Thèse de Church. Autres Modèles de Calculs

Olivier Bournez

November 23, 2009

1 Contribution

1.1 Thèse(s) de Church

Un des résultats fondamentaux les plus inattendus du vingtième siècle est le théorème d'incomplétude de Gödel, qui affirme qu'aucun système de preuve ne peut capturer pleinement le raisonnement mathématique: toute théorie suffisante pour capturer les raisonnements arithmétiques est nécessairement incomplète, c'est-à-dire telle qu'il existe des énoncés qui ne sont pas démontrables et dont la négation n'est pas non plus démontrable. En particulier, on peut exprimer la cohérence d'une théorie mathématique par un énoncé, qui ne peut être démontré, ou infirmé. Les arguments de Kurt Gödel dans l'article original [1] sont en fait très intimement basés sur une notion (informelle) de déduction algorithmique.

Alan Turing, travaillant sur le problème de la décision de Hilbert (*Entscheidungsproblem*, formulé ainsi par Turing: "peut-on décider mécaniquement si un énoncé est démontrable ou non") proposa dans l'article [52] son célèbre modèle de machine, capable de capturer la déduction dans les systèmes formels, et en particulier la notion de déduction utilisée par Gödel dans sa preuve.

Turing prouve dans [52] en fait schématiquement le résultat suivant: ce qui peut être calculé par un humain qui travaille mécaniquement avec un papier et un crayon en un nombre fini d'étapes est calculable par une machine de Turing [52], [18]. En particulier, cela couvre la déduction dans les systèmes formels, et la notion d'algorithme pour les systèmes programmables actuels.

En fait, Alonzo Church était arrivé quelques mois plus tôt dans [17] de façon indépendante à la même réponse négative au problème de la décision de Hilbert, en proposant le λ-calcul, et en argumentant que ce calcul capturait la notion de calcul effectif.

Très vite, on réalisa que la puissance des machines de Turing pouvait être prouvée égale à celle du formalisme du lambda-calcul de Alonzo Church [17], et à d'autres formalismes comme les fonctions récursives de Kurt Gödel, étendues par la suite par Stephen Kleene [35]. Ces considérations ont mené naissance à la thèse de Church-Turing, exprimée en fait pour la première fois par Stephen Kleene, étudiant de Alonzo Church. Cette thèse affirme que "ce qui est effectivement *calculable* est calculable par une machine de Turing." Dans cette formulation, la première notion de *calculable* fait référence à une notion donnée intuitive, alors que la seconde notion de calculable signifie "calculable par une machine de Turing" [26], [18], [40].

Les discussions originales de Church, Kleene et Turing sont relatives à des dispositifs algorithmiques de déduction, si l'on veut à la puissance de la déduction formelle, ou d'algorithmes, et non pas à la notion de machine ou de dispositif physique de calcul. Comme l'argumente très justement Jack Copeland dans [18], cette thèse a subi un glissement sémantique historique qui mène souvent à la confondre maintenant avec la thèse suivante, nommée thèse M dans [26], et parfois nommée *version physique de la thèse de Church*: "Ce qui peut être calculé par une machine est calculable par une machine de Turing".

Dans cette dernière, la notion de machine fait référence à une notion intuitive de machine, avec la contrainte que la machine est supposée obéir aux lois physiques[1] du monde réel [18]. Sans ces hypothèses, la thèse est facile à contredire: voir par exemple tous les contre-exemples dans [40] ou dans [19].

Observons que cette variante de la thèse est intimement reliée au problème de la modélisation de notre monde physique, c'est-à-dire à comprendre si les modèles de notre monde physique sont correctement reliés à ce dernier.

En fait, une variante, proche de la dernière thèse est la suivante: "Tout processus qui admet une description mathématique peut être simulé par une machine de Turing" [18]. Encore une fois (et pour globalement les même contre-exemples en fait que pour la thèse physique), si le processus n'est pas contraint de satisfaire les contraintes physiques de notre monde réel la thèse est facile à contredire [18].

Ces trois thèses sont indépendantes: la première concerne la puissance des systèmes formels, comme les systèmes de déduction ou de preuve [18]; la seconde concerne la physique du monde qui nous entoure [50], [18], [56]; la troisième concerne les modèles que nous avons du monde qui nous entoure [50], [18], [56].

Chacune de ces thèses, faisant référence à une notion intuitive, ne saurait être complètement prouvée.

Il peut toutefois être intéressant de discuter ces thèses.

D'une part, il est possible de chercher à définir un certain nombre d'axiomes minimaux que doivent satisfaire un système formel, ou une machine physique pour rendre ces thèses prouvables. Cela permet de réduire ces thèses à des hypothèses minimales sur les systèmes considérés, et peut aider à se convaincre de leur validité [26], [22, 11].

D'autre part, si l'on prend chacune de ces thèses de façon contraposée, chacune signifie que tout système qui calcule quelque chose qui ne l'est pas par une machine de Turing doit utiliser quelque chose, que nous appellerons *ressource*, qui soit n'est pas calculable algorithmiquement, pour la première, soit n'est pas

[1] Sinon à ses contraintes sur les ressources.

calculable par une machine physique, pour la seconde, soit n'est pas calculable par un modèle de machine physique, pour la troisième. Dans tous les cas, on peut qualifier (si besoin par définition) une telle ressource de "non-raisonnable". Il peut alors sembler important de discuter ce qui fait qu'une ressource peut être non raisonnable, indépendamment de la véracité de chacune des thèses, pour mieux comprendre le monde et les modèles du monde qui nous entourent.

Enfin, on peut observer que ces thèses expriment des faits très profonds sur la possibilité de décrire par les mathématiques ou par la logique le monde qui nous entoure [24], et plus globalement sur les liens entre calculabilité, mathématiques et physique: la nature calcule t'elle? quels sont les liens entre non-déterminisme, chaos, non-prédictibilité, et aléatoire [25]?

1.2 Des pistes pour des hyper-calculs

Aussi, il peut être intéressant de suivre [40], et de faire un rapide panorama (non-exhaustif) de différents moyens d'obtenir des systèmes ou des machines plus puissants que les machines de Turing.

La première façon, proposée par Turing lui-même dans [2] consiste à considérer des machines avec oracles, qui correspondent à des boites noires capables de répondre à certaines questions. Cela permet classiquement de "relativiser" nombre de résultats en calculabilité et complexité, mais aussi d'obtenir des modèles plus puissants que les machines de Turing (dès que l'oracle n'est pas calculable).

Il y a plusieurs types de machines ou de systèmes dans l'esprit des machines à oracles. Cela inclue les machines de Turing couplées de [20], qui sont des machines de Turing connectées à un canal d'entrée, ou les réseaux de machines qui exploitent le fait qu'un temps de synchronisation peut être irrationnel et non-calculable [19], ou les machines couplées à des dispositifs physiques comme les *scatter-machines* de [6], ou utilisant des tirages aléatoires biaisés comme dans [39].

Une seconde approche consiste à considérer des machines accélérantes, c'est-à-dire à considérer des machines qui effectuent leur première opération en temps unitaire, et chaque opération ultérieure en un temps égal à la moitié de l'opération précédente. Cette idée, ancienne, déjà présente dans [7], [41] et [55], est aussi présente dans les machines à temps ordinal: voir par exemple [33]. Il a été argumenté que différents systèmes physiques pouvaient en réaliser des implémentations: voir par exemple [16] à propos de la mécanique quantique, ou [34] à propos de calculs par trous noirs.

Il est aussi possible de considérer différentes variantes des machines de Turing: des machines exploitant un non-déterminisme non borné [51], travaillant sur des entrées infinies (voir par exemple [54] pour une présentation des machines de Turing de type 2), des machines avec une infinité d'états [38], des machines bruitées [5], etc...

Une autre façon consiste à considérer des machines analogiques, travaillant sur les réels. Nous y reviendrons.

1.3 Axiomatisations

Robin Gandy propose dans [26] de réduire la thèse physique de Church à quatre principes. L'intérêt est alors que l'on peut prouver mathématiquement que ce qui est calculable par un dispositif physique satisfaisant à ces quatre principes est calculable par machine de Turing. Autrement dit, Gandy propose de remplacer la thèse physique de Church par la thèse qu'un système mécanique physique déterministe discret satisfait à ces quatre principes basiques.

En suivant [24], on peut dire que l'argumentation de Gandy, qui suppose que l'espace physique est l'espace géométrique ordinaire à trois dimensions, repose sur deux grandes hypothèses sur la nature physique: la finitude de la densité de l'information et la finitude de la vitesse de la transmission de l'information.

La discussion de Gandy écarte explicitement les systèmes analogiques dès son début. Elle se limite aussi aux systèmes déterministes, mais peut s'étendre aux systèmes non-déterministes assez facilement. Par contre, les deux hypothèses plus haut reposent sur des hypothèses physiques, et ne sont pas nécessairement vraies dans toute théorie physique. Il peut alors être intéressant de comprendre ce qui se passe par exemple en mécanique quantique, ou en physique relativiste. Pour la physique quantique, nous renvoyons ainsi à [37] pour une discussion des sources de non-calculabilité, et les conséquences à en tirer sur la thèse de Church ou sur nos modèles physiques. L'article [4] discute par ailleurs de principes à la Gandy qui permettraient de capturer la théorie des calculs quantiques.

L'axiomatisation de Gandy se place au niveau de la thèse physique de Church, et utilise un formalisme mathématique assez particulier: les ensembles héréditairement finis. Elle a été simplifiée et étendue ultérieurement par différents travaux de Wilfried Sieg: voir par exemple [45, 43, 44, 46].

Pour la thèse originale de Church-Turing, Nachum Dershowitz et Yuri Gurevich ont proposé récemment une axiomatisation basée sur le formalisme beaucoup plus universel de la logique du premier ordre. Leur axiomatisation, qui fait suite aux travaux antérieurs de Udi Boker et Nachum Dershowitz dans [11], se repose sur l'axiomatisation de la notion d'algorithme offerte par les machines abstraites à états de Yuri Gurevich [32]. Nachum Dershowitz et Yuri Gurevich prouvent ainsi dans [22] que tout processus calculatoire qui vérifie les quatre axiomes suivants vérifie la thèse de Church. i) Un algorithme détermine une suite "d'états de calcul" pour chaque entrée valide; ii) les états d'une suite d'états de calcul sont des structures. Les descriptions sont à isomorphisme près; iii) les transitions d'état à état dans une suite d'états de calcul sont gouvernées par une description finie et fixée; iv) seulement des opérations considérées comme indéniablement calculables sont disponibles dans les états initiaux.

Les trois premiers axiomes définissent en fait la notion d'algorithme. Cette axiomatisation est à la base de la théorie des machines abstraites à états de Yuri Gurevich [32]. Il est possible de l'étendre à la notion d'algorithme parallèle [8], et de la reliée à la notion d'algorithme quantique [29].

La beauté de l'axiomatisation de [22] est d'être très générique, formelle, et basée sur des formalismes communément admis. D'autre part, elle permet de réduire prouvablement tout doute sur sa validité en doute sur l'un de ces quatre principes élémentaires.

1.4 Calculs sur les réels

Il existe différents modèles pour capturer les calculs sur les réels. Ces modèles sont issus de motivations très différentes: l'analyse récursive, née de Turing [52], Grzegorczyk [31], et Lacombe [36], considère qu'un réel est calculable s'il est possible d'en produire algorithmiquement une représentation, c'est-à-dire une suite convergente (rapidement) vers ce réel. Une fonction est calculable s'il est possible de produire algorithmiquement une représentation de son image, à partir de toute représentation de son argument.

Le modèle de Blum Shub et Smale [9, 10] a été introduit comme un modèle pour discuter de la complexité algébrique de problèmes sur les réels. Dans celui-ci, on se fixe des opérations arithmétiques réalisables en temps unitaire, et la complexité d'un problème est mesurée en termes du nombre d'opérations.

Le General Purpose Analog Computer a été introduit par Claude Shannon [42] comme un modèle des machines analogiques de son époque, en particulier de l'Analyseur Différentiel de Vannevar Bush [15]. Il demeure un modèle fidèle de ce qui est calculable par circuits analogiques comme les circuits électroniques [27].

Nous pouvons aussi citer les modèles issus des réseaux de neurones formels [48]. Nous renvoyons à [12] pour une synthèse des différents modèles de calcul à temps continu, et sur leurs propriétés du point de vue de la calculabilité ou de la complexité.

En particulier, dans plusieurs de ces modèles, il est possible d'exploiter le fait que l'on peut coder une suite (non-calculable) dans un réel, ou utiliser le fait que l'on suppose une arithmétique exacte pour réaliser des calculs super-Turing: voir par exemple [47] pour les réseaux de neurones.

Il est clairement impossible d'unifier toutes les approches: par exemple toute fonction calculable en analyse récursive est nécessairement continue, alors que des fonctions discontinues sont calculables dans le modèle initial de Blum Shub et Smale [9]. Cependant, si l'on met de coté l'approche algébrique du modèle de Blum Shub et Smale, certains résultats récents établissent l'équivalence entre différents modèles (voir par exemple [13, 14, 28]). Il n'est pas cependant clair à ce jour qu'il puisse exister un concept unificateur pour les modèles continus analogue à la thèse de Church pour les modèles discrets/digitaux: nous renvoyons à la synthèse [12] pour des discussions sur le sujet.

1.5 De la calculabilité à la complexité

On trouve parfois exprimées des versions renforcées des thèses précédentes, qui ne se placent pas uniquement au niveau de la calculabilité, mais aussi de la théorie de la complexité. Par exemple, la monographie [3], exprime l'existence d'une *thèse forte de la thèse de Church*: "tout modèle de calcul physiquement réalisable peut être simulé par une machine de Turing au prix d'un surcoût polynomial".

Ces versions sont sujettes à plus de controverses que les autres variantes. Ne serait-ce parce que tous les modèles historiques de calcul ne la vérifie pas: par exemple, il n'est pas facile de parler de complexité en λ-calcul. En fait, pour illustrer certaines des difficultés, remarquons qu'elle semble avoir été mentionnée pour la première fois dans [49] sous la forme suivante: "il existe une classe standard de modèles de machines, qui inclue entre autre toutes les variantes des machines de Turing, les variantes de RAM et RASP avec une mesure de temps logarithmique, et les RAM et RASP avec une mesure de temps uniforme, avec la condition que l'on se limite aux opérations arithmétiques standards (Addition, Soustraction, Incrément, Décrément, Test à zéro). Les machines de cette classe se simulent chacune avec un surcoût au plus polynomial, et un facteur constant en espace".

Chacune des discussions précédentes peut alors être interprétée au niveau de la complexité. Par exemple, dans [53], les auteurs se demandent si les machines analogiques peuvent être plus efficaces que les machines digitales, et prouvent, au prix de simplifications discutables, que la thèse de Church forte est prouvable pour les systèmes dynamiques à temps continu.

Observons que pour les systèmes quantiques, il est généralement admis qu'au niveau de la calculabilité, on ne peut calculer plus qu'avec une machine de Turing [23]. Par contre, au niveau de la complexité, les algorithmes quantiques permettent prouvablement de résoudre certains problèmes plus rapidement [30].

2 Remerciements

Olivier Bournez souhaite remercier Pablo Arrighi, Nachum Dershowitz, Gilles Dowek et Yuri Gurevich pour leurs relectures, réactions et commentaires pertinants.

References

[1] Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatschefte fur Mathematik und Physik*, 38:173–198, 1931. Traduction anglaise dans [21].

[2] M. Alan. Turing. Systems of logic based on ordinals. *Proceedings of the London Mathematical Society*, 45:161–228, 1939.

[3] S. Arora and B. Barak. *Computational Complexity: A Modern Approach.* Cambridge University Press, 2009.

[4] Pablo Arrighi and Gilles Dowek. A quantum extension of gandy's theorem,. Colloque "La thèse de Church: hier, aujourd'hui, demain",, Avril 2008.

[5] Eugene Asarin and Pieter Collins. Noisy Turing machines. In Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung,

editors, *International Colloquium on Automata, Languages and Programming (ICALP 2005)*, volume 3580 of *Lecture Notes in Computer Science*, pages 1031–1042, 2005.

[6] E. Beggs, J.F. Costa, B. Loff, and J.V. Tucker. Computational complexity with experiments as oracles. *Proceedings of the Royal Society of London-A*, 464(2098):2777, 2008.

[7] RM Blake. The paradox of temporal process. *The Journal of Philosophy*, 23(24):645–654, 1926.

[8] A. Blass and Y. Gurevich. Abstract state machines capture parallel algorithms. *ACM Transactions on Computational Logic (TOCL)*, 4(4):578–651, 2003.

[9] L. Blum, M. Shub, and S. Smale. On a theory of computation and complexity over the real numbers; NP completeness, recursive functions and universal machines. *Bulletin of the American Mathematical Society*, 21(1):1–46, July 1989.

[10] Lenore Blum, Felipe Cucker, Michael Shub, and Steve Smale. *Complexity and Real Computation*. Springer-Verlag, 1998.

[11] Udi Boker and Nachum Dershowitz. The Church-Turing thesis over arbitrary domains. In Arnon Avron, Nachum Dershowitz, and Alexander Rabinovich, editors, *Pillars of Computer Science: Essays Dedicated to Boris (Boaz) Trakhtenbrot on the Occasion of His 85th Birthday*, volume 4800, pages 199–229. Springer, 2008.

[12] Olivier Bournez and Manuel L. Campagnolo. *New Computational Paradigms. Changing Conceptions of What is Computable*, chapter A Survey on Continuous Time Computations, pages 383–423. Springer-Verlag, New York, 2008.

[13] Olivier Bournez, Manuel L. Campagnolo, Daniel S. Graça, and Emmanuel Hainry. The general purpose analog computer and computable analysis are two equivalent paradigms of analog computation. In Jin-yi Cai, S. Barry Cooper, and Angsheng Li, editors, *Theory and Applications of Models of Computation, Third International Conference, TAMC 2006, Beijing, China, May 15-20, 2006, Proceedings*, volume 3959 of *Lecture Notes in Computer Science*, pages 631–643. Springer, 2006.

[14] Olivier Bournez, Manuel L. Campagnolo, Daniel S. Graça, and Emmanuel Hainry. Polynomial differential equations compute all real computable functions on computable compact intervals. *Journal of Complexity*, 23(3):317–335, June 2007.

[15] V. Bush. The differential analyzer. A new machine for solving differential equations. *J. Franklin Inst.*, 212:447–488, 1931.

[16] C. S. Calude and B. Pavlov. Coins, quantum measurements, and Turing's barrier. *Quantum Information Processing*, 1(1-2):107–127, April 2002.

[17] Alonzo Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics,*, 58:345–363, 1936. Reprinted in [21].

[18] B. Jack Copeland. The Church-Turing thesis. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy.* Stanford University, Fall 2002. Available online at: *http://plato.stanford.edu/entries/church-turing/.*

[19] B. Jack Copeland and Richard Sylvan. Beyond the universal Turing machine. *Australasian Journal of Philosophy*, 77:46–66, January 27 1999.

[20] B.J. Copeland. The broad conception of computation. *American Behavioural Scientist*, 40:690–716, 1997.

[21] Martin Davis. *The Undecidable: Basic Papers on Undecidable Propositions, Unsolvable Problems and Computable Functions.* Raven Press, 1965. Reprinted by Dover Publications, Incorporated in 2004.

[22] N. Dershowitz and Y. Gurevich. A natural axiomatization of computability and proof of Church's Thesis. *The Bulletin of Symbolic Logic*, 14(3):299–350, 2008.

[23] D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proc. R. Soc. Lond. Ser. A*, A400:97–117, 1985.

[24] G. Dowek. *Les métamorphoses du calcul: une étonnante histoire de mathématiques.* Le Pommier, 2007.

[25] Giuseppe Longo et Thierry Paul. Le monde et le calcul. Réflexions sur calculabilité, mathématiques et physique.

[26] Robin Gandy. Church's thesis and principles for mechanisms. *The Kleene Symposium*, pages 123–148, 1980.

[27] Daniel S. Graça. *Computability with Polynomial Differential Equations.* PhD thesis, Instituto Superior Técnico, 2007.

[28] Daniel S. Graça and José Félix Costa. Analog computers and recursive functions over the reals. *Journal of Complexity*, 19(5):644–664, 2003.

[29] E. Gradel and A. Nowack. Quantum computing and abstract state machines. *Lecture Notes in Computer Science*, pages 309–323, 2003.

[30] L.K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, page 219. ACM, 1996.

[31] A. Grzegorczyk. On the definitions of computable real continuous functions. *Fundamenta Mathematicae*, 44:61–71, 1957.

[32] Gurevich. Sequential abstract-state machines capture sequential algorithms. *ACMTCL: ACM Transactions on Computational Logic*, 1, 2000.

[33] J.D. Hamkins. Infinite time Turing machines. *Minds and Machines*, 12(4):521–539, 2002.

[34] Mark Hogarth. Non-Turing computers and non-Turing computability. *Philosophy of Science Association*, i:126–138, 1994.

[35] Stephen C. Kleene. General recursive functions of natural numbers. *Mathematical Annals*, 112:727–742, 1936. Reprinted in [21].

[36] D. Lacombe. Extension de la notion de fonction récursive aux fonctions d'une ou plusieurs variables réelles III. *Comptes Rendus de l'Académie des Sciences Paris*, 241:151–153, 1955.

[37] MA Nielsen. Computable functions, quantum measurements, and quantum dynamics. *Physical Review Letters*, 79(15):2915–2918, 1997.

[38] T. Ord. Hypercomputation: computing more than the Turing machine. *Arxiv preprint math/0209332*, 2002.

[39] T. Ord and T.D. Kieu. Using biased coins as oracles. *Arxiv preprint cs/0401019*, 2004.

[40] Toby Ord. The many forms of hypercomputation. *Applied Mathematics and Computation*, 178(1):143–153, 2006.

[41] B. Russell. The limits of empiricism. In *Proceedings of the Aristotelian Society*, volume 36, pages 131–150. Blackwell Publishing; The Aristotelian Society, 1935.

[42] C. E. Shannon. Mathematical theory of the differential analyser. *Journal of Mathematics and Physics MIT*, 20:337–354, 1941.

[43] W. Sieg. Mechanical procedures and mathematical experience. *Mathematics and mind*, pages 71–117, 1994.

[44] W. Sieg. Step by recursive step: Church's analysis of effective calculability. *The Bulletin of Symbolic Logic*, 3(2):154–180, 1997.

[45] W. Sieg. Hilbert's programs: 1917-1922. *The Bulletin of Symbolic Logic*, 5(1):1–44, 1999.

[46] Wilfried Sieg. *New Computational Paradigms. Changing Conceptions of What is Computable*, chapter Church without dogma–Axioms for computability. Springer, New York, 2008.

[47] Hava T. Siegelmann and Shmuel Fishman. Analog computation with dynamical systems. *Physica D*, 120:214–235, 1998.

[48] Hava T. Siegelmann and Eduardo D. Sontag. On the computational power of neural nets. *Journal of Computer and System Sciences*, 50(1):132–150, February 1995.

[49] C. Slot and P. van Emde Boas. On tape versus core an application of space efficient perfect hash functions to the invariance of space. In ACM, editor, *Proceedings of the sixteenth annual ACM Symposium on Theory of Computing, Washington, DC, April 30–May 2, 1984*, pages 391–400, New York, NY, USA, 1984. ACM Press.

[50] Warren D. Smith. History of "Church's theses" and a manifesto on converting physics into a rigorous algorithmic discipline. Technical report, NEC Research Institute, 1999. Avalaible on *http://www.math.temple.edu/ wds/homepage/works.html*.

[51] E. Spaan, L. Torenvliet, and P.E. Boas. Nondeterminism fairness and a fundamental analogy. *Bulletin of the EATCS*, 37:186–193, 1989.

[52] Alan Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42(2):230–265, 1936. Reprinted in [21].

[53] Anastasios Vergis, Kenneth Steiglitz, and Bradley Dickinson. The complexity of analog computation. *Mathematics and Computers in Simulation*, 28(2):91–113, 1986.

[54] K. Weihrauch. *Computable Analysis: an Introduction*. Springer, 2000.

[55] H. Weyl and G. Kirschmer. *Philosophie der Mathematik und Naturwissenschaft*. Oldenbourg Wissenschaftsverlag, 1927.

[56] Andrew Chi-Chih Yao. Classical physics and the Church–Turing Thesis. *Journal of the ACM*, 50(1):100–105, January 2003.