Introduction au Calcul Quantique - 1 - Prolégomènes

JM.Torres - IBM Quantum Ambassador

16 novembre 2022

Algorithme

Liste finie d'instructions non ambigües permettant de résoudre un problème dont les entrées et les sorties sont définies.

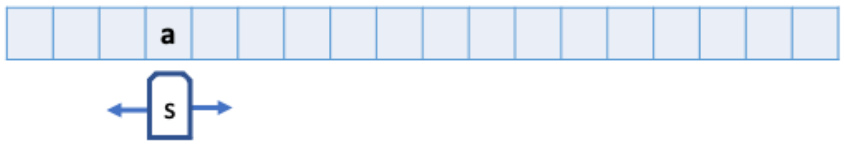
Muhammad Ibn Mūsā al-Khuwārizmī, né aux alentours de 780 à Khiva dans la région de Khwarezm (Ouzbekistan actuel)(wikipedia)

Machine de Turing

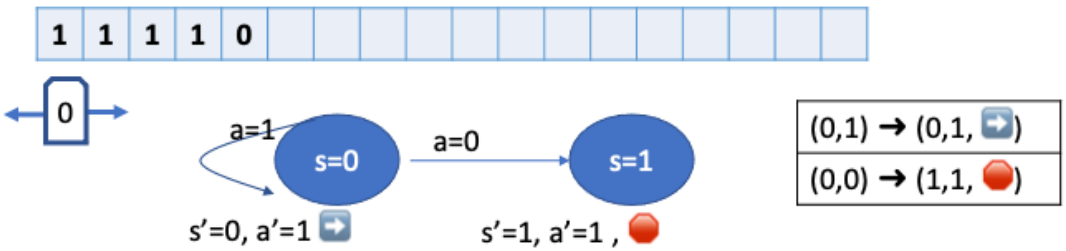
Une machine de Turing est un concept, permettant de définir rigoureusement la notion de calcul et d'algorithme, elle comporte :

- un ruban infini (contenant un nombre infini de cases, ou positions),
- une tête de lecture/écriture,
- un ensemble d'états pour la machine $\{s_0, s_1, s_2, ...\}$,
- un "alphabet" de symboles pouvant être écrits sur le ruban $\{a_0, a_1, a_2, ...\}$,
- une table de transitions :

  (état courant, symbole lu) $\rightarrow$ (nouvel état, symbole écrit, mouvement : gauche, droite, stop)



Par exemple (incrément unaire) :



Thèse de Church-Turing, et MTU

Thèse de Chruch-Turing

Une suite calculable γ est déterminée par une description d'une machine qui calcule γ [...] et, en fait, n'importe quelle suite calculable est susceptible d'être décrite en termes d'une table c'est-à-dire d'une machine de Turing.

En clair (mais approximatif) : tout ce qui est calculable peut-être calculé par une machine de Turing, dont la table de transition est l'algorithme.

Machine de Turing Universelle

En 1936 Alan Turing montre qu'il existe des machines de Turing pouvant simuler n'importe quelle autre machine de Turing.

En 2002, Yuri Roghozin prouve une MTU avec seulement 4 états, 9 symboles et 24 transitions.

- https ://fr.wikipedia.org/wiki/Thèse_de_Church

Thèse de Church-Turing et calcul quantique

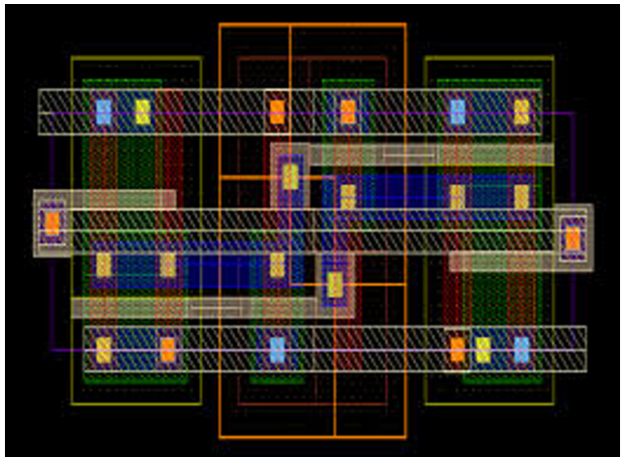
Thèse de Church et calcul quantique [modifier | modifier le code]

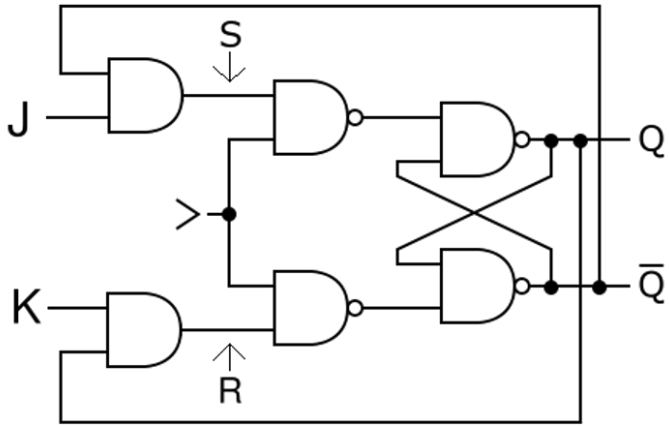
Les différents modèles de calcul purement mathématiques élaborés pour modéliser la calculabilité ne prennent pas en compte de processus physiques. Comme l'ont montré Robin Gandy[11] , Nachum Dershowitz (en) et Yuri Gurevich (en)[12] les considérations physiques ne sont pas négligeables. Or, d'après le Principe de Church-Turing-Deutsch, tout processus physique fini peut être simulé par un mécanisme physique fini. C'est donc pour cela que l'informatique s'inspire de la physique classique, et utilise des bits et non des qbits (le modèle quantique "rompt le charme de la thèse de Church-Turing étendue"[13]).

En 1982, le physicien Richard Feynman s'est posé la question de savoir si les modèles de calcul pouvaient calculer l'évolution de processus quantiques. Il est parvenu à démontrer que cela était possible, mais de manière inefficace, inapplicable en pratique. Or, la nature est visiblement capable de « calculer » cette évolution de manière efficace. La question se pose donc inévitablement de savoir si les processus quantiques sont en relation avec une autre forme de calculabilité et s'ils remettent en cause la forme physique de la thèse de Church.

Modèle de Circuit et Universalité

Manipulation de bits (à valeur 0 / 1, 0V / 5V, True/False ...) dans des circuits (en général du silicium) au moyen de l'agèbre de Boole





Universalité

L'ensemble $\{\text{NOT}, \text{AND}\}$ constitue un ensemble universel d'opérateurs :

- il est possible de constituer toute fonction logique à partir de cet ensemble d'opérateurs.

Démonstration :

1. Théorème de Morgan

Théorème de Morgan

L'opérateur OR (ou) peut être construit à partir de l'ensemble $\{\text{NOT}, \text{AND}\}$

En effet : $a \vee b = \neg(\neg a \wedge \neg b)$

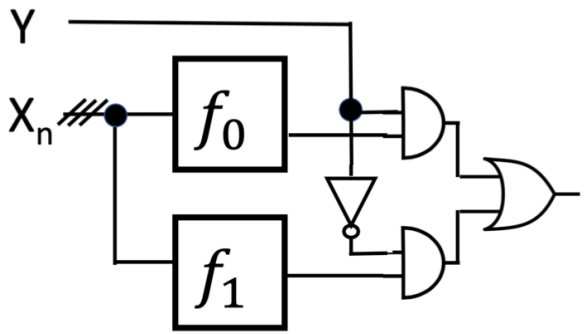
2. Par récurrence
   On suppose que l'on peut former toute fonction de 2 variables à partir de $\{\text{NOT}, \text{AND}, \text{OR}\}$ (c'est à dire $\{\neg, \wedge, \vee\}$)
   Soit f une fonction logique de $n+1$ variables booléenes à valeurs dans $\{0, 1\}$, on définit à partir de f deux autres fonctions (de n variables) :
   - $f_0(x_1, x_2, ..., x_n) = f(0, x_1, x_2, ..., x_n)$
   - $f_1(x_1, x_2, ..., x_n) = f(1, x_1, x_2, ..., x_n)$

   f_0 et f_1 peuvent être construites avec $\{\neg, \wedge, \vee\}$, d'après l'axiome de recurrence (car elles sont de taille n), alors on remarque que :
   $(\neg y \wedge f_0(0, x_1, x_2, ..., x_n)) \vee (y \wedge f_1(1, x_1, x_2, ..., x_n)) = f(y, x_1, x_2, ..., x_n)$ est une expression de f construite avec $\{\neg, \wedge, \vee\}$.

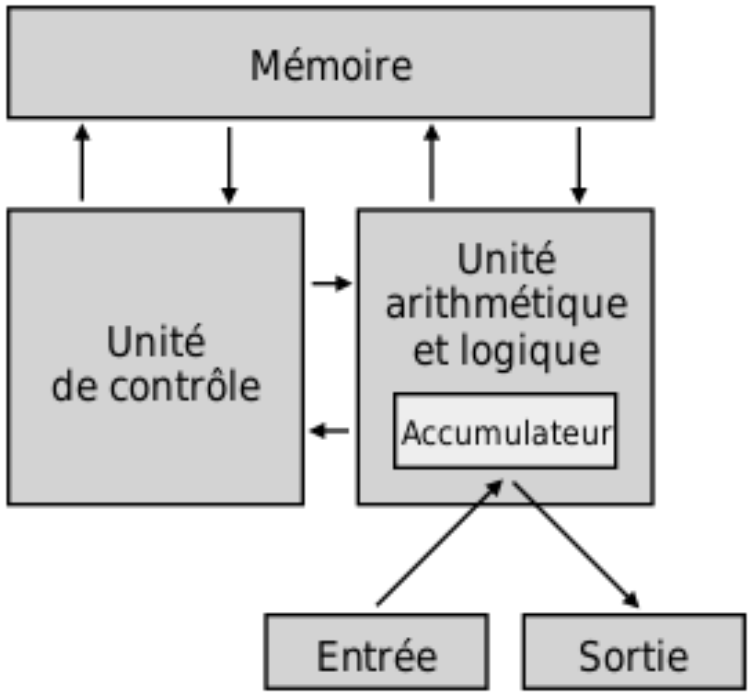
Universalité ...

Le modèle ainsi (construit), appelé modèle à circuit pose des problèmes pratiques !

- la nombre de portes logiques croit exponentiellement avec le nombre de variables
- la taille du circuit et sa consommation électrique aussi
- les problèmes de fanout (alimentation des portes en aval) et de croisement (pour implémenter sur du silicium)
- il faudrait construire un nouveau circuit pour chaque nouvelle fonction



Modèle de Von Neumann



L'architecture de Von Neumann contourne le problème de la taille du circuit en rendant le matériel programmable. Cependant, le problème de la durée (ou de la taille mémoire) n'est pas résolu.

Complexité

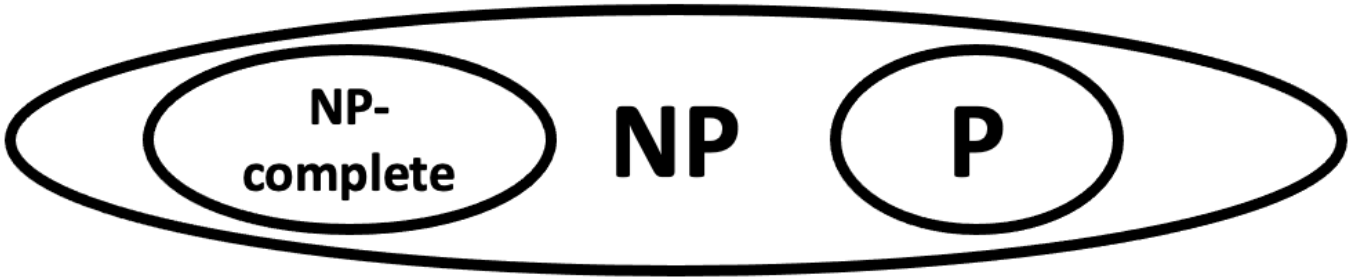
- La complexité en informatique s'intéresse au comportement des algorithmes en terme d'espace (mémoire) et de temps de calcul.
- Pour un problème donné P, la complexité en temps étudie la manière dont le nombre d'instructions à exécuter avant d'obtenir la solution varie en fonction de la taille n du problème.
- Si $C(n)$ est ce nombre d'instructions, on cherche à évaluer $C(n)$ lorque n devient grand. Pour cela une possibilité est de comparer $C(n+1)$ à $C(n)$
- On utilise la notation de Landau "grand O"

Variation de C(n)	Complexité	Notation « grand O »
C(n+1) = C(n)	constant	$\mathcal{O}(1)$
C(n+1) = C(n) + ε (par exemple C(n+n) = C(n) + 1)	Logarithmic	$\mathcal{O}(\log(n))$
C(n+1) = C(n) + k	Linear (kn)	$\mathcal{O}(n)$
C(n+1) = C(n) + n	Polynomial	$\mathcal{O}(n^2)$
C(n+1) = C(n) + n^k	Polynomial	$\mathcal{O}(n^{k+1})$
C(n+1) = C(n)*2	Exponential	$\mathcal{O}(2^n)$
C(n+1) = C(n)*n	Exponential	$\mathcal{O}(n!)$

P=NP ?

P et NP sont deux classes de problèmes de décision :

- Un problème p peut être résolu de manière "efficace" s'il existe un algorithme en temps polynomial permettant de le traiter. Ce problème est dans P.
- Il y a une classe de problèmes pour lesquels nous connaissons un algorithme qui les traite en temps exponentiel, et pour lesquels il existe un algorithme qui peut vérifier en temps polynomial si une solution est correcte ou pas. C'est la classe des problèmes NP
- On ne sait pas si $P = NP$
- Parmi les problèmes de classe NP certains comportent la difficulté de tous les problèmes NP, ils sont dit NP-complets, et si pour un seul de ceux là on arrive à prouver un algorithme en temps polynomial, alors on pourra conclure que $P = NP$.
- $P = NP$? est un des problèmes du millénaire.



Exemples de NP complets

- CLIQUE (graphes) : une CLIQUE est un graphe non orienté dont chaque sommet est connecté à tous les autres. La taille de la CLIQUE est le nombre de ses sommets. Dans un graphe G existe-t-il un sous-graphe qui soit une CLIQUE de taille n ?
- SOMME de sous-ensemble (arithmétique) : étant donné un ensemble fini d'entiers E et un entier s, existe-t'il un sous ensemble de E dont la somme des éléments vaut s ?
- MAXCUT (graph) : étant donné un graphe (dont les arêtes peuvent être pondérées), une coupe sépare les sommets en deux ensembles. La valeur de la coupe est la somme des poids des arêtes dont les sommets ne sont pas dans le même sous ensemble. Une coupe est MAXCUT si sa valeur est la plus grande de toutes les coupes possibles.
- SAC A DOS (optimisation combinatoire) : étant donné un ensemble d'objets ayant un poids et une valeur : quel est le choix d'objets à emporter maximisant la valeur et respectant un poids total maximal ?

https ://fr.wikipedia.org/wiki/Liste_de_problèmes_NP-complets

3-SAT

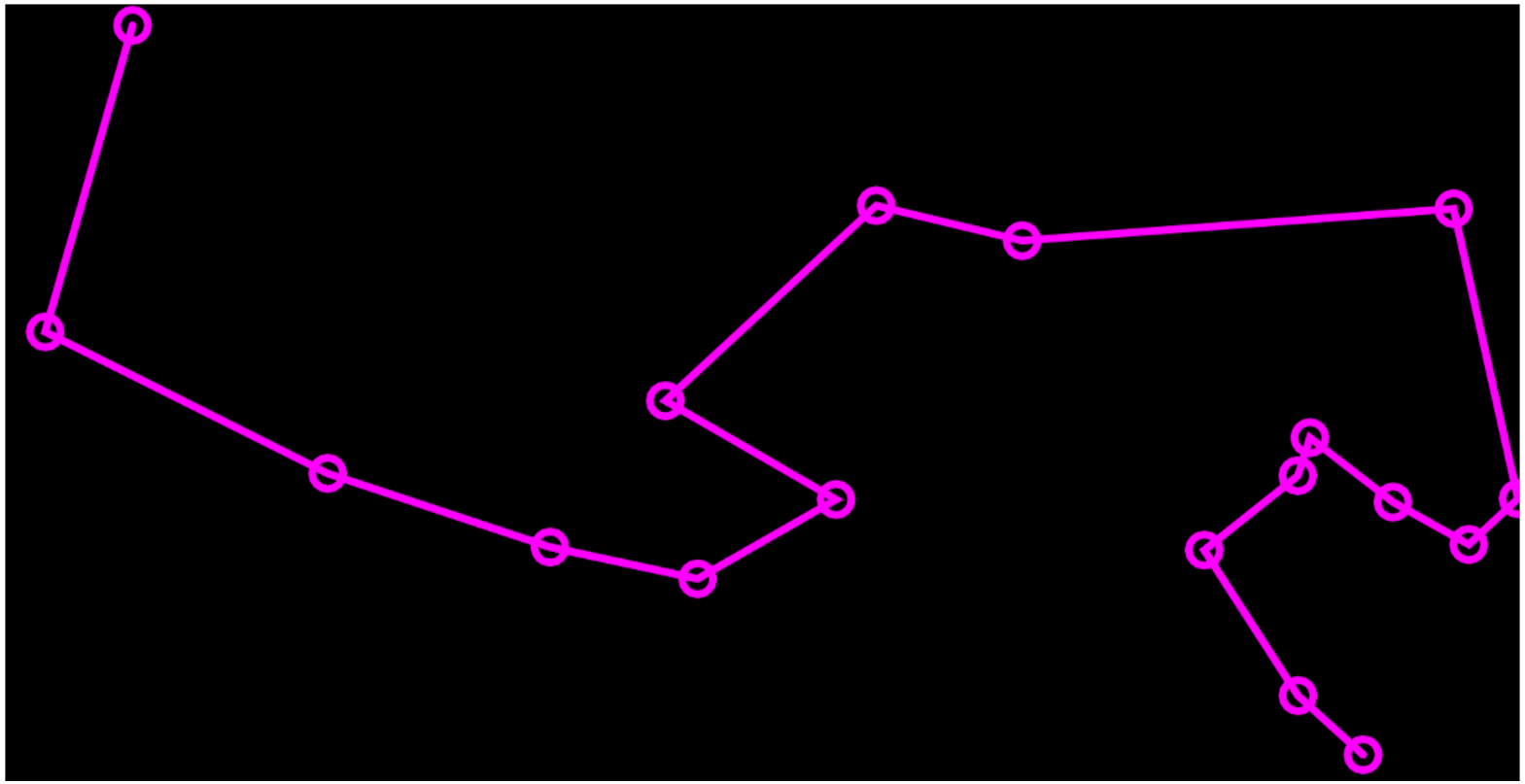
Problème de statisfiabilité : étant donné une expression logique F utilisant n variables booléennes b_i, existe-t-il une affectation des valeurs des b_i telle que $F = True$?
En particulier un problème 3-SAT s'écrit sous forme de conjonction de 3-disjonctions ("et de 3 ou"), par exemple :

$$F = (b_0 \vee b_3 \vee \neg b_7) \wedge (b_2 \vee \neg b_3 \vee \neg b_5) \wedge (b_1 \vee b_3 \vee b_7) \wedge ... \wedge (b_4 \vee \neg b_5 \vee b_6)$$

La résolution du problème 3-SAT impose d'explorer toutes les affectations possibles des n variables, soit 2^n possibilités. L'algorithme de recherche est en temps exponentiel par rapport au nombre des variables n. Et il s'agit d'un problème de classe NP.

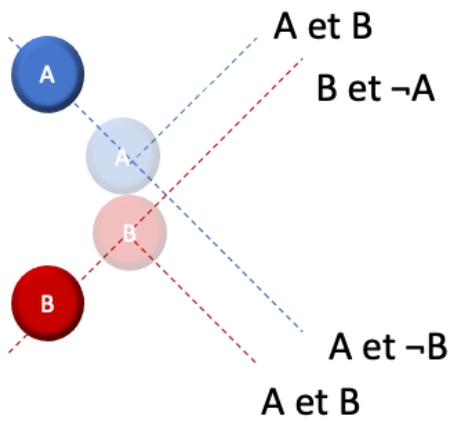
Note : 2-SAT est de classe P

TSP Démo



Reversibilité

Ordinateur mécanique reversible : "machine à boules de billard" (BBM) : Edward Fredkin et Tommaso Toffoli, 1982



Reversibilité ...

Ordinateur mécanique reversible : "machine à boules de billard" (BBM) : Edward Fredkin et Tommaso Toffoli, 1982

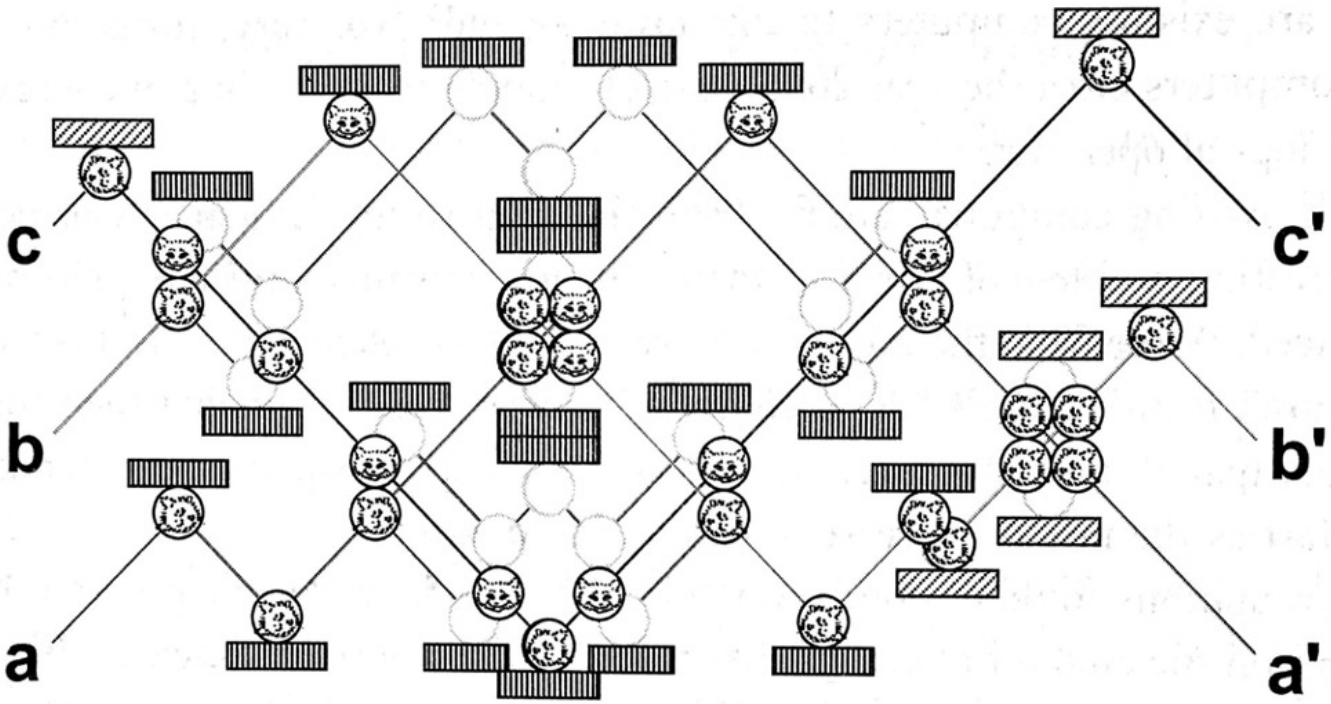
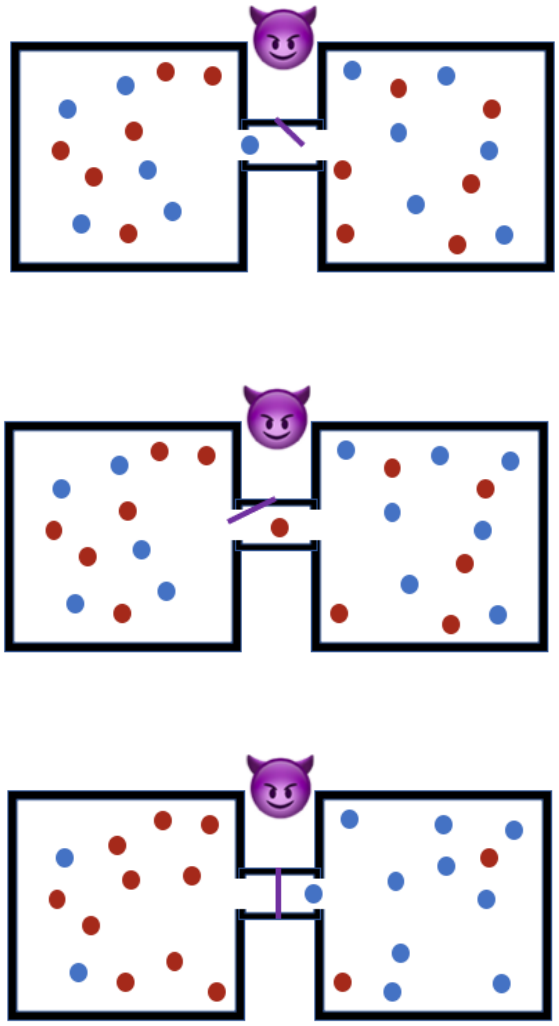


Figure 3.14. A simple billiard ball computer, with three input bits and three output bits, shown entering on the left and leaving on the right, respectively. The presence or absence of a billiard ball indicates a 1 or a 0, respectively. Empty circles illustrate potential paths due to collisions. This particular computer implements the Fredkin classical reversible logic gate, discussed in the text.

Michael Nielsen and Isaac Chuang (2000), Quantum Computation and Quantum Information, Cambridge University Press.

Le démon de Maxwell



Principe de Landauer (1ère forme)

Lorsqu'un bit d'information est effacé d'un ordinateur la valeur de l'énergie échangée avec son environnement est au minimum $kT\ln(2)J$.

Principe de Landauer (2nde forme)

Lorsqu'un bit d'information est effacé d'un ordinateur l'entropie de son environnement augmente d'au moins la valeur de l'énergie échangée avec son environnement est au minimum $kT\ln(2)J$.

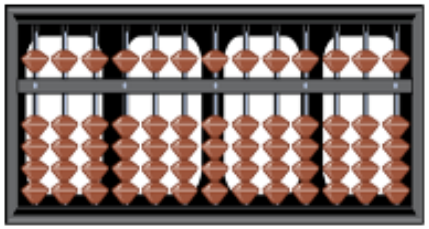
$$\mathbf{3.10^{-21}\ J}$$

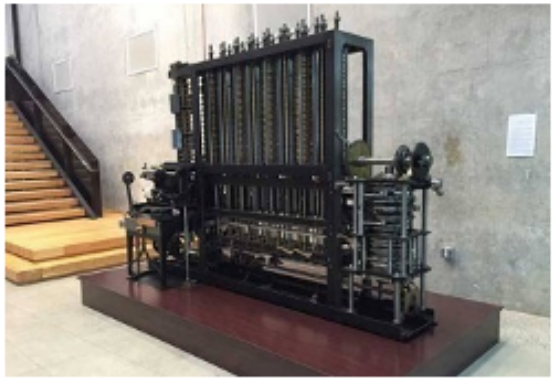
(les ordinateurs actuels évoluent à environ 500 fois au dessus de cette valeur)

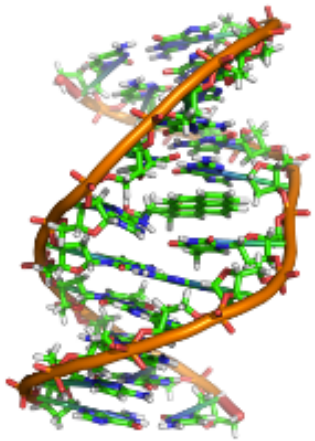
Le calcul réversible (parmi lesquels le calcul quantique) n'est pas concerné par cette limite.

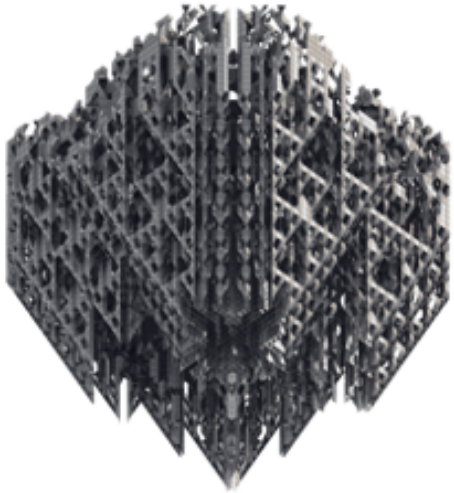
Ref : Rolf Landauer : Irreversibility and heat generation in the computing process. IBM Journal of Research and Development, 5 :183, 1961.

Modèles de calcul



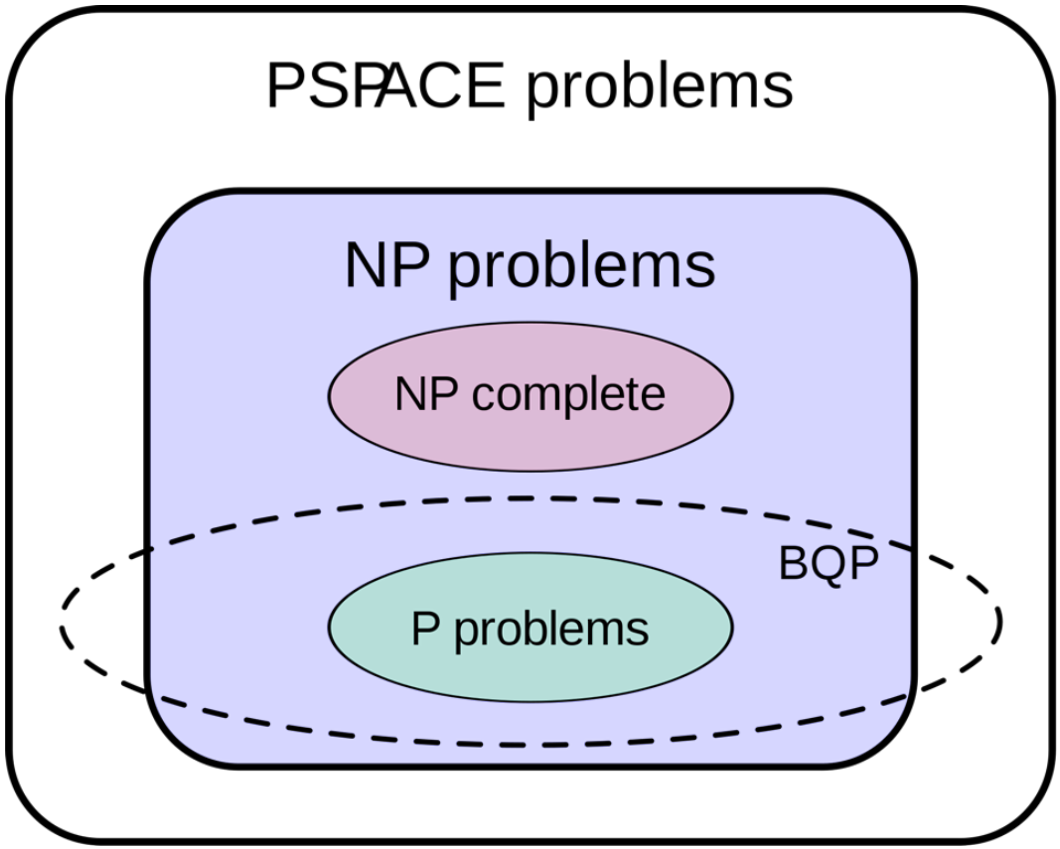






Calcul Quantique et complexité informatique

BQP (Bounded Quantum Probabilistic) est la classe d'algorithmes qui peuvent être résolus efficacement avec des algorithmes quantiques. La situation est supposée être comme ceci :



Michael Nielsen and Isaac Chuang (2000). Quantum Computation and Quantum Information. Cambridge : Cambridge University Press. (ISBN 0-521-63503-9).

Calcul Quantique et espace de caclul