

Autour des diagrammes de décision quantiques

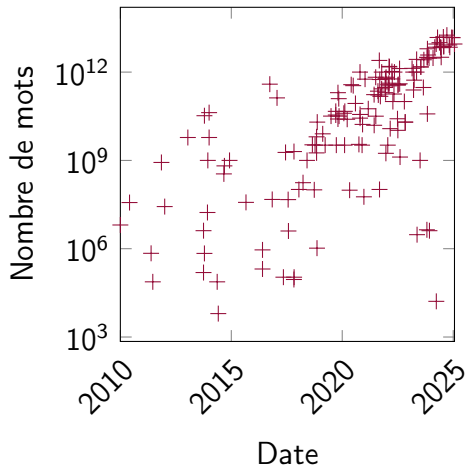
Malo Leroy

Parcours recherche – CentraleSupélec

17 février 2025

Les bases de données croissent rapidement

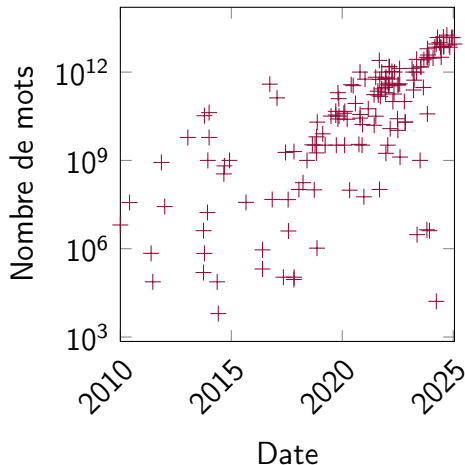
Les algorithmes classiques sont parfois inefficaces



. Nombre de mots utilisés pour l'entraînement des modèles de langage

Les bases de données croissent rapidement

Les algorithmes classiques sont parfois inefficaces



Les **algorithmes quantiques** permettent de résoudre certains problèmes plus efficacement

. Nombre de mots utilisés pour l'entraînement des modèles de langage

Les machines quantiques sont en développement et resteront coûteuses financièrement



Il y a un besoin d'outils de simulation et de vérification d'algorithmes quantiques

Les simulations sont très coûteuses en temps de calcul

<i>Grover</i>	Classique	Quantique	Simulation
Complexité	N	\sqrt{N}	$N\sqrt{N}$

Elles nécessitent une **structure de données** adaptée

État de l'art

- Interprétation abstraite
- Arithmétique des intervalles réels
- Diagrammes de décision quantiques

État de l'art

- Interprétation abstraite
- Arithmétique des intervalles réels
- Diagrammes de décision quantiques

Solution : diagrammes additifs abstraits

L'**interprétation abstraite** permet de déterminer des propriétés ou d'accélérer des calculs

Exemple : signe d'une expression $e = (3 + 2) \times (-5)$

$$\begin{aligned}\text{signe}(e) &= (\text{signe}(3) + \text{signe}(2)) \times \text{signe}(-5) \\ &= (\oplus + \oplus) \times \ominus \\ &= \oplus \times \ominus \\ &= \ominus\end{aligned}$$

L'interprétation abstraite permet de déterminer des propriétés ou d'**accélérer des calculs**

Elle peut être exacte ou **approximative**

L'interprétation abstraite est applicable aux **intervalles réels**

$$[1,2] * [-1,1] = [-2,2]$$

$$[1,2] + [-1,1] = [0,3]$$

$$[1,2] \wedge [-1,1] = [1,1]$$

Le résultat de l'opération est **le plus petit intervalle** contenant tous les résultats élément par élément

Une fonction booléenne

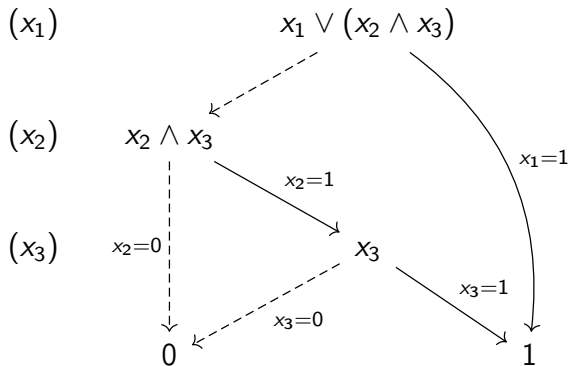
$$f : \{0,1\}^n \rightarrow \{0,1\}$$

peut être représentée par une **table de vérité**

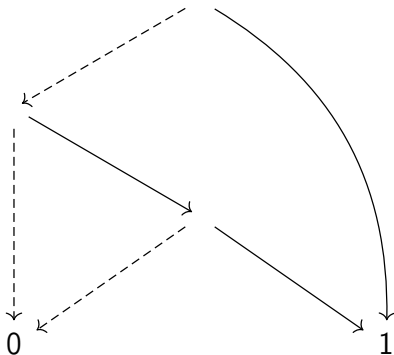
x_1	x_2	x_3	$f(x_1, x_2)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

pour $f(x_1, x_2) = x_1 \vee (x_2 \wedge x_3)$

Les **diagrammes de décision** permettent de représenter des fonctions booléennes



Les **diagrammes de décision** permettent de représenter des fonctions booléennes



On tire parti de la **structure** de la fonction

Un **état quantique** est une superposition d'états incompatibles

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle \quad (\text{un qubit})$$

Un **état quantique** est une superposition d'états incompatibles

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle \quad (\text{un qubit})$$

n qubits $\Rightarrow 2^n$ états incompatibles

On note les états sous forme de **vecteurs**

$$\alpha |01\rangle + \beta |10\rangle = \begin{pmatrix} \alpha \\ 0 \\ \beta \\ 0 \end{pmatrix}$$

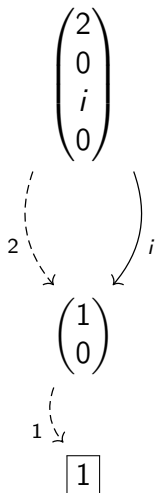
La représentation usuelle est proche des **tables de vérité**

x_1	x_2	$\langle x_1 x_2 \psi \rangle$
0	0	α
0	1	0
1	0	β
1	1	0

pour $|\psi\rangle = \alpha |00\rangle + \beta |10\rangle$

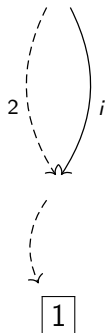
Les états peuvent être représentés par des **diagrammes de décision quantiques**

On tire parti de la **structure** de l'état



Les états peuvent être représentés par des **diagrammes de décision quantiques**

Dans le pire cela reste **exponentiel**



Retour sur l'état de l'art

- ✓ Interprétation abstraite
- ✓ Arithmétique des intervalles réels
- ✓ Diagrammes de décision quantiques

On va utiliser ces concepts ensemble,
avec une nouveauté : l'**additivité**



Retour sur l'état de l'art

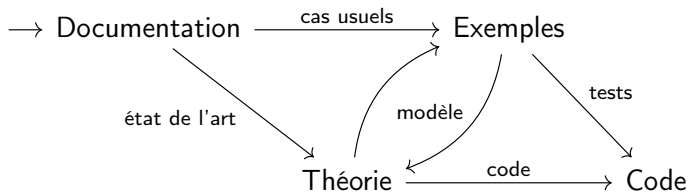
- ✓ Interprétation abstraite
- ✓ Arithmétique des intervalles réels
- ✓ Diagrammes de décision quantiques
- + Nouveauté : additivité

Solution : diagrammes additifs abstraits

Objectifs

- **Modèle formel** de diagrammes de décision additifs abstraits
- **Implémentation** du modèle

Méthodologie



GitHub pour la gestion de projet

- **Issues** pour les tâches et le bugs
- Priorités, tailles, et deadlines
- Branches et **merge requests**

Modèle

S6 Intervalles de \mathbb{C} cartésiens & polaires

S6 Diagrammes

S6 Approximation locale, globale

S6 Fusion forcée

S6 Algorithmes de réduction

S7 Erreur

S7 Application de portes

Exemple : on considère l'état $\begin{pmatrix} 10i + 2 \\ 4i + 1 \\ 2i \\ i \end{pmatrix}$

Exemple : on considère l'état $\begin{pmatrix} 10i + 2 \\ 4i + 1 \\ 2i \\ i \end{pmatrix}$

Il existe des **régularités**.

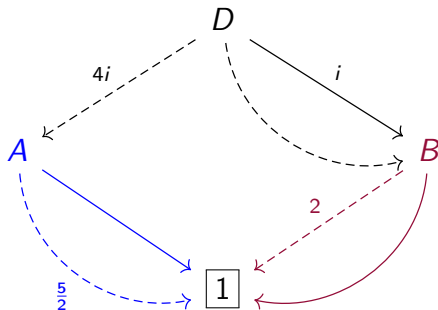
Exemple : on considère l'état $\left(\begin{array}{c} \left(\begin{array}{c} 10i \\ 4i \end{array} \right) + \left(\begin{array}{c} 2 \\ 1 \end{array} \right) \\ i \left(\begin{array}{c} 2 \\ 1 \end{array} \right) \end{array} \right)$

Il existe des **régularités**.

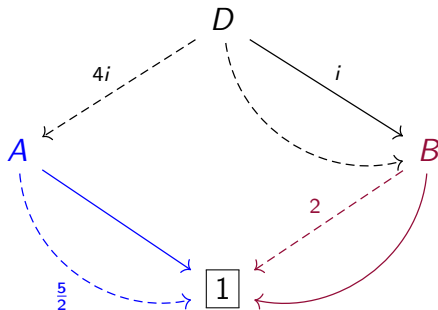
Exemple : on considère l'état
$$\begin{pmatrix} 4i \begin{pmatrix} 5/2 \\ 1 \end{pmatrix} + \begin{pmatrix} 2 \\ 1 \end{pmatrix} \\ i \begin{pmatrix} 2 \\ 1 \end{pmatrix} \end{pmatrix}$$

Il existe des **régularités**.

Exemple : on obtient le diagramme additif

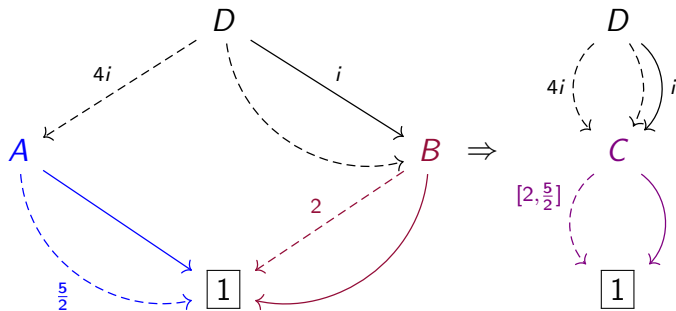


Exemple : on obtient le diagramme additif

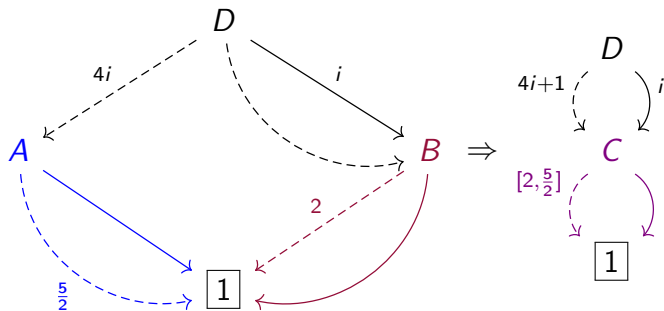


Réduisons ce diagramme

On peut forcer la fusion de A et B



On peut forcer la fusion de A et B



On peut toujours plus réduire les diagrammes



Gain en espace arbitrairement grand (jusqu'à exponentiel)

Comment choisir quels diagrammes fusionner ?



Erreur : on peut calculer l'erreur induite par un diagramme

Les circuits quantiques sont basés sur des **portes**

On les modélise par des **matrices**

Les circuits quantiques sont basés sur des **portes**

On les modélise par des **matrices**

Exemple : Hadamard

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$H \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

On veut appliquer une porte M à un diagramme D

On veut appliquer une porte M à un diagramme D

Si $\mathcal{E}(D)$ = évaluation du diagramme

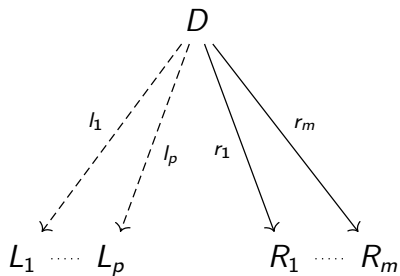
On veut $\mathcal{E}(M(D)) = M \mathcal{E}(D)$

On veut appliquer une porte M à un diagramme D

Si $\mathcal{E}(D)$ = évaluation du diagramme

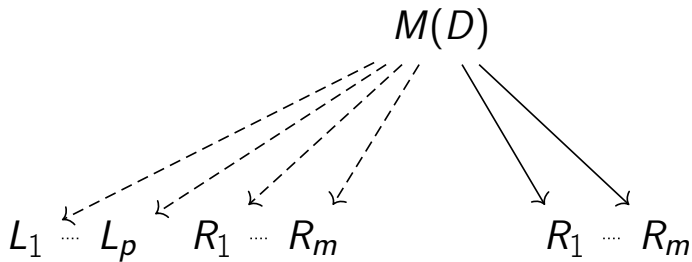
On veut $\mathcal{E}(M(D)) = M \mathcal{E}(D)$

$$D = (\{(l_1, L_1), \dots, (l_p, L_p)\}, \{(r_1, R_1), \dots, (r_m, R_m)\})$$

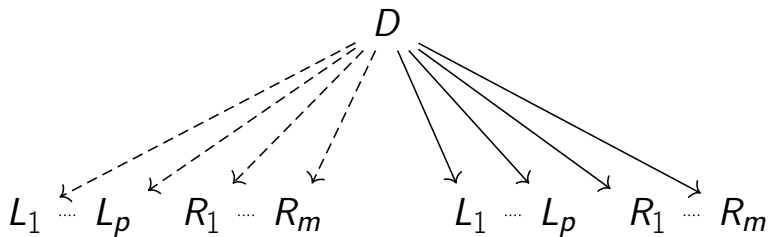


Avant application de la porte

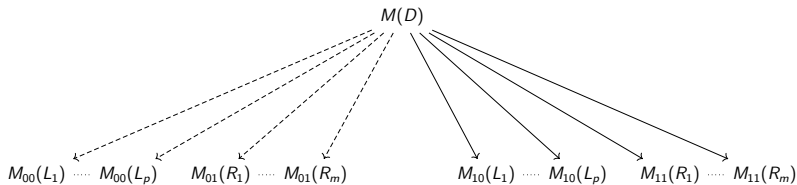
1.



2.

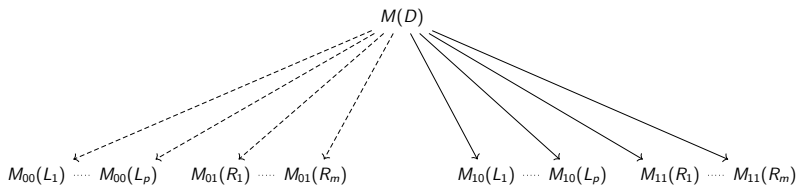


3.



$$M = \begin{pmatrix} M_{00} & M_{01} \\ M_{10} & M_{11} \end{pmatrix}$$

3.



$$\mathcal{E}(M(D)) = \begin{pmatrix} \sum l_i M_{00} \mathcal{E}(L_i) + \sum r_j M_{01} \mathcal{E}(R_j) \\ \sum l_i M_{10} \mathcal{E}(L_i) + \sum r_j M_{11} \mathcal{E}(R_j) \end{pmatrix} = M \mathcal{E}(D)$$

Implémentation

- S6 Intervalles de \mathbb{C} cartésiens & polaires
- S6 Diagrammes : construction, évaluation
- S6 Fusion forcée
- S6 Algorithmes de réduction
- S7 Diagrammes aléatoires
- S7 Erreur
- S7 Application de portes
- S7 QASM

Suite du projet

■ Implémentation

- Interface graphique
- Benchmarks

■ Ajustements

- Fonctions d'erreur
- Algorithmes de réduction

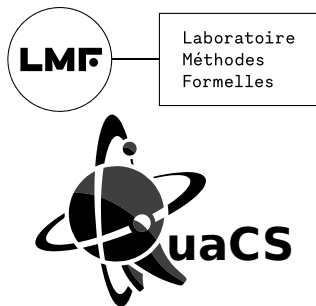
■ Nouveaux concepts

- Automates d'arbres
- Diagrammes de décisions et applications localement inversibles (LIMDD)

Cadre du projet

Formation future

- **Encadrant** : Renaud Vilmart
- **Équipe** : QuaCS
- **Laboratoire** : Laboratoire
Méthodes Formelles



Année de césure *Digital Tech Year*

- Semestre au Paris Digital Lab
- Projets tech variés en équipe
- Stage de 6 mois en entreprise ou laboratoire, en France ou à l'international

Année de césure *Digital Tech Year*

- Semestre au Paris Digital Lab
- Projets tech variés en équipe
- Stage de 6 mois en entreprise ou laboratoire, en France ou à l'international



ALICE & BOB



QUANDELA

Après la césure

- S8 à CentraleSupélec
- S8 Pro (stage)
- S8 académique international

Dominantes / mentions

- **Informatique et numérique**

- Sciences du logiciel
- Architecture des systèmes informatiques

- **Physique et nanotechnologies**

- Quantum engineering

Autres formations

- ARTeQ (ENS Paris-Saclay)
- QMI M2 (Télécom Paris, entre autres)

Conclusion

Questions

Implémentation

■ Code (4,9k lignes)

- Langage C++
- LLVM / Clang
- Ninja
- CMake

■ Tests

- Google Test
- GitHub Actions



Mise en forme

- **Versionnage**

- Git
- GitHub

- **Documentation**

Doxygen

