arxiv.org/abs/quant-ph/0512024>

« Je viens de découvrir les paragraphes suivants de J R Pierce, Introduction à la théorie de l'information, p.206, que je soupçonne serait également soutenir la pensée que l'on pourrait pratiquement utiliser un ordinateur et un programme déterministe pour transformer une chaîne A en une chaîne B de plus entropie de Shannon: "Nous pouvons considérer tout processus qui spécifie quelque chose au sujet desquelles indiquer un système en tant que source de message. Cette source génère un message qui réduit notre incertitude quant à quel état le système est en. Une telle source a une certaine entropie communication théorie par message. Cette entropie est égal au nombre de chiffres binaires nécessaires pour transmettre un message généré par la source. Il faut une énergie particulière par chiffre binaire pour transmettre le message à l'encontre d'un bruit correspondant à la température T du système ». "Le message réduit notre incertitude quant à quel état le système est en, réduisant ainsi l'entropie (de la mécanique statistique) du système. La réduction de l'entropie augmente l'énergie libre du système. Mais l'augmentation de l'énergie libre est tout simplement égal à l'énergie minimale nécessaire pour transmettre le message qui a conduit à l'augmentation de l'énergie libre, une énergie proportionnelle à l'entropie de la théorie de la communication ". "Ceci, je crois, est la relation entre l'entropie de la théorie de la communication et de la mécanique statistique. On paie un prix pour l'information qui conduit à une réduction de l'entropie statistique-mécanique d'un système. Ce prix est proportionnel à la communication- théorie entropie de la source de message qui procède l'information. Il est toujours juste assez haut de sorte qu'une machine à mouvement perpétuel de seconde espèce est impossible »⁵¹

Une autre idée est celle-là conservation de l'information est une éventuelle relation avec les théorèmes d'Emmy Neother une discussion plus pertinente est développée dans ce poste⁵²⁶⁰⁵³⁶¹

Une relation super forte entre ordinateur et univers, trous noirs, Roddier à évoquer ça dans son papier, un papier fort intéressant de Seth Lloyd⁵⁴⁶² explique encore plus , cela renforce ma **théorie Katmerienne** sur la ressemblance entre ordinateur et physique surtout les trous noirs, ce choix n'est pas anodin seulement pour faire appelle au formalisme de Hawking !

l'entropie de Reyni :

<http://csttheory.stackexchange.com/questions/31025/the-utility-of-renyi-entropies/31033#31033>

<https://johncarlosbaez.wordpress.com/2011/02/10/rnyi-entropy-and-free-energy/>

<http://physics.stackexchange.com/questions/73424/deriving-entanglement-entropy-from-renyi-entropy>

Un résultat très intéressant consiste à trouver le lien entre

E) L'entropie software est-elle une force entropique ? ^{55 56 57} Cette question est fondamentale et primordiale elle incite à des recherches encore plus approfondies, publier dans physical letters review B,

Si le Katmerien peut être une information comment cette quantité d'information ne se dégrade pas dans l'espace-temps en d'autre terme il y a aucun bruit un effet rétroactif de l'information ?

Quelle est son énergie ? est-ce qu'elle a une température si oui, comment la calculer ?

⁵¹<http://csttheory.stackexchange.com/questions/12763/are-shannon-entropy-and-boltzmann-entropy-mutually-convertible>

⁵²<http://csttheory.stackexchange.com/questions/3168/energy-considerations-on-computation>

⁵³<http://arxiv.org/abs/quant-ph/9908043>

⁵⁴Black hole computers,

⁵⁵https://fr.wikipedia.org/wiki/Force_entropique

⁵⁶<http://dare.uva.nl/document/2/105001>

⁵⁷<https://arxiv.org/abs/0911.5004>

Par dégradation on suppose qu'il y a un émetteur et récepteur

Quels sont les limites de la théorie Katmerien ?

2 Computer Science mathematics view

Technical Debt, **Control Theory(Package theory)** ; **Kolmogorov Complexity**
and Promise Theory

2.1 Dette technique

S'installer dans le contexte des recherches *actuelles Detecting and Quantifying Architectural Debt , the identification and reduction of Technical debt*

2.1.1 **Loi d'Al-Maqrizi-Gresham**

La dette technique à été inventé par Cunningham en 1992 dans un papier académique en utilisant la métaphore avec la dette qui devient dans le cas du logiciel la dette technique, en allons loin de la métaphore entre dette financière et dette logiciel ou technique, nous poursuivons cette métaphore pour donnée un sens financé au langage de programmation ainsi on utilise la proposition de l'article de German[] langage de programmation est la monnaie. Dans cet article on applique la loi dite de *d'Al-Maqrizi-Gresham* est énonce un théorème correspondant.

La dette technique c'est quoi ?.

Pourquoi langage de programmation est une monnaie ?. Au premier coup ce n'est qu'une hypothèse qui puise dans cette métaphore pour compléter la métaphore de la dette technique, mais il semble que cette simple observation soit payante, en effet un langage programmation fonctionne presque comme une monnaie courante, à titre d'exemple un langage de programmation comme la monnaie possède de la valeur, un moyen pour vendre est acheter en d'autre terme échange, idem pour un langage de programmation par exemple un développeur, société ou groupe voir une communauté de logiciel libre se valorise par sa connaissance et son expertise de ce langage de programmation dans le développement de solution, dans le cadre d'une monnaie il existe aussi l'échange d'une monnaie à une autre ci on plaque sa avec les monnaies papiers ou même on métal qui fera l'objet de notre papier encore plus loin, échangé quand trouve dans les langages de programmations par le biais de syntaxe automatisée tel les traducteurs ou convertisseur, en vient de voir qu'il existe des points de ressemblances entre les deux objets l'un qui appartient à économie et la finance et l'autre faisant partie essentielle de l'informatique, car on peut pas concevoir une tache ou un produit son passé le biais d'un langage de programmation, une autre ressemblance entre les deux est dans la diversité et

La mauvaise monnaie chasse la bonne http://www.aleqt.com/2013/02/19/article_732905.html. Intéressant nous dans cet article à développer l'analogie langage de programmation == la monnaie, ici il y a une « **sourat** ». Et un hadith qui décrit **Refactoring** pour diminuer la complexité(mon idée) !!, cette idée doit être minutieusement étudiât est élaboré dans un contexte.

De quoi il s'agit ?. La loi de Gresham a été annoncée avant Gresham au XVI^e siècle lui par le savant musulman^{58 59 60 61 62}

Simultanément et librement, le système monétaire est appelé « bimétallisme ». Le poids de métal fixe la valeur d'une unité de monnaie. Mais c'est très instable, car l'or ou l'argent peut se dévaloriser l'un par rapport à l'autre si on découvre de nouvelles mines. D'ailleurs, les masses d'or que les Espagnols ont rapportées du Nouveau Monde ont fait baisser son prix relatif par rapport à l'argent qui est devenu du coup plus rare. L'or circulait, alors que l'argent était conservé et disparaissait de la circulation. La « mauvaise » monnaie avait « chassé » la bonne.

Quelques contres exemples. Cette loi ne s'applique pas toujours. Un exemple et un contre-exemple : la disparition de la circulation de la pièce de 5 francs émise avant 1964 au profit de la nouvelle pièce de 5 francs. La valeur du métal argent de l'ancienne pièce étant supérieure à sa valeur faciale, cette pièce a été thésaurisée (la « mauvaise » a donc chassé la « bonne »). En revanche, la dollarisation des économies sud-américaines dans les années 1980 est la conséquence du refus de conserver (et d'utiliser) une monnaie nationale rongée par l'inflation. Dans ce cas c'est la bonne monnaie (le dollar) qui a chassé la mauvaise (la monnaie nationale).

Applications. Au marketing, elle symbolise le fait qu'un produit de qualité supérieure ou d'avancée technologique par rapport à ses concurrents n'est pas pour autant assuré de devenir leader du marché. Il peut parfois en être chassé par un produit plus accessible, de qualité moindre mais qui satisfera pleinement le consommateur. Pour décrire ce phénomène, les professionnels américains du marketing emploient parfois l'expression « *to betamax* » en référence au cas du magnétoscope **Betamax** commercialisé par Sony et qui fut sorti du marché au milieu des années 1980, par son concurrent VHS de JVC Matsushita, de moindre qualité, mais plus accessible au grand public et plus répandu. Dans une entreprise, la loi de Gresham traduit un dysfonctionnement entre ce qui est bon et ce qui est mauvais. En ne refusant pas les mauvais comportements, graduellement ceux-ci chassent les bons. L'inaction des dirigeants face à un dysfonctionnement au sein de leur entreprise entraîne des mutations importantes dans l'organisation et provoque un renversement de situation. Exemple concret de la loi de Gresham: la démonétisation de l'Inde, dont les autorités ont décidé l'arrêt de la circulation de 24 milliards de billets de 500 et de 1000 roupies. Cette politique a mené à la thésaurisation massive de billets de 100 roupies. Peinant à mettre de nouveau billets en circulation, l'administration indienne a donc déclenché une pénurie financière qui a fait considérablement monter la valeur des billets restants. De ce fait, les Indiens effectuent principale-

58<https://arabic.cnn.com/business/2014/08/16/maqrizi-ibntaymiyah-gresham-law>

59<https://www.aliqtisadalislami.net/%D8%A7%D9%84%D8%AA%D8%B6%D8%AF%D9%85-%D8%A7%D9%84%D9%86%D9%82%D8%AF%D9%8A-%D9%88%D8%A2%D8%AB%D8%A7%D8%B1%D9%87-%D8%B9%D9%86%D8%AF-%D8%A7%D9%84%D9%85%D9%82%D8%B1%D9%8A%D8%B2%D9%8A-%D9%88%D9%81%D9%8A/>

60<https://www.dohainstitute.org/ar/Events/Pages/ACRPS-Seminar-Al-Maqrizi%27s-Contribution-to-Economic-Thought.aspx>

61<https://books.google.dz/books?id=Qe5nDQAAQBAJ&pg=PA304&lpg=PA304&dq=%D9%82%D8%A7%D9%86%D9%88%D9%86+%D8%A7%D9%84%D9%85%D9%82%D8%B1%D9%8A%D8%B2%D9%8A+%D8%A7%D9%84%D8%A7%D9%82%D8%AA%D8%B5%D8%A7%D8%AF&source=bl&ots=VQgOjvMfo2&sig=ACfU3U3Tu0vHXQTp2MRvd4LMkjWVAjBN0A&hl=fr&sa=X&ved=2ahUKEwj-sHqr9noAhWHzoUKHaTMAPMQ6AEwCHoECAwQKw#v=onepage&q=%D9%82%D8%A7%D9%86%D9%88%D9%86%20%D8%A7%D9%84%D9%85%D9%82%D8%B1%D9%8A%D8%B2%D9%8A%20%D8%A7%D9%84%D8%A7%D9%82%D8%AA%D8%B5%D8%A7%D8%AF&f=false>

62<https://www.rqiim.com/ahmad.abdulaziz/%D8%A7%D9%84%D9%85%D9%82%D8%B1%D9%8A%D8%B2%D9%8A-%D8%A3%D8%A8%D9%88-%D8%A7%D9%84%D9%86%D9%82%D9%88%D8%AF>

ment des transactions avec leur carte crédit, tandis que l'argent liquide disparaît peu à peu de la circulation.

En informatique. Il suffit de fixer ce que nous entend nous par monnaie ? Et comment la mesurer ? Ou la détectée ?, une première piste s'avère intéressante celle de parler de **dette technique** qui sera notre fil de recherche, peut être qu'il existe d'autres approches ou d'autres voie de développement mais ce que nous intéresse ici c'est celle-ci, la dette technique à bien évidemment des liens très étroites avec l'organisation de l'entreprise ou l'équipe, avec le produit lui-même ou peut être avec le choix du langage de programmation, une première solution s'impose qui est d'ordre organisationnelle

« **Pour contrer la loi de Gresham et ses effets sur le climat de travail, il importe aux dirigeants de faire preuve de courage et d'agir rapidement lorsqu'une situation** commence à se **dégrader**. Ces actions rapides permettront de conserver les bons comportements et de chasser les mauvais. »

Une autre piste qui découle comme corollaire de la dette technique est de l'entropie, quand pourrait bien la voir, en le reformulant ainsi, si un logiciel argumente son entropie, c-a-d qu'un l'intérieur de lui-même le mauvais code chasse le bon code, ou bien la méthode de travail/Workflow au sein d'une équipe est très mauvaise par rapport à une autre,

Première piste? Dans un projet d'informatique scientifique en tombe parfois sur des situations où il est impératif de faire appel à d'autre langage de programmation, pour l'optimisation et accélération du calcul et douté de ce logiciel d'un GUI qui forcément pas dans le même langage en pense la bibliothèque Qt ou PyQt avec un noyau C++ ou en Fortran ou parfois Cython, en constate ici que faire du C++ n'est pas chose aisé par rapport à Python ou PyQt, ce qui nous laisse pensés à éventuelle dégradation dans le futur, ou dans de passage de maintenance un code qui marche mais très mauvaise quand appel souvent un choix de conception ou de configuration, ici on n'est confronté à la Loi de Gresham ? Quels sont les cas les plus probables d'arrivées en informatique plus précisément en génie logiciel. Un exemple simple et rapide est d'utiliser un script shell pour automatiser administration par rapport à une solution déjà existante et complète, ou par exemple écrire une recette point de vue configuration avec un mélange de fichier d'extensions différents et de syntaxes éloignés, la loi de Gresham s'applique le mauvais code qui le mélange de script, fichier Makefile chasse un build configuration pour projet en l'occurrence CMAKE, qui est beaucoup plus organisé, ou bien pleins de solutions qui existantes dans les repositories de distribution Unix/Linux, ressemble énormément aux médicaments, par exemple quand cherche forké ou faire du bindings des protocoles propriétaires, comme l'exemple d'écrire des pilotes pour les drivers en partant par la rétro ingénieurs produisant un code moins bon que celui fournit par le constructeur même si ce dernier publier les spécifications, donc en ici face à la loi de Gresham « le mauvais code chasse le bon code », sur un autre volet l'évolution d'un milestone est une preuve de la Loi de Gresham ou chaque nouvelle version sera considéré comme la bonne et la précédente la mauvaise or il s'avère que, aussi le changement ou la migration d'un langage de programmation vers un autre pour autre utilisée dans un logiciel ou dans une entreprise, est un signe de l'application de la loi de Gresham, réécrire une application GUI lourd en GUI Web, migré d'un framework en PHP vers JS par exemple rentre dans le cadre de la loi de Gresham. Bien que beau

nombre d'auteurs en génie logiciel jugent erronée toute analogie avec économie, peut être que cette loi semble être une exception à la règle

2.1.1.1 Le cas d'un logiciel un seul langage

Dans le cas d'un seul logiciel écrit avec un seul langage de programmation, on parle de Refactoring, pour chasser le mauvais code, suite à dégradation de l'état du logiciel décrit par l'augmentation de son entropie, bien que la situation ne semble pas aussi évidente au premier coup, car si une classe est bien écrite mais semble ne pas être très bien testée ou testée mais en ajoutant d'autres fonctionnalités par exemple pour accélérer la livraison du produit cela peut créer un mauvais code qui serait lui en circulation auquel toute l'équipe y travaille pour l'améliorer oublie le code, qui permet de

2.1.1.2 Le cas des équipes de développements

2.1.1.3 Le cas multi-langage

Le cas d'un logiciel avec deux langages et plus est beaucoup plus compliqué et comporte une complexité en rencontre ce genre de cas dans

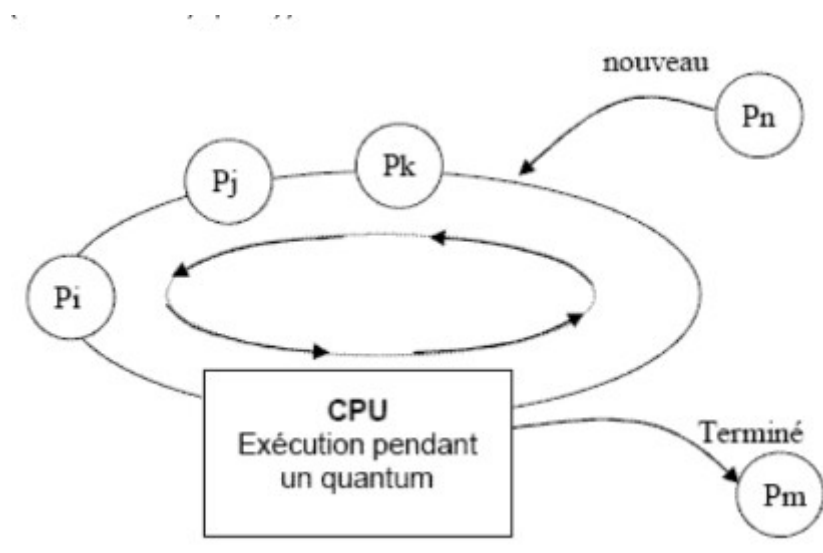
2.1.1.4 Résultat

La loi *d'Al-Maqrizi-Gresham* fait partie de la dette technique, donc on propose une formulation qui suit une logique de la dette technique, Donc d'après cette loi tôt ou tard un seul langage de programmation chasse les autres langages de programmation dans un logiciel aux multi-langages, ou passé à une autre philosophie. Comment expliquer cette loi pour les projets de grande taille tels : système d'exploitation, les navigateurs, suite bureautique, multimédia, calcul scientifique, N'oublions pas que la loi de Al-Maqrizi-Gresham possède des contre-exemples.

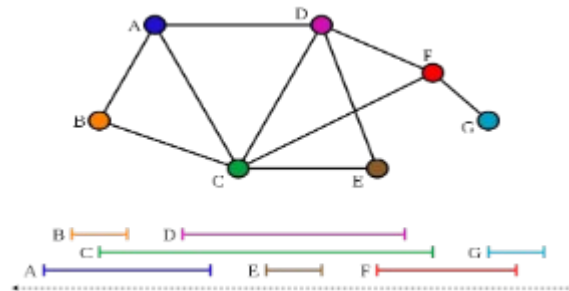
Permet de Prouver que la dette technique est inévitable, ou avec un pourcentage très faible, mais elle y, car la probabilité de son apparition n'est pas seulement au niveau du code, mais aussi sur le plan organisationnel, sans évoquer l'aspect psychologique qui est bien détaillé dans

Ce que aussi conclure de cette loi est que ce proverbe tourne qu'à chaque fois, on injecte une monnaie l'occurrence un langage de programmation mauvais que le précédent ce qui induit une opération de chasse ou que nous interprétons ici d'élection et on fait cette election à la référence de l'algorithme d'élection⁷¹, qui devient « L'élection de la mauvaise monnaie par rapport à la bonne monnaie »,

La loi *d'Al-Maqrizi-Gresham* nous permet de mettre en place un théorème plus puissant sur l'inévitabilité de la dette technique dans le génie logiciel, c-à-d cas à tous les plans que soient niveau code ou autre la dette technique reste toujours même s'il est caché voire difficile d'être délectable ou bien impossible d'éviter, comme si on s'applique un avec elle un algorithme de tourniquet ou Round Robin, ou le combiné avec l'algorithme de l'élection



2.1.2 Projet : Graphe d'intervalle



d'un côté de comprendre l'interaction et la cohésion et de détecter le retard et aussi reformuler pas mal de problématique avec un graphe d'intervalle en s'appuyant sur l'hypothèse qu'il existe une dette technique minimale caché (en référence à deux choses : la première l'article EPR et l'**hypothèse Bayésienne**). Dans le site Xebia exprime ça d'une manière très palpable. En résumé on va exploiter deux directions, la première purement algorithmique, immédiate mais nécessite un certain bagage théorique mathématique, le deuxième et purement mathématique celui du transport optimal dans tous ces états

2.1.2.1 Graphe d'intervalle et transport optimal

Définition : Un graphe est un graphe d'intervalle s'il existe une bijection entre les sommets du graphe et un ensemble de segments d'une droite tel que deux sommets u et v sont reliés par une arête si et seulement si les segments correspondant à u et v

La deuxième approche est intéressante, car elle sous entend que nous choisissons notre ensemble de départ qui est \mathbb{R} , nombre problème et question se résume en une seule dimension sur le mouvement et répartition de segment, qui coïncide avec le transport optimal, comment ordonner une équipe pour éviter de tomber à chaque fois dans le même piège de la dette technique ?, ce la résume le fait d'éviter de transporter à chaque fois la même masse, mais aussi il ressemble beau-

coup à une question de mariage stable entre segment, tâche et code à analysé, je suis plus intéressé par un problème de transport optimal pur qu'autre chose, ou tout se transforme en panoplie de théorème et définition, mais pour le moment on va rester terre à terre. L'élément clé est bien sûr la dette technique, le cas du portage est intéressant aussi en liaison avec les graphes d'intervalles.

2.1.2.2 Y a-t-il un bruit dans la dette technique ?

2.1.2.3 **Projet : Quel lien entre dette technique et Kolmogorov Complexité ?**

2.1.2.4 **Project : DSM & technical debt**

2.1.2.5 **eDSM & Hypergraphe**

2.2 Promise theory⁶³

2.2.1 **Promise et vitesse souhaité et réel dans le cadre de son espace-temps sémantique**

2.2.2 **Promise theory et Digital Twin**

Une théorie originale consiste à lier promise et digital Twin pourquoi? Le passage est naturel et fluide entre les deux, mais pas aussi simple le réel et le virtuel son en dépendance mais ayant un comportement sans obligation ni opposition

2.2.2.1 **En Ajoutant le fait que chaque individu à sa propre composition physique qui est différents des autres**

2.2.3 **Une visualisation plus intéressante entre eDSM et Promise theory**

Pourquoi c'est une très bonne idée de la faire, par une approche analogique et de symétrie d'approche on constate que la théorie de la promise s'occupe de l'organisation et de l'interaction mettant en jeu la coopération, on prend n'importe quels schémas générés par la promise on voit très vite qu'il reflète des dépendances entre composant équivalente d'une coopération, celle-ci est semblable d'une représentation par une matrice carrée N^2 permet de bien expliquer la densité et etc,, des promises, Question peut-on trouver un théorème pour assuré un isomorphisme entre Promise et DSM?, un cadre plus technique consiste aussi à développer un théorème-outil pour tout ce qui est DevOps ?

63 Mark Burgess a mis au point **Game theory**, sur le volet application et interprétation non seulement ça mais plutôt faire de cette idée une théorie mathématique au propre sens du terme, ayant une interprétation physique, dans un cadre plus large cette interprétation permet d'aller plus loin surtout sur la représentation formelle, d'après le livre de Burgess, cette théorie est en étroite liaison avec les probabilités, économie et théorie des jeux, dans un chapitre consacré à la mécanique quantique il évoque une attitude plutôt quantique de la théorie, il est aussi important de noter un autre lien tout à fait intéressant celui avec la théorie de jeux à champ moyen l'usage des EDP et une grandeur très importante.
<http://gdrtop.math.cnrs.fr/>

2.2.4 Digital Twin et technologie DevOps (idée pour papier technique)

2.2.5 Digital Twin et Agilité des projets

<https://www.supmeca.fr/recherche/ingenierie-des-systemes-mecatroniques-et-multiphysiques/>

2.2.6 Promise theory et mariage stable

Au-delà du contexte linguistique et sémantique apparent entre promesse et mariage, il existe une autre relation beaucoup plus formel défini par le concept de promesse tel défini par Mark Burgess, dans le cadre de la théorie de la promesse liant deux individus ou deux entités non rationnel ; cette liaison pourrait être vue comme étant un mariage stable c-a-d une complétude de la forme finale d'une promesse, l'idée semble extraordinaire et originale, par définition la promesse comprend deux agents le corps et le scope de la promesse, sans obligation ni possession. Si on commence par l'idée d'affectation entre des notes d'étudiants et leur affection aux universités. Posons la question au sens de la promesse,

L'algorithme du mariage stable, est responsable de marier deux entités, dire responsable est équivalent au mot promet. En d'autre terme promet de marier deux entités. Un peu de précision ; on a n hommes et n femmes

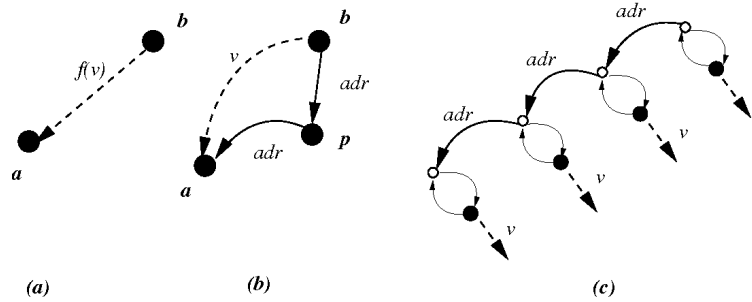
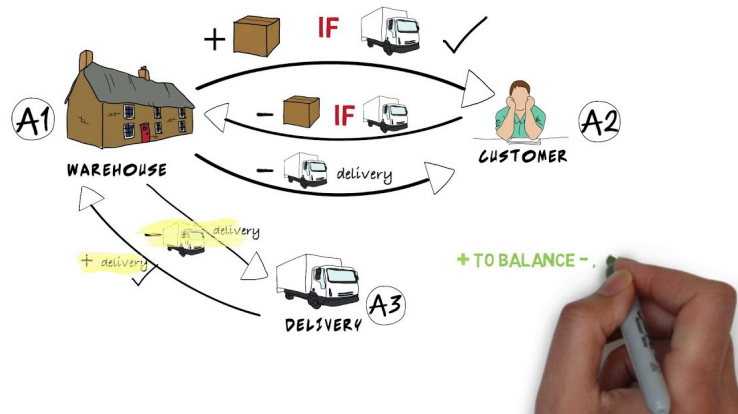
Remarque : une situation instable dans le cas de mariage stable se traduit en *obligation* dans le cas de la promesse, ou dans le cas de forcer un pair, je tiens à préciser que l'assemblage entre les deux théories n'est pas très clair et rigoureuse, car l'idée est en c'est début, cependant il doit rester le fait que mariage stable et promesse doit se concorder, dans le cas le plus classique pour un deuxième rappel, quand il y a un **mariage stable**, avant le mariage doivent être une **promesse** de mariage, reste à situer le rôle des préférences qui à priori pourrait être équivalent au body le corps de la promesse,

Théorème (WIP): On s'intéresse à déduire une condition nécessaire ou suffisante entre la théorie de la promesse ou bien le mariage stable, la réponse approche très rapidement

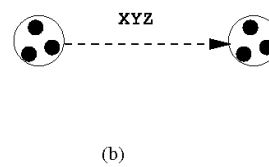
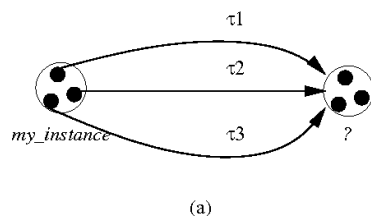
Un autre point qui fortifie le lien entre les deux théories et celle de la représentation géométrique du mariage stable à savoir les graphes bipartis, hypergraphes

2.2.7 Projet Hypergraphe-Promise

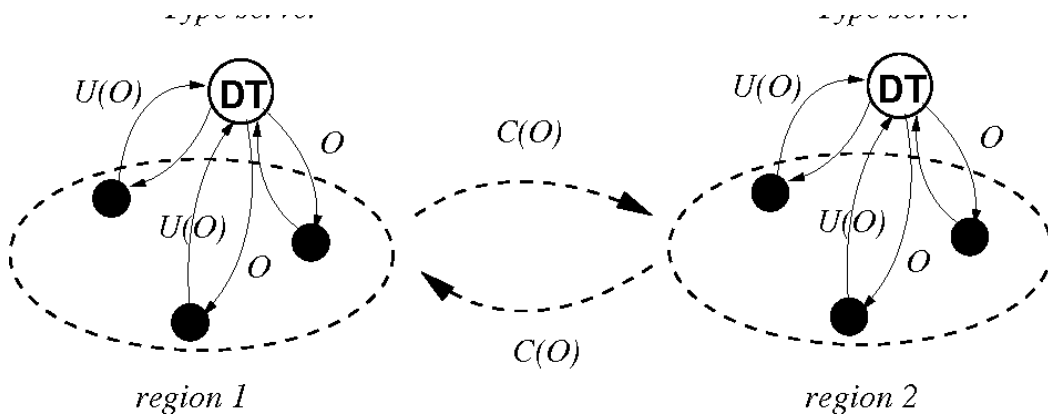
Vu l'importance et la généralisation des hypergraphes pour les graphes, dans ce billet on s'intéresse à proposer un langage graphique basé sur les hypergraphes pour la théorie de la promesse, nous montrons qu'il est très intéressant de proposer les hypergraphes que les graphes et en quoi nous auront un plus dans cette direction, bien que une représentation est simplement représentable pour une Promise



```
class XYZ
{
    type1 name1;
    type2 name2;
    type3 method(type4 input);
}
my_instance;
```



Par exemple, Pourquoi compliquer alors qu'il y a plus simple nous diront n'est-ce pas ?



Il arrive que dans de nombreuses situations l'usage de la promesse nous met un peu en déroute vue que c'est une théorie de l'interaction entre différent acteur, facile de la dessinée, mais elle pose quand même quelques difficultés : en appliquant dans le cadre de système distribué[], DNS[] ou dans l'interaction de management et très forte. Par exemple quand une promesse contient différente flèche chacun d'elle en contienne une, alors simplifié par exemple aussi de la limitée

1.1. Références :

(a) Indexation de graphes à partir d'une structure d'hypergraphe

2.2.8 Une formulation catégorique de la promise theory

2.2.9 Une réécriture des concepts théoriques de l'informaticien

américain Dewayne E. Perry /Unifying Theoretical Foundation for Software Engineering⁶⁴

Un autre point qui fortifie le lien entre les deux théories et celle de la représentation géométrique du mariage stable à savoir les graphes bipartis, hypergraphes

2.3 Control theory

2.3.1 Package theory

La théorie des packages telle nommé par Board dans son article est une intéressante branche de génie logiciel qui étudier le portage et le déploiement dans le gestionnaire de paquet fait appel à différents outils classiques d'analyse de donnée, méthode formelle et statique pour tout ce qui est liée au génie logiciel expérimental. Cette partie comporte plusieurs aspects intéressant de la théorie des packages, mais nous allons développer seulement quelques problématiques à savoir: les EDO/EDP de retards, réseaux de réactions chimiques et le lien avec la théorie de la représentation à base d'un résultat obtenu par Burgess, qui fait l'objet, les directions de recherche possibles :

2.3.2 Fixé les définitions de stationnaire, stable, dynamique, incompressible et compressible

2.3.3 Variability de la configuration

2.3.4 Source d'idée et d'analyse

<https://www.pkgsr.org/pkgsrCon/2006/slides/updates.html> (update packages)

⁶⁴ <https://sites.google.com/site/logiquecatégorique/autres-seminaires/nantes/20160401-Lafforgue-Topos>

<https://www.pkgsrcon.org/pkgsrcon/2006/slides/pkgsrcon-on-df/index.html> (pkgsrcon of DragonFly)

<https://www.netbsd.org/gallery/presentations/maya/2018-pkgsrcon-gcc/index.html>

2.3.5 Chemical Reaction

Dans le contexte formel de la théorie des packages, le principal moteur et gère les dépendances et surtout le concept de transformation d'un état à un autre, il existe une théorie concurrente nouvelle ou ancienne sur ce volet allant de l'analyse, topologie ou même catégorique les faire tous est un

2.3.6 Hypergraphe et package theory

Dans ce paragraphe nous listons les questions que pourrait traiter la théorie des packages en utilisant les hypergraphes

- (a) Un modèle formel usage de la théorie des catégories.
- (b) Modèle visuelle des graphes de dépendances vue par les hypergraphes
- (c) Simplification algorithmique
- (d) Par exemple s'en intéresse aux cas de réunir à la fois des **dépendances forte et faible** qui est un peu difficile à présenter à la fois ou bien les détecter dans les gestionnaires de paquets
- (e) Décomposition et partitionnement des graphes de dépendances complexes, l'article « **Hypergraph partitioning for social networks based on information entropy modularity** » pourrait être un très bon démarrage
- (f) Réinterprétation des questions, l'exemple du théorème de Helly (sujet d'article à part entière)
- (g) Une approche d'utiliser la géométrie tropicale, par exemple dans le papier « Computing the Vertices of Tropical Polyhedra Using Directed Hypergraphs » et « THE TROPICAL DOUBLE DESCRIPTION METHOD » de [Xavier Allamigeon](#), [Stéphane Gaubert](#) & [Éric Goubault](#) et

Où est l'originalité dans tout ça ?

2.3.7 Projet : Portabilité

Remarque 1.0: en peu constaté que qu'il existe une *portabilité réflexive* à savoir le mécanisme qui permet d'expliquer de porter un logiciel sur la même plateforme directement

- 2.3.7.1 Un exemple empirique : une approche coté programmation vers un OS, la difficulté de langage Golang

BSDCan 2019
The Technical BSD Conference

Porting Go to netbsd/arm64

Starting out with no knowledge in Go or ARM64, how do you begin porting a language? Go might be one of the hardest languages to port.

Having done that, I came back to tell the tale of how.

- Unusual characteristics of Go
- Rules to write the 400 lines of assembly code necessary for a port
- Debugging in the absence of a functional debugger

The talk will describe how to start on such a large project, how to find your way around a large unfamiliar codebase, what's a calling convention, how to debug mistakes in your Go port, and some notes about why Go chose to make such strange choices.

Attached files

- [slides \(application/pdf - 507.1 KB\)](#)

SPEAKERS
Maya Rashish

SCHEDULE
Day: Talks #2 - 18 May - 2019-05-18
Room: DMS 1129
Start time: 16:00
Duration: 01:00

INFO
ID: 1000
Event type: Lecture
Track: Hacking
Language used for presentation: English

FEEDBACK
Did you attend this event? Give Feedback

https://www.bsdcan.org/2019/schedule/attachments/529_Porting_Go_to_netbsd_arm64.pdf

https://docs.google.com/presentation/d/1Ii_il41s53XCK5KRFBFAELgOYLGog1nTXJwRCtaZ4X4/edit#slide=id.g4708423b54_0_367

2.4 La thèse de Hassine(intéressante pour utiliser un graphe d'intervalle)

mots-clefs : espace métrique fini, espace ultra-métrique

Hassine dans son mémoire «Evolution biologique et applications en génie logiciel » à évoque une correspondance entre bio-informatique et génie logiciel, en donnant à chaque composant du logiciel un code en 0 et 1

Il y a une autre ancienne remarque évoqué par Burgess dans son livre sur l'intérêt des distances, une idée intéressante consiste à voir en profondeur le papier « Ultrametrics in the genetic code and the genome » ou en remplace le code génétique avec du code source ou le génome est remplacé par les composants ou les classes, ou apparaît le travail de Hassine.

<http://www.les-mathematiques.net/phorum/read.php?14,592702,592810>

<https://math.stackexchange.com/questions/129978/proving-a-distance-between-molecules-defines-a-metric-space>

<https://mathoverflow.net/questions/12394/representability-of-finite-metric-spaces>

2.5 L'usine pkgsrc

Une minutieuse observation nous fait découvrir qu'un gestionnaire de paquet est similaire à une usine industrielle ou biologique l'occurrence une cellule, les ressemblances sont frappantes, un paquet source est identique à la matière première, ou les pièces d'une voiture avant l'assemblage de celle-ci, l'existence de chaîne de montage le cas pour un gestionnaire de paquet, une autre ressemblance et l'intervention humaine dans les deux cas, malgré ancienneté de processus de fabrication dans les usines et l'avancée technologique lors de la création de paquet on remarque que l'humain est indispensable dans les deux cas, les choses changent un peu avec l'industrie 4.0 et l'avenue de l'IoT et la robotique dans les usines classiques mais l'humain reste présent dans le cas du logiciel, malgré l'automatisation et génération intelligente du code, dans le cas d'une usine 4.0

l'humain prend le rôle d'un superviseur avancé de l'administration système, dans les usines il y a une phase de vérification ça existe aussi dans le cas des gestionnaires de paquets, avec des outils pkglint, ur2pkg, lintpkgsrc dans le cas de pkgsrc ou celle de Debian, Fedora ou Gentoo etc. Qu'est-ce qu'on fait de tout ça ? Si en prend la figure ci-dessous

2.6 Project : pkgsrc-wip

Deux choses intéressantes pour ce papier peut être trois mais manque de temps et de pertinence en rester sur deux, la troisième est sur l'essence même du pkgsrc-wip à savoir, le comportement des newbies qui peu être formulé dans un contexte de la théorie des jeux, bifurcation des testes/packages tester. Mon objectif pour clore ce papier et de voir l'évolution de pkgsrc-wip comme étant une catégorie dans l'espace de Burgess.

2.6.1 pkgsrc-wip/Digital Twin

Donnons un autre souffle pour la technique clone package tester

2.6.2 Analyse de complexité de pkgsrc sous un autre regards

En s'inspirant sur les présentations techniques données lors des conférences pkgsrc.org, on choisit quelqu'une pour un approfondissement

2.6.3 What pkgsrc needs most

Un travail de Amitai Schlair présenté dans le pkgsrccon 2010

<https://pkgsrc.org/pkgsrcCon/2010/slides/schmonz/what-pkgsrc-needs-most.html>

2.7 Hypothèse : Une théorie de mariage stable pour la portabilité

Ici j'annonce le résultat suivant qui sera formulé en terme plus formel dans le futur :

«Le clonage est une forme faible du portage »

le lien entre le portage et le clonage logiciel, l'approche et beaucoup plus profond, je propose une théorie qui explique que le clonage est une propriété technique de la portabilité, l'idée de la démonstration s'appuie sur le **mariage stable**

Explication :

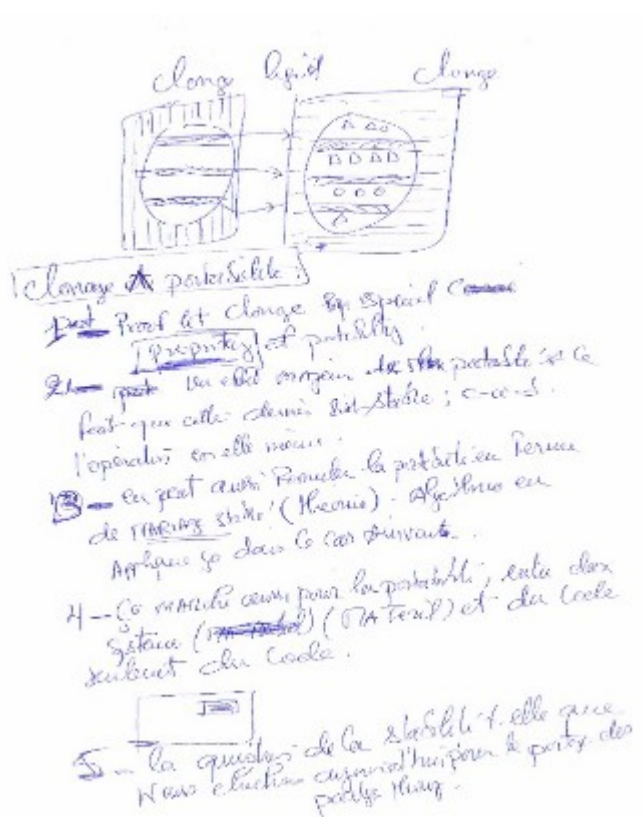
Est plutôt elle facilite la tâche du portage et non pas le contraire, nous identifiant aussi qu'il existe des classes bien particulières de Type comme c'est le cas pour le clonage classique, donc en somme notre idée n'est pas de voir le clonage comme étant une chose négative mais bien au contraire le comprendre comme étant un cas particulier et très faible du portage logiciel qui lui est beaucoup plus complexe vue qu'il introduit à la fois le matériel et logiciel.

L'exemple le plus simple et de constater le fait qu'il existe une portabilité parfaite en peut l'assimilé à un **clonage** est les packages généralement le package en une certain classification dans ce cadre mais cela n'est pas suffisant.

Comment le démontrer ?:

Nous utilisant le mariage stable pour démontrer ce théorème, bien que ce théorème ne suffit pas intégralement, est inédite de donnée une autre interprétation au clonage et la portabilité en utilisant le mariage

stable, cette exception se résume en deux chose la première la stabilité entre deux environnements, la deuxième est que le clonage est une propriété ou un cas particulier de la Portabilité, mettre la lumière sur ça c'est le prouvait rigoureusement. Un article sur le sujet « **AN EXTENDED STABLE MARRIAGE PROBLEM ALGORITHM FOR CLONE DETECTION** »



3 Espace de Burgess

Les espaces de Burgess est une sorte d'espace classe d'**espace de configuration** défini par Mark Burgess mais c'est un grands rigueur mathématique ou il défini un espace tel-quel les variables sont mémoires⁶⁵, disque dure, etc. Intérêt de telle construction c'est que ce type d'espace fait appel au système dynamique une autre formulation utilise la théorie des jeux quand en peu la généralisé sur différente problématique rencontrée en informatique mais aussi dans d'autres domaines, de facto en peu ajouter toute une mécanique telle-que l'espace de phase, utilisation d'un Hamiltonien et d'autre ébauche fructueuse, en trouve aussi des résultat original, une mécanique de groupe de Lie dit l'idée de transformation de groupe, dans cette voix voir l'évolution de ce type de groupe utilisé la géométrie des groupes pour ces transformations, le théorème de convergence d'une configuration, quand peut le généralisé d'une manière analytique. Un autre point innovant dans l'approche de Burgess est le pont entre génie logiciel et théorie d'administration système notamment avec son papier fondateur. Une autre partie qui à attirer mon attention est la politique de configuration ou la stratégie de configuration⁶⁶ en utilisant

3.1 Unicité et existence

⁶⁵Dans un papier tout récent en va mettre en éclairer l'analogie entre réaction chimique et package théorie, plus exactement les étapes de fabrications

⁶⁶Mark Burgess, Configurable immunity for evolving human-computer systems. 2003

L'un des exemples de l'unicité de paquet et le mot clef **CONFLITS** dans pkgsrc et **conflit** dans le gestionnaire de paquet de Debian GNU/Linux, deuxième.

Les paquets ne sont pas traités différemment qu'ils aient le même nom ou non : deux paquets du même nom mais différant dans leurs dépendances, L'existence d'un paquet et triviale.

3.2 **Observation: Diagramme de Feynman**⁶⁷

On peut se positionner dans le cadre du livre **Quantum Techniques for Stochastic Mechanics** du mathématicien de John Baez et Analytique Administration theory de Mark Burgess, les gestionnaires de paquets sont un excellent environnement pour observer et étudier l'évolution des logiciels pour deux visions purement génie logiciel et la deuxième purement administration système,

3.3 **Un impressionnant papier dû à Vito Volterra**

Un impressionnant papier dû à Vito Volterra publier un an avant son décès, sur Fluctuations dans la lutte pour la vie⁶⁸, l'intérêt de cet article et l'originalité de l'idée de porter le principe du moindre action à la biologie, facilitant ainsi sa réécriture vers la dette technique

4 **Liste de publication à venir pour 2020-**

4.1 **Research publications:**

4.1.1 **Mathématique**

4.1.1.1 **Sur la formule de Terrence Tao de la fonction** $f(x) = \sqrt{x+1}$ un programme pour SymPy

4.1.2 **Computer science**

<http://oioannou.com/2016/blog/copyright-and-patches/>

<https://www.computer.org>

<csdl/mags/so/2016/01/mso2016010105.html>

<https://www.computer.org/csdl/mags/so/2016/01/mso2016010101.html>

<http://www.dmst.aueb.gr/dds/pubs/conf/2015-ESEM-CodeStyle/html/SLK15.pdf>

<https://lwn.net/Articles/656294/>

4.1.3 **pkgsrc-wip(l) :[classic analyse]{*utilisation de diagramme de collaboration*⁶⁹, *process graph*}[Long] **Program 1089** /Theory/**

4.1.4 **Le cas multi-projects/ le cas non multi-projects**

4.1.4.1 **Le concept du WIP? (evoqué dans le premier article avant d'arrivée ici, je remarque que il n'y a personne qui a proposé une définition du nom WIP**

⁶⁷<https://journals.openedition.org/philosophiascientiae/301>

⁶⁸<https://zbmath.org/?q=ai%3Avolterra.vito+py%3A1967>

⁶⁹ *La théorie Learning de Stephan Smale.*

dans le contexte de l'informatique, donc je dois le définir dans l'état de l'art de la publication scientifique)

Reference :

4.2 Ici on a besoin des papiers sur les patches logiciels

4.3 Understanding personality differences in software organisations using Keirsey temperament sorter

Référence :

Forking and coordination in multi-platform development: a case study

Building It Together: Synchronous Development in OSS <http://web.cs.ucdavis.edu/~filkov/papers/together.pdf> (2014), il pourra être utilisé deux fois dans le contexte du wip

The Relationship Between Folder Use and the Number of Forks: A Case Study on Github Repositories

OU publier cet article :

<http://www.doublecause.net/index.php?page=physique.htm>

ici : <http://www.igi-global.com/journal/international-journal-open-source-software/1123>

ici : <http://onlinelibrary.wiley.com/journal/10.1002/%28ISSN%292047-7481>

ici : <http://onlinelibrary.wiley.com/journal/10.1002/%28ISSN%291097-024X>

Reference : Who is going to Mentor Newcomers in Open Source Projects ?

Objet connecté + principe d'Asimov-Dror-Derouiche

Hypothèse d'observateur en parémiologie (application de la théorie d'information de Shannon et avec ne approche épistémologique)

Vers une nouvelle reformulation à base proverbiale des 18 problèmes de Smale

Exemple de chercheur qui n'est pas liée à l'université : Clément de Seguin Pazzis ; <http://dsp.prod.free.fr/>

Cours Vidéos :

<https://fabricebaudoin.wordpress.com/> (EDP/Geometry)

<http://www.les-mathematiques.net/phorum/read.php?8,507657,507657>

Concept : L'individu doit être le centre du progrès de la communauté à condition que cet individu doit agir, contribuer et combattre le mal quand il se manifeste à l'intérieur de la communauté sans donner beaucoup d'importance à l'existence ou l'inexistence des dirigeants et des élites du pouvoir.

<http://www.freefullpdf.com/>

Noyau de découverte et d'invention: le noyau de ma pensée pour ajouter une nouvelle théorie, ou une approche originale, par exemple dans le cadre d'une théorie informatique plus particulièrement est la suivante

Application: il s'agit d'appliquer des théories mathématiques en informatique, physique : en suivant une stratégie brute ou directe, par exemple appliqué la théorie des catégories dans le contexte de la programmation composant ou pour proposer une formulation rigoureuse de la théorie des packages, mais la théorie mathématique qui a connu un grand succès est son précédent c'est « la théorie des graphes » jusqu'au point d'être un outil naturel pour toute , même si elle développait par une approche algorithmique cela nous pousse à poser des questions connexes, est entamer dans un contexte

ce qui permet d'expliquer et simplifier beaucoup de questions purement informatiques. La remarque vitale est que l'informatique est un vaste champ de recherche pour les Comptes première approximation

Lemme: sorte d'étape préliminaire à une démonstration; Proposition déduite d'un ou de plusieurs postulats dont la démonstration prépare celle d'un théorème. Un lemme est un petit théorème de nature technique. En pratique, c'est une sorte d'aparté qui allège la démonstration principale.

Utilité : La question d'utilité est très active et

Prolongement/extension : il s'agit de faire apporter des solutions et de nouveautés pour une seule théorie valable pour les mathématiques, l'informatique. Par exemple dans une surface de Kummer est ce que on peut voir dans cette dernière une évolution

Réécriture d'une théorie mathématiquement : Est ce que on peut refaire une théorie mathématique à travers un phénomène informatique

Note sur l'évolution Pendant le cycle de vie du logiciel , environ 52 % de l'effort est consacré à l'entretien et à 47% de cet effort est estimé à consacrer à la compréhension du programme. Ce qui rend l'entretien plus difficile est l'hétérogénéité des langages et paradigmes de programmation , en grande logiciel de fait sont composés de modules souvent codés avec une programmation différente et langages de spécification . Mainteneurs ainsi confrontés à une variété de différents langages de programmation qui rend la compréhension du programme et la maintenance des logiciels difficile et coûteux. Cela introduit de nouveaux défis à la fois pour le domaine de l'analyse des logiciels et de l'évolution du programme en tant que programmeurs doivent faire face à une variété de paradigmes et langages de programmation . Nous pensons qu'il y a la nécessité de vues globales de soutien aux développeurs de faire face efficacement à la complexité et à faciliter la compréhension du programme et l'analyse de ces systèmes hétérogènes. En outre, l'hétérogénéité des systèmes ne se limite pas à la langue mais également l'incidence de la licence des composants. En fait , la licence est un autre type d'hétérogénéité introduit par la grande réutilisation de code source ouvert . Système de développement en réutilisant des composants existants diminue le coût de développement de phase . Pourtant, il peut introduire un autre type de problème : en raison des différents droits et obligations de chaque licence et le grand nombre de licences et leurs différentes versions qui peuvent être contradictoires, il est difficile de respecter toutes les obligations. En outre , les ingénieurs logiciels ne sont pas

pas bien formés aux questions juridiques sont confrontés à un ensemble complexe de droits et obligations juridiques qu'ils ont de la difficulté à suivre et à comprendre. Et la majorité de l'effort dans la recherche cible problème technique du développement de logiciels et re-engineering et d'un peu d'attention est dirigée vers la complexité juridique . Sans l'automatisation des tâches liées aux questions juridiques , il peut souvent être complétée seulement en employant des avocats à un coût énorme , les avantages financiers potentiels sont clairs.

Dans ce contexte de programmes hétérogènes , notre objectif est de développer un outil de re-engineering pour analyser l'impact des changements dans les systèmes hétérogènes licence et multi- langue. Tout d'abord, nous voulons étudier l'impact du changement dans les systèmes multi- langues en général et de l'étendre à la délivrance des licences. Nous voulons gérer l'évolution du système en fournissant l'impact des changements , par exemple l'évolution de l'architecture comme la modification de l'interconnexion entre les deux modules ou de l'évolution de la licence (mise à jour vers une nouvelle version) peut introduire des conflits inter- licence. En outre , nous voulons suggérer architectures possibles pour éviter un conflit juridique entre les licences en utilisant expert du système.

<https://sites.google.com/site/ferdaousboughanmi/Research-Interest>

Idea::

Preliminary lessons from a software evolution analysis of Moodle

Organisations éducatives comptent plus fréquemment sur les logiciels libres / open source pour leur infrastructure. Ce est principalement en raison de coûts plus faibles et plus grande polyvalence. Mais gagner aperçu sur ce type de projets ne est pas facile, car ils sont généralement développés avec une équipe géographiquement distribué, avec de nombreux bénévoles et même plusieurs entreprises collaboratrices. La qualité de ces projets ne est pas facile d'évaluer, ou le risque existe que le projet ne est pas viable. Dans cet article, nous présentons une analyse préliminaire de l'évolution du logiciel qui permet de mieux comprendre un projet libre / open source logiciel. Cette analyse offre la possibilité de prendre des décisions éclairées. Nous avons sélectionné par conséquent, le projet éducatif Moodle connue comme une étude de cas

in mathematics you don't understand things. You just get used to them.

En mathématiques, on ne comprend pas les choses. On s'y habitue (Johann von Neumann)