Working Paper #004/2019

"Controlling Algorithmic Collusion: Short Review of the Literature, Undecidability, and Alternative Approaches"

João E. Gata[1]

Abstract

Algorithms have played an increasingly important role in economic activity, as they becoming faster and smarter. Together with the increasing use of ever larger data sets, they may lead to significant changes in the way markets work. These developments have been raising concerns not only over the rights to privacy and consumers' autonomy, but also on competition. Infringements of antitrust laws involving the use of algorithms have occurred in the past. However, current concerns are of a different nature as they relate to the role algorithms can play as facilitators of collusive behavior in repeated games, and the role increasingly sophisticated algorithms can play as autonomous implementers of pricing strategies, learning to collude without any explicit instructions provided by human agents. In particular, it is recognized that the use of 'learning algorithms' can facilitate tacit collusion and lead to an increased blurring of borders between tacit and explicit collusion. Several authors who have addressed the possibilities for achieving tacit collusion equilibrium outcomes by algorithms interacting autonomously, have also considered some form of ex-ante assessment and regulation over the type of algorithms used by firms. By using well-known results in the theory of computation, I show that such option faces serious challenges to its effectiveness due to undecidability results. Ex-post assessment may be constrained as well. Notwithstanding several challenges face by current software testing methodologies, competition law enforcement and policy have much to gain from an interdisciplinary collaboration with computer science and mathematics.

---

[1] Principal Advisor, Cabinet of the President, Portuguese Competition Authority. E-mail: jgata@concorrencia.pt.
Ph.D. in Economics, University of Minnesota/USA, and Post-Graduation in EU Competition Law, King's College, London/UK. I thank my colleagues Miguel Sousa Ferro, João S. Lopes, Javier Garcia-Verdugo, Mariana Tavares and Pierluigi Sabbatini, for their comments and suggestions.
This paper is my sole responsibility and represents my own views. Its contents do not necessarily represent the opinions and views of third entities, including the Portuguese Competition Authority, unless expressly noted.

I. Introduction

Algorithms have played an increasingly important role in economic agents' decision making and in economic activity in general[2]. They have been present in some markets and industries for a long time: in finance and banking, insurance, airlines, food retail, e-commerce, and in many other industries. However, their use is becoming widespread, especially across more advanced economies, namely in e-commerce and in the definition and implementation of pricing strategies in general, as algorithmic pricing software becomes increasingly affordable even to smaller businesses[3]. Moreover, algorithms are becoming faster and smarter which, together with the increasing use of ever larger data sets by market participants, may lead to significant changes in the way markets work[4].

The use of algorithms by suppliers and consumers can both benefit and hurt consumer welfare[5]. For example, by reducing information and transactions costs, e.g., by making widely available product comparison sites, they allow disposable income to go further, enabling more consumers to consume more and make better choices. On the other hand, through the use of ever increasing big data sets on individual consumers' habits and behavioral patterns, algorithms can increase the scope for personalized pricing by suppliers, leading to a larger appropriation of consumer surplus, even if such personalized pricing may render price collusion less attractive and more difficult to sustain in equilibrium.

These developments have been raising some concerns not only over the rights to privacy and consumers' autonomy, but also over the way markets work and on the level of competition they can sustain.

Infringements of antitrust laws involving the use of algorithms have occurred in the past. As an example, recall the Airline Tariff Publishing Case, dealt with by the US Justice Department and settled with a consent decree in March 1994, eight major US airlines colluded to raise prices and restrict competition in the airline industry. Collusion was sustained through the transmission of relevant information via the Airline Tariff Publishing Company (ATPCO), such as information on 'first and last ticket dates' and on 'first and last travel dates'[6].

---

2   An 'algorithm' can be defined as a finite sequence of instructions, expressed in a precise based upon a certain alphabet, such that, when confronted with a question of some kind and carried out in the most literal- minded way, will invariably terminate, sooner or later, with the correct answer. Notice that this does not constitute a mathematical definition. There is no agreed upon mathematical definition of 'algorithm'. Sometimes the term 'effective procedure' is used instead of the term 'algorithm'.

3   Amazon, Google, Microsoft and other companies supply off-the-shelf machine learning solutions and computing capability - see, E. Calvano et al. (2018a).

4   As pointed out by Petit (2017), a common thread to the emerging literature on Antitrust and Artificial Intelligence Literature (AAI), is to describe the increasing use of algorithms on markets as a 'game changer'.

5   On the use of algorithms by consumers, and its effects on markets and welfare, see Gal & Elkin-Koren (2017).

6   As reported in the March 18th, 1994, issue of The New York Times, «*Anne K. Bingaman, Assistant Attorney General in charge of the antitrust division, said the airlines used the Airline Tariff Publishing system "to carry on conversations just as direct and detailed as those traditionally conducted by conspirators over the telephone or in hotel rooms. Although their method was novel, their conduct amounted to price-fixing, plain and simple," she said*». According to S. Borenstein (2004), «*While an agreement among competitors to fix prices is per se illegal, computer technology that permits rapid announcements and responses has blurred the meaning of "agreement" and has made it difficult for antitrust authorities to distinguish public announcements from conversations among competitors*» - See also J. Klein (1999). The Airline Tariff Publishing system is the property of the Airline Tariff Publishing Company (ATPCO) a corporation owned by several airlines, formed to serve as agent for those owners (and for other airlines or vendors) to file and publish tariffs and related products. It operates as a clearinghouse for distribution of fare change information. At least once a day ATPCO produces a compilation of

In the more recent *US v. David Topkins Case*, a US District Court ruled that Title 15, US Code, Section 1, had been violated. David Topkins apparently had coded an algorithm that enabled him and his co-conspirators to agree to fix the prices of certain posters sold in the US through Amazon Marketplace[7].

In December 2015, a UK citizen was indicted for an allegedly similar price fixing strategy applied to posters sold through the online site Amazon Marketplace. The indictment, unsealed on December 4th and originally filed in the Northern District of California on August 27th, 2015, names the UK citizen Daniel Aston and his company named 'Trod', doing business as 'Buy 4 Less', as conspiring to fix prices for online posters sales from September 2013 to January 2014. According to Aston's indictment, he used commercially available algorithm-based pricing software to fix the prices of posters sold on Amazon Marketplace[8].

In the *Eturas Case* (Case C-74/14), the European Court of Justice (CJEU) dealt with concerted practices between travel agents through the use of an online platform. The alleged coordination would have taken place via an online travel booking system (E-turas, owned by Eturas) used by more than 30 travel agents in Lithuania. The Lithuanian Competition Council (LCC) imposed fines on Eturas and these 30 travel agencies for applying a common cap on discounts applicable to services provided through the Eturas online booking platform. The discount cap was communicated to the agencies through an internal messaging system in the form of an amendment to the platform terms and conditions. It was then implemented by Eturas using technical means[9].

Online platforms may facilitate an unlawful cooperation between platform users without involving their direct contact. These examples may involve the use of 'adaptive algorithms', to borrow a terminology used in E. Calvano et al. (2018a)[10].

The greatest concerns recently expressed in the literature on competition law and policy relate: (i) to the role algorithms in general can play as facilitators of collusive behavior in repeated games[11], and (ii) the role increasingly sophisticated algorithms can come to play as autonomous implementers of pricing strategies which, through their strategic interaction and

---

all industry fare change information and sends the computer file, containing thousands of fare changes, to a list of recipients that includes all major airlines and the computer reservations systems operating in the US – see S. Borenstein.

[7] Ruling by the US District Court, Northern District of California, San Francisco Division, April 2014.

[8] For this case involving Daniel Aston, see MLex, 26 Feb 2016.

[9] See http://competitionlawblog.kluwercompetitionlaw.com/2017/01/19/eturas-conclusions-platform-collusion/. The authors of this blog rightly claim that «(…) *the Eturas decision demonstrates how information technology can distort markets in the digital space.*»

[10] As described by Calvano et al. (2018a), an 'adaptive algorithm' incorporates a model of the market and seek to maximize the firm's profit. An example is provided by 'dynamic pricing for revenue management', used in hotel booking and airline services. It may estimate market demand using data on sold quantities and prices, and then will estimate the optimal price given the estimated demand and the firm's rivals past behavior (prices charged, etc).

[11] See Mehra (2016) for a thorough discussion of the role played by "robo-sellers". The author adds that «*The rise of the robo-seller exacerbates antitrust law's longstanding weakness at addressing social harm from oligopoly. Black-letter law's blind spot when it comes to independent price coordination—that is, without overt acts such as communication or the adoption of facilitating practices—may become a cloaking device behind which algorithmic price coordination can readily hide. Additionally, the challenges that face explicit collusion by oligopolists may become easier to surmount with mass data collection and algorithmic assistance.*» I believe this assertion calls for further work. As stated by McSweeny (2017), «*We have a lot to learn about the effects of pricing algorithms and AI. Further research will contribute to better and more effective competition enforcement in this area.*»

their access to large data sets (made up of market information on prices, sales, and other relevant variables for the definition and implementation of pricing strategies), become capable of adapting over time and of learning to collude without any explicit instructions provided by human agents[12] [13]. In particular, it is recognized that the use of 'learning algorithms' can facilitate tacit collusion and lead to an increased blurring of borders between tacit and explicit collusion[14], as they are not programmed with the intent of converging to a collusive equilibrium outcome but can reach it through learning and intelligent adaptation[15]. Moreover, their choices, namely of pricing strategies, and subsequent implementation, would be unencumbered by moral and ethical considerations and constraints, or other human behavioral "biases", unlike the case with human economic agents[16]. In this sense, the evolution of artificial intelligence is bringing once more to the fore the persistently controversial distinction between tacit and explicit collusion. A distinction that may be of little consequence in economic theory but is hugely relevant in competition law and policy[17].

At least part of the competition law and economics community recognizes this challenge, to the point of questioning whether the existing antitrust regimes across many jurisdictions are

---

[12] Competition issues raised by this type of algorithms have been analyzed by Ezrachi & Stucke (2015), namely under the categories of collusion they call "Predictable Agent" and "Autonomous Machine".

[13] Quoting the OECD Report (2017) «*Artificial intelligence [AI] refers to the broad branch of computer science that studies and designs intelligent agents, who should be able to carry out tasks of significant difficulty in a way that is perceived as "intelligent". (…) At the initial stages of AI, machines were programmed with extensive lists of detailed rules in order to attempt to replicate human thoughts, which could easily become a burdensome process. AI became a more effective tool after the development of algorithms that teach machines to learn, an idea that evolved from the study of pattern recognition and learning theory, and which would establish the new branch of machine learning. Machine learning (ML) is a subfield of AI which designs intelligent machines through the use of algorithms that iteratively learn from data and experience. (…) machine learning gives "computers the ability to learn without being explicitly programmed". Machine learning algorithms can be classified into three broad categories, depending on their learning pattern (Anitha et al., 2014): (i) Supervised learning, where the algorithm uses a sample of labelled data to learn a general rule that maps inputs to outputs; (ii) Unsupervised learning, where the algorithm attempts to identify hidden structures and patterns from unlabeled data; (iii) Reinforcement learning, where the algorithm performs a task in a dynamic environment, such as driving a vehicle or playing a game (…) and learns through trial and error.*»

[14] Following OECD (2017): «*Economists usually distinguish between two forms of collusion, explicit and tacit. Explicit collusion refers to anti-competitive conducts that are maintained with explicit agreements, whether they are written or oral. The most direct way for firms to achieve an explicit collusive outcome is to interact directly and agree on the optimal level of price or output. Tacit collusion, on the contrary, refers to forms of anti-competitive co-ordination which can be achieved without any need for an explicit agreement, but which competitors are able to maintain by recognizing their mutual interdependence. In a tacitly collusive context, the non-competitive outcome is achieved by each participant deciding its own profit-maximizing strategy independently of its competitors. (…) Contrary to the economic approach, which considers collusion a market outcome, the legal approach focuses on the means used by competitors to achieve such a collusive outcome. For this reason, competition laws generally do not prohibit collusion as such, but prohibit anti-competitive agreements. If collusion is the result of such as agreement then an infringement of the law can be successfully established. Although there is great variance in how jurisdictions interpret the notion of agreement, they traditionally require some sort of proof of direct or indirect contact showing that firms have not acted independently from each other (the so-called "meeting of the minds").*»* For an economic analysis of 'tacit collusion', see Ivaldi et al. (2003). See also Kaplow (2011).

[15] See T. Klein (2018) for an example of how autonomous 'Q-learning algorithms' are able to achieve supra-competitive profits in a stylized oligopoly environment with sequential price competition. See also Calvano et al. (2018a).

[16] Unless and until algorithms may be so sophisticated as to take into account ethical and moral considerations. In any case, it is certainly already possible to program algorithms to take into consideration certain simple ethical rules such as "split the gain in half".

[17] See e.g., Kaplow (2011).

capable to meet it[18]. For example, it has been pointed out that antitrust legislation was drafted having human agents in mind. Concepts such as "meeting of the minds", "mutual understanding", "mutual assent", "concurrence of wills"[19], can hardly be applied to the case of autonomous artificial agents[20], if they cannot be regarded as mere tools used by firms (otherwise firms would undoubtedly remain liable for their own collusive behavior), but behave as truly autonomous agents[21]. As pointed out by McSweeny (2017), «*Concerns about algorithmic tacit collusion are still largely theoretical at this point. Nonetheless, recent examples suggest that the concern is not fanciful.*»

The challenge goes beyond liability. For example, the EU can assert that «(…) *companies should ultimately be held responsible for the activities of any algorithm or pricing software they deploy»,* and that *«like an employee or an outside consultant working under a firm's 'direction or control,' an algorithm remains under the firm's control, and therefore the firm is liable for its actions*.»[22] But that does not prevent that the use of increasingly sophisticated algorithms will make collusion more difficult to detect and prosecute, even if the design and use of algorithms may be regarded as a 'plus factor' to 'an agreement' between firms employing such algorithms[23].

II. Algorithmic Collusion

The use of algorithms in game theory goes a long way back. In particular, several authors[24] have analyzed the play of non-cooperative games and their equilibria when finite automata play the game as models of rational players with limited memory and reasoning capacity. Such

---

[18] As referred by Kroll et al. (2017): « *(…) the accountability mechanisms and legal standards that govern decision processes have not kept pace with technology. The tools currently available to policymakers, legislators, and courts were developed primarily to oversee human decision makers. Many observers have argued that our current frameworks are not well-adapted for situations in which a potentially incorrect, unjustified, or unfair outcome emerges from a computer.* » - p. 636. However, in a recent intervention, M. Ohlhausen (2017), Acting Chairman, U.S. Federal Trade Commission, stated that from an antitrust perspective, the expanding use of algorithms raises familiar issues that are well within the existing canon (i.e., within the existing competition legislation and policy).

[19] See e.g., Kaplow (2011) for an analysis of some of these concepts and others, such as 'conspiracy', 'collusion', 'parallelism', and 'conscious parallelism'. Behind the concept of "meeting of the minds" is, or at least seems to be, the well-defined concept of "common knowledge" in game theory. About this latter concept, see e.g., Fudenberg & Tirole (1991). See Aumann (1976) for a mathematical definition of "common knowledge", starting with some *probability space* on the "states of the world", and using the topological notions of *join* and *meet* of sets. See also Lewis (2002) for a more philosophical discussion of this concept.

[20] The term 'Autonomous Artificial Agent' is used by Harrington and is defined as 'a software program that carries out a set of operations on behalf of a human agent without intervention by this agent, and does so with some knowledge of the human agent's objective' – see Harrington (2018), p. 7.

[21] See e.g., Mehra (2016), and Harrington (2018).

[22] See M. Vestager (2017).

[23] See Gal & Elkin-Koren (2017), pp. 346/7, albeit these two authors discuss 'plus factors' in the context of consumer algorithms. See also Gal (2018), p. 37. A 'plus factor' is to be understood as additional economic circumstantial evidence that, together with parallel conduct by different firms, e.g., a parallel movement in prices, can merit an investigation under antitrust legislation – see Kovacic et al. (2011). Such investigation is merited when all alternative explanations for such parallel movement, but for some sort of agreement between firms, are non-credible.

[24] See Rubinstein (1986, 1998), Gilboa (1988), and Kalai (1990), among other authors. See also Conlisk (1996).

capacity can be measured by the number of states in each finite automaton[25]. The study of algorithms as collusive devices has been somewhat more recent[26].

In a recent paper, Salcedo (2015) explores a symmetric dynamic model of price competition with two firms, where firms choose pricing algorithms (henceforth, PA's) simultaneously and independently at the beginning of the repeated game. He shows that when four conditions are met simultaneously, namely, firms set prices through algorithms that can respond to market conditions, these algorithms are fixed in the short run, can be decoded by the rival, and can be revised over time, then every equilibrium of the game leads in the long run to monopolistic, or collusive, profits. He claims his findings provide theoretical support for the idea that optimal use of PAs is an effective tool for tacit collusion and that the similar results will be attained in the context of general repeated games and not just a pricing duopoly.

In another recent paper, T. Klein (2018) shows how in a stylized duopoly environment with a homogeneous good, unrestricted production capacity, and with repeated sequential price competition, independent 'Q-learning algorithms' are able to achieve higher-than static prices and profits. According to the author, his result provides ground for competition authorities and regulators to remain vigilant when observing the rise of autonomous PAs in the marketplace, in particular in cases where firms may be short-run price committed. He also claims that the general framework used in the paper may be used to similarly assess the capacity of other, perhaps more advanced algorithms to collude in various environments.

In another paper, Calvano et al. (2018a) put forth five important questions, namely: (i) Can "smart" PAs learn to collude? (ii) Is collusion among algorithms any different from collusion among humans? (iii) In particular, is algorithmic pricing conducive to collusion more often than what humans could do? If the answers to these questions are affirmative, further issues will arise: (iv) How can we detect algorithmic collusion? (v) What are the appropriate new standards for competition policy?

In my paper I deal with questions (iv) e (v). If the answer to questions (i) and (ii) is a clear 'no', then there is not much to say that has not been said before. Hence, my paper is only relevant if we cannot give such a clear negative answer to both questions.

The authors also distinguish between 'adaptive algorithms' and 'learning algorithms', claiming that the serious challenges to current competition legislation and policy come from the latter. As referred in their paper, and contrary to 'adaptive algorithms', learning (pricing) algorithms are 'active learners', as they are 'willing' to adopt strategies that may be suboptimal so as to learn from experience. A learning algorithm "*learns to play optimally from experience*", which gives such algorithms an advantage over adaptive algorithms in more complex environments. This also allows them to reach a collusive equilibrium without being designed to do so. Through the simulation of a repeated game played by two 'Q-learning' algorithms, representing two competing firms playing a prisoner's dilemma game with strategies 'Price

---

[25] See annex for a formal definition of 'finite automaton', or Lewis & Papadimitriou (1981). The finite automata that implement pre-defined repeated game strategies are 'Moore machines', as their output is not necessarily binary – see annex. There are ways of measuring the complexity of Moore machines other than just their number of states. E.g., the Moore machine implementing the "grim trigger strategy" in the infinitely repeated prisoner's dilemma game can be regarded as simpler than the one implementing "tit-for-tat". In fact, both machines have two states but the transition function is simpler for the "grim trigger" strategy.

[26] See additional discussion in Schwalbe (2018).

High' and Price Low' and with a one-period memory[27], the authors show that even if it takes some time for Q-learning algorithms to 'realize' that collusion can be profitable, collusion will occur most of the time, even if some experimentation periods will occur when the algorithms engage in a "price war".

Both Klein and Salcedo employ in their work 'Q-learning algorithms'. These algorithms are examples of model-free active reinforcement learning agents[28]. They learn the value of taking a certain action a in a state s, where this value, called *Q-value*, is directly related to a payoff $U(\cdot)$, which is a function of state s[29]. However, in both papers the very large number of time periods required for 'Q-learning algorithms' to explore their environment and learn to adequately balance their dual functions of 'exploitation' (reaping benefits) and 'exploration' (learning), is typically far greater than the frequency with which firms interact in most markets and carry out effective price changes. It is possible that more sophisticated algorithms can learn more and learn faster[30]. Calvano et al. (2018a) discuss these possibilities as well, including communication between algorithms.

III. 'Preventing Algorithmic Collusion'

Several authors who have addressed the possibilities for achieving tacit collusion equilibrium outcomes by algorithms interacting autonomously from any instructions by human agents, have also opened the possibility for some form of *ex-ante* assessment and regulation over the type of algorithms being used by firms. As referred by Mehra (2016), «*Looking further into the future, regulators may need to develop the ability to test and probe the effects of algorithmic sales on consumers; agencies may need to conduct their own 'algorithmic enforcement'*»[31]. And, as already mentioned, to regard the use of algorithms, or some types of algorithms, as 'plus factors' to 'an agreement' between firms employing such algorithms[32], possibly in an *ex-post* evaluation.

Calvano et al. (2018a) distinguish three possible policy approaches to the risk of algorithmic collusion. A total ban on the use of algorithms is set rightfully aside is an unreasonable approach. The first approach takes 'business-as-usual', where algorithmic pricing is regarded as not posing any new problem that cannot be dealt with by current antitrust legislation. In particular, the legal distinction between tacit and explicit collusion is maintained, as attempting to sanction tacit collusion would remain subject to unreasonably high type I and II errors. The second approach calls for an *ex-ante* regulation, or supervision, of PAs, to be carried out by a regulatory (or competition) agency. This agency would have the power to

---

[27] The common time discount factor equals 0.995, the learning rate equals 0.15, and the experimentation rate is constant and equals 0.04.

[28] It is worth mentioning that reinforcement learning models have been developed in game theory for many years. They attempt to model the behavior of less than fully rational economic agents who interact strategically, and where equilibria, when they exist, arise as long-run outcomes of this interaction. Players learn to improve their strategic choices as they play the game period after period - see Fudenberg & Levine (1998).

[29] See Russell & Norvig (2016), chs. 17 and 21, for a more extensive discussion of "Q-learning algorithms".

[30] It is worth mentioning that in infinitely repeated (non-cooperative) games, the computation of best-response strategies may not be trivial, and can be quite complex – see e.g., Gilboa (1988), Ben Porath (1990), and Papadimitriou (1992).

[31] See Mehra (2016), p. 1331.

[32] See e.g., Harrington (2018); Calvano et al. (2018a); Gal (May 2017, and 2018); Mehra (2016); Ezrachi & Stucke (2016); T. Klein (2018).

prohibit certain PAs that exhibited a 'tendency to collude', a characteristic that would have to be defined in a precise and rational way, not least for the sake of legal certainty. As we will see, this second approach is also favored by Harrington (2018). The third approach calls for an *ex-post* regulation, or control, the same way competition agencies currently deal with antitrust practices, but under legal standards somewhat different from the current ones[33]. Perhaps these standards would take a more assertive, yet careful stance towards 'tacit collusion'. Calvano et al. (2018a) seem to favor this third approach, where the legal distinction between tacit and explicit collusion would have to be reassessed.

Harrington (2018) draws a distinction between the current legal doctrine on collusion by human agents and the situation where prices are set by autonomous artificial agents (AAs)[34]. In this latter case, the strategy determining the price to be charged is written down in the algorithm's code which means that it can, in principle, be accessed, contrary to the mind of a colluding manager. Based on this crucial distinction, this author proposes that liability be defined by a *per se* prohibition of certain PAs that support supra-competitive prices, so as to make collusion by AAs unlawful. Liability would be determined by an examination of a pricing algorithm's (PA) code to determine whether it is a prohibited PA, or by entering data into the PA and monitoring the output in terms of prices to determine whether the algorithm exhibits a prohibited property[35]. As stated, ideally the liability rule would prohibit all algorithms that promote collusion, and would exclude from such prohibition all algorithms that promote efficiency[36]. I.e., an ideal liability rule would allow the decision maker not to commit errors of type I and type II. A realistic alternative would be to design or choose a liability rule that would allow for the maximization of *Likelihood Ratio* $LR(PPA) \stackrel{\text{def}}{=} (1 - Prob[error\ type\ II]) \div Prob[error\ type\ I]$[37]. Harrington proceeds by drawing a broad outline of a three-step research program to identify which PAs will be prohibited, as follows:

Step 1: (1.i) Create a simulated market setting with learning algorithms that produce collusion and competition as outcomes; (1.ii) Keep track of when competitive prices emerge and when supra-competitive prices emerge; (1.iii) Perform this exercise with different learning algorithms and for a variety of market conditions.

---

[33] See Calvano et al. (2018a), pp. 14/15.

[34] In Harrington, an AA is composed of two elements: a '*pricing algorithm*' that prescribes what price to charge depending on the history of the (repeated) game played by the various firms competing in the same market; and a '*learning algorithm*' that chooses and modifies the pricing algorithm based on a pricing algorithm's performance relative to the performance of other pricing algorithms. When the co-domain, or at least the range, of a strategy in a game has more than two elements, then a FA implementing it is a *Moore machine*, a generalization of the basic FA – see annex for a definition of a *Moore machine*.

[35] See 'white-box settings' and 'black-box settings' in Desai & Kroll (2018).

[36] Following Harrington (2018), let *pa* denote a "pricing algorithm" and *PPA* denote the set of prohibited pricing algorithms. Given a specification of *PPA*, Prob [*pa* ∈ *PPA*|*pa* is collusive] is the probability that a pricing algorithm is determined to be in the prohibited set when the pricing algorithm is collusive. Prob [*pa* ∈ *PPA* |*pa* is competitive] is the probability that a pricing algorithm is determined to be in the prohibited set when the pricing algorithm is competitive. Ideally, Prob [*pa* ∈ *PPA*|*pa* is collusive] = 1 and Prob [*pa* ∈ *PPA* |*pa* is competitive] = 0 so that a pricing algorithm is concluded to be unlawful if and only if it is collusive. That is, ideally, errors type I and type II would be zero. That such an ideal is not reached will be due to misspecification of set *PPA* - some collusive pricing algorithms are excluded from *PPA* or some competitive pricing algorithms are included - or incomplete data or inadequate methods for evaluating whether a particular pricing algorithm is in *PPA*.

[37] Where $(1 - Prob[error\ type\ II]) = 1 - Prob[pa \notin PPA|pa$ is collusive], and $Prob[error\ type\ I] = Prob[pa \in PPA |pa$ is competitive], and where *PPA* denotes the collection of prohibited pricing algorithms.

Step 2: (2.i) Inspect or test the resulting pricing algorithms for the purpose of identifying those properties that are present when supra-competitive prices emerge but are not present when competitive prices emerge, (2.ii) Pricing algorithms with those properties will have a high likelihood ratio and thus be a candidate for the set of prohibited pricing algorithms.

Step 3: Test the effect of prohibiting a set of pricing algorithms. This would be done by re-running the learning algorithms in the simulated market setting but where the learning algorithms are constrained not to select pricing algorithms in the prohibited set. The aim is to test whether supra-competitive prices are less frequently set and competitive prices are not distorted. If so, then the prohibition of some pricing algorithms would make lower prices more frequent, with a corresponding increase in social welfare.

Harrington's three step research program can be applied both *ex-ante* as well as *ex-post*, i.e., as a prevention tool or a sanctioning tool. Nevertheless, its wording suggests an *ex-ante* approach. Clearly, this research program can be carried out for some subset of all possible inputs that can be fed into the different algorithms. But this subset can be smaller than necessary. This can be a serious limitation to the efficacy of such research program.

Other authors have similarly called for some sort of regulation, assessment or auditing of algorithms as a way to prevent or sanction 'algorithmic collusion'[38].

We can then pose the following questions: Do these approaches define the way forward? More generally, can competition and/or regulatory agencies rise to the challenge M. Gal encapsulates when she recommends that "*'smart coordination' by suppliers requires 'smart regulation'*"?[39] What type of regulation? And how smart can regulation be? The next two sections will explore this challenge, by appealing to recent literature on algorithms and the law and to the theory of computation.

IV. *Ex-ante* regulation, self-regulation and their limits

Given the multiplicity of algorithms firms can employ to implement pricing strategies, an *ex-ante* regulatory agency would be truly effective only if, when given any set/vector of algorithms, one or more per firm, it could ascertain whether they exhibited a "tendency to collude", and if yes then prohibit their use. Apart from the existence of limits to such ascertainment, as we will see below, the property "tendency to collude" must be properly defined. For Harrington (2018) it means choosing PA's that through their interaction, eventually result in supra-competitive prices. According to this author, the Finite Automata (FA's) implementing the 'tit-for-tat' strategy or the 'grim trigger' strategy are good PA candidates to fulfill such property[40]. The regulatory agency would need to define its own understanding in a clear and transparent way to minimize legal uncertainty and the probability of committing errors types I and II, through the collection of as much relevant information as possible.

---

[38] See Calvano et al. (2018a), pp. 14-16; Gal (May 2017) p. 6, through the creation of an "internal algorithmic police"; Mehra (2016), Part IV; Ezrachi & Stucke (2016), p. 21; Gal (2018), p. 39; T. Klein (2018), p. 14. See also Schwalbe (2018), pp. 21-23.

[39] See Gal (2018), p. 25.

[40] Note that in Harrington (2018), the prohibition is applied to PAs, not to learning algorithms themselves.

Pricing algorithms themselves can become quite complex. Consider the infinitely repeated prisoner's dilemma game (IRPD) as the paradigm for the strategic interaction between two potentially colluding firms in a market. The 'tit-for-tat' strategy can be implemented by a very simple FA with two states and a straightforward transition function. The 'grim trigger' strategy is also implementable by a FA with two states and an even simpler transition function. Both these strategies played by both players in the IRPD game can support (price) collusion as a subgame perfect equilibrium, provided certain conditions are satisfied. Similar strategies in an oligopoly game with more than two players can also support (price) collusion as a subgame perfect equilibrium, under certain conditions. But so can infinitely denumerable[41] many other vectors of more complex strategies, implementable by FA, with more states and more complex transition functions.

The regulatory agency could also attempt to identify families of 'learning algorithms' (typically, one learning algorithm per firm) such that when they play an infinitely repeated oligopoly game, they will eventually learn to collude. Each of these algorithms receives as input in each period publicly available relevant market data, such as prices charged by the different firms (through PAs) which, together with information on its own firm's costs, profit levels, and possibly other types of data, will choose a PA to be applied for a certain number of time periods. Whether each 'learning algorithm' will eventually learn to decode other 'learning algorithms' in the repeated game, even if only partially so, might also become possible.

An ever-increasing complexity of algorithms employed by firms would pose a serious challenge to the regulatory agency. Prohibited algorithms could be replaced by new ones that could either escape an ex-ante assessment altogether - depending on how easily any such prohibition could be circumvented - or, if not, burden yet again the regulatory agency, which would be pressed to decide in a timely fashion, lest it delay the use of efficiency enhancing algorithms.

It seems reasonable then to ask what kind of regulatory agency would have the information and knowledge required to ascertain the properties of ever evolving sets of algorithms submitted for its evaluation, while at the same time controlling for errors type I and II [42].

In face of such a demanding task, could this regulatory agency itself employ a "meta-algorithm" that could do the job, i.e., accept or reject any set of algorithms under submission, once the criteria for prohibition is clearly defined? Not only would such a "meta-algorithm", still being an algorithm, never commit type I and II errors - these could only come from ill-defined prohibition criteria -, but its computational capabilities could surpass any human regulator's computational capabilities, unless one is prepared to boldly assert otherwise, as it could question the validity of the '*Church-Turing Thesis*', discussed below.

If we accept the *Church-Turing Thesis*[43], which states that Turing Machines (TMs) are formal versions of algorithms, and that no computational procedure can be considered an algorithm

---

[41]   When applied to a set, the term "*infinitely denumerable*", which is equivalent to the term "*infinitely countable*", means that the cardinality of that set equals the cardinality of the set $\mathbb{N}$ of natural numbers. In contrast, the set $\mathbb{R}$ of all real numbers is uncountable.

[42]   Maybe competition agencies would consider useful to issue of guidelines on the use of algorithms by firms in the market, along the lines of what happens with the application of Article 101(3) of TFEU, where 'block exemptions' were created, or with the notion of 'hard core restrictions', such as RPM, as included in the 2010 EC guidelines on vertical restrictions (EC Guidelines on Vertical Restrictions, OJEU C130/01, 19.05.2010).

unless it can be presented as a TM, then, such "meta-algorithm" would be a TM, as are all the algorithms we have been talking about. And it would represent an upper bound to the computational capabilities of any human regulator, endowing the regulator with the means to perform its job. This job could be extended to allow the characterization of a particular algorithm, or set of algorithms, as "plus factor(s)" to 'an agreement' between firms employing such algorithms.

Since 'algorithmic collusion' is a possible outcome from the interaction of several algorithms, typically one per firm, that the regulator wants to prevent, the meta-algorithm would be simulating their interaction for any given data set that includes prices, costs and other information relevant for the pursuit of the firms' ultimate goals. In other words, this meta-algorithm is best interpreted as a Universal Turing Machine (UTM)[44].

Having reached this point we are confronted with the following:

Result 1: *There is no such meta-algorithm which, when presented with a set of algorithms, one per firm, together with any data set, suitably encoded, is able to decide whether they belong to the set of algorithms that should be prohibited*. [A line of proof for this result is given in annex 1]

In other words, the "*ex-ante* algorithmic assessment problem", as characterized above, is unsolvable[45]. I believe that this result sets a limit on how "smart" and "transparent" *ex-ante* regulation of PAs can be, and casts some doubt over the efficacy of such regulation.

In contrast with Harrington (2018), I am redefining the class of algorithms that may be prohibited *per se* as learning algorithms and not PAs. In Harrington's set up 'PAs' are chosen by 'learning algorithms' as the infinitely repeated oligopoly game under analysis unfolds. In any case, all these different types of algorithms can be interpreted as Turing Machines. Therefore, the above result still would apply.

Note that this result does not say that for any given algorithm or set of algorithms there does not exist a "meta-algorithm" that is able to decide whether they should be prohibited or not. What it says is that there is no 'meta-algorithm' that can decide whether or not any algorithm or set of algorithms should be prohibited[46]. Therefore, a certain meta-algorithm may do the job for a particular algorithm or set of algorithms but may be unable to do the job for another

---

[43] See Lewis & Papadimitriou (1981) for a presentation of this thesis or conjecture. See also "*Church's Thesis*" in Davis & Weyuker (1983). Sometimes, the '*Church-Turing Thesis'* is presented as follows: *The Universal Turing Machine can perform any calculation that any 'human computer' can carry out.* By 'human computer' one means a human being using his/her own mind and any other tools to perform any type of computation. Note that this '*Church-Turing Thesis'* is not presented as a theorem, as it is not a mathematical result. It may be disproved in the future. However, according to most scholars that is quite unlikely to happen.

[44] See annex for a definition of a UTM.

[45] Desai & Kroll (2018) refer in their paper that the 'Halting problem' – see annex for a description of this problem - which is a well-known unsolvable problem in the theory of computation, implies that several other interesting problems are also unsolvable. And that these inherent limits to solvability indicate that «*insofar as law and policy seeks a general transparency tool that analyzes all disclosed algorithms for compliance with desired norms, such a tool will not be possible*». See also Kroll et al. (2017). Note that Rice's Theorem (1953), which I use to prove the above result – see annex - generalizes the theorem that states the insolvability of the 'Halting problem' (due to Alan Turing, 1936-7) – see Rogers (1987), p. 34. I used Rice's Theorem in another context more than twenty years ago – see Gata (1995). Kroll et al. (2017) make a reference to this theorem on p. 652, relating it to the Halting problem.

[46] Using *First Order Predicate Calculus* (or *First Order Logic*), the result can be stated along the following lines: $\neg \exists TM : \forall (\dots)$. Which does not mean it is not true that: $\forall (\dots) \exists TM : (\dots)$.

particular algorithm or set of algorithms. This lack of robustness may question the viability of an *ex-ante* assessment exercise.

The above impossibility result places serious limits to an *ex-ante* assessment exercise as seemingly proposed by some authors. Some will say that this is the price to pay for eschewing an assessment program that is allowed to commit errors type I and II, and that there is no need to pay such a heavy price. Maybe Harrington's three step research program, together with a definition of liability that maximizes the Likelihood Ratio $LR(PPA)$ points us in the right direction.

As a realistic alternative to never committing errors type I and II, Harrington proposes the maximization of the likelihood ratio $LR(PPA) \stackrel{\text{def}}{=} (1 - Prob[error\ type\ II]) \div Prob[error\ type\ I]$, where $LR(PPA)$ measures the efficacy of a particular definition of liability. In order to calculate such ratio, one will need to define a probability measure on the sample space Ω of all pricing algorithms PA. If we do not restrict at the outset the set of allowed pricing algorithms to be FA (more precisely, Moore machines) with at most two-states, then the space Ω is infinite. For example, we can take the strategy "*tit-for-tat*" in the IRPD Game and build another (supergame) strategy, call it φ_n, where, in the first period of the game the player cooperates, and after a deviation by the other player, this player deviates as well for n consecutive periods, where $n \geq 2$. For each number n, there is a low enough time discount factor for both players such that, under perfect information (a sufficient but not a necessary condition), the pair of strategies ("*tit-for-tat*"; φ_n) constitutes a subgame perfect equilibrium supporting cooperation/collusion in every period of the IRPD game. Clearly, there are infinitely many (supergame) strategies φ_n, where $n \geq 2$. That the space Ω is infinite but denumerable results from the well-known fact that the set of all algorithms (or TMs), of which Ω is a (strict) subset, is an infinitely denumerable set.

Assume now we can partition the space Ω into two disjoint subsets PPA and $\Omega \setminus PPA$, where PPA denotes the set of all prohibited pricing algorithms. Hence, all pricing algorithms in set $\Omega \setminus PPA$ are not prohibited (by the regulatory agency). Assume as well we can partition the space Ω into the disjoint subsets $\{pa\ is\ collusive\}$ and $\{pa\ is\ competitive\}$, where 'competitive' means 'not collusive'. Then, one can (theoretically) define the conditional probabilities $Prob[pa \notin PPA | pa\ is\ collusive]$ and $Prob[pa \in PPA | pa\ is\ competitive]$. However, can we decide whether any pricing algorithm pa is in the set PPA? More precisely, is there an algorithm (TM) that decides membership in set PPA? The same query can be asked about the subset $\{pa\ is\ competitive\}$.

Result 2: *The answer to both these questions is negative* – see annex 1 for a line of proof. This means none of the conditional probabilities is computable.

Since $(1 - Prob[error\ type\ II]) = 1 - Prob[pa \notin PPA | pa\ is\ collusive]$, and $Prob[error\ type\ I] = Prob[pa \in PPA | pa\ is\ competitive]$, it follows that *likelihood ratio* given by $(1 - Prob[error\ type\ II]) \div Prob[error\ type\ I]$, is a non-computable function[47]. Hence, its

---

[47] Given a probability space $(\Omega, \mathcal{F}, P)$ and a sub-sigma-algebra $\mathcal{B}$ of $\mathcal{F}$, the conditional probability $P(A|\mathcal{B})$ of a measurable subset $A \in \mathcal{F}$ is defined as the conditional expectation $E(A|\mathcal{B})$ of indicator function i_A of subset A given $\mathcal{B}$. I.e., $P(A|\mathcal{B}) \stackrel{\text{def}}{=} E(A|\mathcal{B}), \forall A \in \mathcal{F}$. The conditional probability $P(\cdot|\mathcal{B})$ is a mapping $\Omega \times \mathcal{F} \to [0,1] \subset \mathbb{R}$. A conditional probability $P(\cdot|\mathcal{B})$ is called *regular* if $(\forall \omega \in \Omega): P(\cdot|\mathcal{B})(\omega)$ is also a probability measure. Assuming $Prob[error\ type\ I] \neq 0$, the above defined *likelihood ratio* is a function mapping $\Omega \times \mathcal{F}$ into $\to [0, +\infty)$, given

maximization is an unsolvable problem. Of course, we do not expect the different jurisdictions to actually go through this type of maximization problem when attempting to define liability. Most likely, how liability is defined can be better understood as the result of a learning process by the jurisdiction itself over several years of law enforcement. In any case, the above result raises doubts on how far we can go in controlling for the efficacy of any definition of liability in competition law.

On a self-regulatory option, notice that at any point in time each firm may only know the algorithms it employs to carry out its market strategies[48], but not necessarily the algorithms employed by other firms operating in the market. Which means that each firm will not be able to simulate with a high enough degree of accuracy how its own algorithms will behave period after period as they interact with other algorithms. Unless each firm engages in repeated simulation exercises by assuming different sets/vectors of algorithms that may be used by other firms. Which could be quite costly and not very informative. Under such limited information scenario, self-regulation may not be an effective enough option to meet the standards imposed by a competition or regulatory agency.

V. *Ex-post* auditing and sanctioning and its limits

In an *ex-post* situation, i.e., when a specific *ex-post* investigation is opened[49], we assume a given set of algorithms employed by the different firms will be analyzed by the regulatory/competition agency. It is likely this investigation will involve simulating the behavior of those algorithms as they are given as input data on relevant variables such as prices. Such data, at least in part, are produced period after period by the algorithms themselves, as their output while they run in an interactive way. The aim will be to compare the output from such simulations to the output observed in the market and decide whether it can be established with a high enough degree of certainty there was algorithmic price collusion[50].

---

the probability space $(\Omega, \mathcal{F}, P)$, where Ω is the set (space) of all pricing algorithms, $\mathcal{F}$ is a σ-algebra in Ω, and P is a probability measure.

[48] Assume that to know an algorithm might mean to know some, but not necessarily all, of its relevant characteristics.

[49] In the EU, Article 101 (TFEU) cases can originate in: (1) a complaint, (2) opening of an own–initiative investigation, (3) information reported by individuals via the "whistleblower" tool, or (4) a leniency application from one of the participants to a cartel. – see http://ec.europa.eu/competition/antitrust/procedures_101_en.html. Recall that Article 101 (TFEU) prohibits agreements between undertakings, decisions by associations of undertakings and concerted practices which may affect trade between Member States and which have as their object or effect the prevention, restriction or distortion of competition within the internal market. Regulation 1/2003 introduced a system of decentralized *ex post* enforcement, in which the European Commission and the national competition authorities of the EU Member States forming together the European Competition Network (ECN), pursue infringements of Articles 101 and 102 TFEU. Article 102 prohibits the abuse of a dominant position within the internal market in so far as it may affect trade between Member States.

[50] In a more recent paper, Calvano et al. (2018b) create an environment to analyze the interaction among a number of 'Q-learning algorithms' in the context of an oligopoly model of price competition with Logit demand and constant marginal costs. They show that «*algorithms consistently learn to charge supra-competitive prices, without communicating with each other. The high prices are sustained by classical collusive strategies with a finite punishment phase followed by a gradual return to cooperation. This finding is robust to asymmetries in cost or demand and to changes in the number of players*».

Desai & Kroll (2018), Kroll et al. (2017), and other authors, discuss different approaches for testing and evaluating algorithms[51]. For example, '*White-box testing*' is a method for analyzing software that tests internal structures or workings of an application. It is a method to test an application at the level of the 'source code'[52]. '*Black-box testing*' is a method for software testing that examines the functionality of an application without peering into its internal structures or workings. Both types of testing can follow methods: (1) static methods, which look at the code without running the program; and (2) dynamic methods, which run the program and assess the outputs for particular inputs or the state of the program as it is running[53]. Nevertheless, and as pointed out by Desai & Kroll, these testing methods cannot escape the unsolvability of the 'Halting Problem'.

Apart from '*White-box testing*' and '*Black-box testing*', Desai & Kroll (2018) analyze a third way to test software and algorithms, namely "*Ex-post Analysis and Oversight*". This approach may have several appealing features but as recognized by the authors, «*software that uses certain types of machine learning or is modified frequently by or is modified frequently by its operators to respond and adapt to dynamic inputs and user behavior, are not addressed well by the solutions presented* ['white-box testing', 'black-box testing', "ex-post analysis and oversight]. *Many systems change often, either because of regular changes by designers or because they use automated processes such as online machine learning models which "can update their predictions after each decision, incorporating each new observation as part of their training data." The approach of creating an audit log showing that everyone is subject to the same decision policy is less useful when systems are dynamic and change over time because the system may (desirably) change between decisions*». The type of algorithms that are raising concerns in the enforcement of antitrust law are exactly learning algorithms that can update their predictions after each decision, incorporating each new observation as part of their training data.

To competition law and economics scholars and practitioners, it still seems unclear how efficient can *ex-post* auditing and sanctioning become when dealing with algorithms as facilitators of collusive behavior in repeated games, and with increasingly sophisticated algorithms that can interact as autonomous implementers of pricing strategies, learning to collude without any explicit instructions provided by human agents.

VI. Conclusions

To what extent can competition policy keep a distinction between tacit and explicit collusion? To the difficulties pointed out by L. Kaplow (2011), namely on a consensual and precise

---

[51] Kroll and Desai & Kroll do mention computational unsolvability in the legal context of algorithmic transparency. My goal is to explore the limits that are imposed by computational unsolvability on *ex-ante* and *ex-post* antitrust policy in the specific context of algorithmic collusion.

[52] According to the *Linux Information Project*[52], 'source code' is defined as the version of software as it is originally written (i.e., typed into a computer) by a human in plain text (i.e., human readable alphanumeric characters) - see http://www.linfo.org/

[53] As referred by Kroll et al. (2017, p. 646/7), «*Computer scientists evaluate* [computer] *programs using two testing methodologies: (1) static methods, which look at the code without running the program; and (2) dynamic methods, which run the program and assess the outputs for particular inputs or the state of the program as it is running. Dynamic methods can be divided into (a) observational methods in which an analyst can see how the program runs in the field with its natural inputs; and (b) testing methods, which are more powerful, where an analyst chooses inputs and submits them to the program*».

definition of the concept "agreement", the increasing use of "smart algorithms" may introduce additional challenges as these type of algorithms can facilitate tacit collusion and lead to an increased blurring of borders between tacit and explicit collusion – see also Harrington (2018).

As shown by some authors in very recent work – see Calvano et al. (2018a,b) - the interaction among Q-learning algorithms in the context of an oligopoly model of price competition, these algorithms consistently learn to charge supra-competitive prices, without communicating with each other. The high prices are sustained by collusive strategies with a finite punishment phase followed by a gradual return to cooperation. Hence, we might not be any more in the realm of science fiction.

How reliable and effective can be an *ex-a*nte supervision and control exercised over algorithms? How reliable and effective cab be an *ex-post* auditing and sanctioning of collusive algorithms? I show that computational unsolvability casts some doubts over how efficient both these approaches can become[54].

Nevertheless, software testing is carried out routinely – as shown by Desai & Kroll (2018) and by Kroll et al. (2017) -, and we may simply acknowledge that the way liability is defined and the way ensuing regulation and sanctioning are exercised rely on a "learning-by-doing" approach and accept that errors type I and II will be committed, as a price to pay for getting away from non-decidability or computational unsolvability. However, it seems we have little clue about the magnitude of such errors. Which in turn may affect how productive the "learning-by-doing" approach may be. Moreover, the ongoing research on various challenges current software systems pose will likely have an impact on how to deal with the legal challenges referred above.

Competition law enforcement and policy have much to gain from an interdisciplinary collaboration with computer science and mathematics. Some familiarity with computability, computational complexity and, I would venture to say, with the theory of languages and grammars, may help scholars working in competition law and economics to better face the legal challenges posed by artificial intelligence. A refinement of some legal concepts could very well be a positive externality from such collaboration.

---

[54] Notice that the use of "quantum TM" does not make solvable a problem that is unsolvable using a non-quantum TM – see D. Deutsch (1985).

Annex 1:

Line of Proof (for Result 1): If we accept the *Church-Turing Thesis* (which states that Turing Machines (TMs) are formal versions of algorithms, and that no computational procedure can be considered an algorithm unless it can be presented as a TM – see Lewis & Papadimitriou, ch.5), a 'learning algorithm' (LA) can be regarded as a TM. In each period a LA will receive as input some data on prices, costs, and other relevant parameters. Given this input (notice that these data can be coded, by use of *Gödel numbers*, into a single integer), the LA outputs a 'pricing algorithm' (that can be played by a FA or a TM). This pricing algorithm can also be coded into a single integer. Hence, the LA computes a function mapping integer numbers into integer numbers. The domain and codomain of such mapping are subsets of the set $\mathbb{N}$ of natural numbers. This mapping is a partial recursive 1-ary function. Meaning that when an input is undefined (e.g., there are natural numbers which do not code any meaningful data) so is its output; and that this mapping is computable, i.e., recursive (which is by assumption). If we want some LA to be prohibited *per se*, it means that for certain inputs, they will compute, or output, unacceptable 'pricing algorithms'. We can then say that each of these LA's computes a partial recursive 1-ary function belonging to a certain "forbidden" or "prohibited" set $\mathbb{C}$. Let $g_{\mathbb{C}}$ denote the set of all codes where each code uniquely identifies a partial recursive 1-ary function in set $\mathbb{C}$. Set $g_{\mathbb{C}}$ is clearly non-empty (as there are acceptable pricing algorithms). And set $g_{\mathbb{C}}$ is also different from the set $\mathbb{N}$ of natural numbers (as there are unacceptable pricing algorithms, such as the ones implementing a grim-trigger strategy). I.e., $\emptyset \neq g_{\mathbb{C}} \neq \mathbb{N}$. Then, by H. G. Rice's Theorem (1953) – see below -, set $g_{\mathbb{C}}$ is not recursive, i.e., there is no 'algorithm' that decides membership in this set. Therefore, there is no 'algorithmic judge' that, when presented with a LA, will be always able to decide whether this LA should be prohibited or not.

It is easy to extend the previous result and proof to the case when a vector of TM's are to be subject to analysis and judgement by an "algorithmic regulator/supervisor". A vector of n TM's can given as an input to n Universal Turing Machine (UTM) through their *Gödel numbers*. Or just encode each of the n TM's into a *Gödel number* g_i and then encode the resulting vector of n *Gödel numbers* as follows: $[g_1, \ g_2, \ \dots, g_n] = \prod_{i=1}^{n} p_i^{g_i}$, where p_i are prime numbers. Give also as input to the UTM the data that is to be given to the n TM's. And let the UTM simulate the calculations carried out by the n TM's. ∎

Line of Proof (for Result 2): It follows from Result 1, by use of Rice's Theorem**.** As there is no algorithm that decides membership in set PPA, there is no algorithm that decides membership in its complement, i.e., in the set of all PA's such that $\{PA$ is competitive$\}$.

Theorem by H. G. Rice (1953): *Let $\mathcal{D}$ be a set of partial recursive one-ary functions. And let $\mathcal{G}_{\mathcal{D}}$ be denote the set $\{n \in \mathbb{N}: \varphi_n^1 \in \mathcal{D}\}$. If $\emptyset \neq \mathcal{G}_{\mathcal{D}} \neq \mathbb{N}$ then the set $\mathcal{G}_{\mathcal{D}}$ is not recursive* - see H. Rogers (1987), or M. Davis & E. Weyuker (1983) for a statement and proof of Rice's Theorem.

Rice's Theorem basically states that in computational theory, all non-trivial (i.e., neither true nor false for every computable function) semantic (i.e., behavior) properties of algorithms/computer programs are undecidable. For example, whether a computer program will eventually halt on any input string is a semantic property.

Annex 2:

A **Finite Automaton** (FA) is a simple computational model with a fixed memory. It can be defined as a 5-tuple $(S, \Sigma, \delta, s_0, F)$, where S is the set of states of the machine, Σ is the input alphabet, $\delta: S \times \Sigma \to S$ is the transition function, s_0 is the initial state, and F is the set of final (or accepting) states, where $F \subseteq S$.

A useful way of looking at a FA is to regard it as a simple 'language recognition device'. When we feed any string x of symbols (over an alphabet) to an FA, the device (FA) can either end up in an 'accepting state' or not. If it does, we say this string has been accepted, or recognized, by the device (FA). The set of all strings accepted by an FA represents the 'language' accepted by this FA. Another way to look at an FA is to regard it as a simple computing device with a fixed (finite) capacity central processing unit (CPU), with no auxiliary memory, and where its output is binary (0 or 1, Yes or No).

There are several variations and extensions of the notion of FA, such as an FA *with output*. For example, a **Moore machine** is defined as a 6-tuple $(S, \Sigma, \Delta, \delta, \lambda, s_0)$, where S, Σ, δ, and s_0 are defined as before, and where Δ is the output alphabet and $\lambda: S \to \Delta$ is the output function. An FA can then be seen as a Moore machine where the output alphabet $\Delta = \{0,1\}$, and where state s is 'accepting' if and only if $\lambda(s) = 1$.

A **Turing Machine** (TM) is a very general model of a computer (a CPU, plus an auxiliary memory and an input/output device). It is a computing device with an unbounded and unrestricted access memory. This feature sets the computational capability of a TM quite above the computational capability of a FA. Informally, a TM consists of:
(1) A collection of distinct symbols called an alphabet A, and which includes a symbol called the 'blank symbol';
(2) A tape, i.e., a "roll of paper" on which calculations are performed, which is divided into cells (or squares) and is infinite in both directions. At any given time during a "computation", all but a finite number of cells are "blank", i.e., contain the blank symbol;
(3) A finite set of states. A state can be thought of as describing the internal configuration of the TM at any particular instant;
(4) A read-write head which at any given time is scanning (i.e., reading) the contents of one cell of the tape, and is capable of replacing the symbol scanned by another symbol;
(5) A finite ordered list of instructions. Each instruction either:
    (5.1) tells the computer to halt (i.e., stop) or,
    (5.2) tells the computer what next state to enter and tells the read-write head to do one of the following:
        (5.2.1) move one cell to the left;
        (5.2.2) move one cell to the right;
        (5.2.3) replace the symbol scanned by another symbol.

Each TM can be thought of as a finite list of instructions. Each instruction describes the present status of the machine (i.e., the present state and the symbol being read by the tape head) and what the next step in the computation will be. To each TM we can assign a number (a positive integer) which uniquely identifies it. That is, to each TM we can associate a "code" number. But not every positive integer is the code for some TM.

Formally, a **Turing Machine** is defined as a quadruple (K, Σ, s_0, δ), where K is a finite set of states (of the machine) not containing the 'halt state' h; Σ is an alphabet, containing the blank symbol, but not containing the symbols L and R (for 'Left' and 'Right'); s_0 is the initial state; and $\delta: K \times \Sigma \to (K \cup \{h\}) \times (\Sigma \cup \{L, R\})$ is the transition function.

A **Universal Turing Machine** (*UTM*) is a TM that takes as arguments both the encoding $\rho(\mathcal{M})$ of any Turing Machine $\mathcal{M}$, and any input string w, and performs whatever operations on w would have been performed by $\mathcal{M}$ – see e.g., Lewis & Papadimitriou (1981), section 5.7. The encoding of any TM can follow the Gödel numbering system, itself based on the Fundamental Theorem of Arithmetic, also called

the Unique Factorization Theorem (due to C. F. Gauss, 1801), which states that: *Any positive integer can be uniquely factored into a product of powers of prime numbers*.

A *UTM* can also take as arguments the encodings of n Turing Machines, plus any input strings, one per TM, and simulate the interaction of these n TMs when initiated with these input strings.

The Halting Problem (when applied to TMs): Given an arbitrary Turing Machine $\mathcal{M}$ and an arbitrary input w, can it be determined whether this TM will eventually halt on input w? It can be shown that: *Given an arbitrary Turing Machine $\mathcal{M}$ and an arbitrary input w, there is no algorithm for determining whether $\mathcal{M}$ will eventually halt on input w*. Hence the **Unsolvability of the Halting Problem**.

Bibliography:

- Anitha, P., G. Krithka and M. D. Choudhry (2014): "*Machine Learning Techniques for learning features of any kind of data: A Case Study*", International Journal of Advanced Research in Computer Engineering & Technology, Vol. 3(12), pp. 4324-4331.
- Aumann, Robert (1976): "*Agreeing to Disagree*", The Annals of Statistics, Vol. 4(6), pp. 1236-1239.
- Ben-Porath, Elchanan (1990): "*The Complexity of Computing a Best Response Automaton in Repeated Games with Mixed Strategies*," Games and Economic Behavior, Vol. 2(1), pp. 1-12.
- Borenstein, Severin (2004): "*Rapid Price Communication and Coordination: The Airline Tariff Publishing Case (1994)*", in John E. Kwoka Jr. & Lawrence J. White (Eds.), The Antitrust Revolution: Economics, Competition, and Policy, Oxford University Press.
- Calvano, Emilio, et al. (2018a): "*Algorithmic Pricing: What Implications for Competition Policy*?" WP, University of Bologna, CEPR & Toulouse School of Economics, June.
- Calvano, Emilio, et al. (2018b): "*Artificial Intelligence, Algorithmic Pricing and Collusion*" WP, University of Bologna, Toulouse School of Economics, EUI & CEPR, December.
- Chopra, Samir & White, Laurence F. (2011): A Legal Theory for Autonomous Artificial Agents. University of Michigan Press.
- Competition & Markets Authority (2018): "*Pricing algorithms: Economic working paper on the use of algorithms to facilitate collusion and personalized pricing*", CMA94, 8th October.
- Conlisk, John (1996): "*Why Bounded Rationality*?" Journal of Economic Literature, Vol. 34, pp. 669-700.
- Davis, Martin & Weyuker, Elaine (1983): Computability, Complexity, and Languages: Fundamentals of Theoretical Computer Science. Academic Press, Inc.
- Desai, Deven & Kroll, Joshua (2018): "*Trust but Verify: A Guide to Algorithms and the Law*", Harvard Journal of Law and Technology, Vol. 31(1), pp. 1-64.
- Deutsch, David (1985): "*Quantum theory, the Church-Turing principle and the universal quantum computer*", Proceedings of the Royal Society of London, A 400, pp. 97-117.
- Fudenberg, Drew & Levine, David K. (1998): The Theory of Learning in Games. MIT Press.
- Fudenberg, Drew & Tirole, Jean (1991): Game Theory. MIT Press.
- Gal, Michal S. (2017): "*Algorithmic-Facilitated Coordination: Market and Legal Solutions*", Competition Policy International (CPI), May.
- Gal, Michal S. & Elkin-Koren, Niva (2017): "*Algorithmic Consumers*," Harvard Journal of Law and Technology, Vol. 30(2), pp. 309-353.
- Gal, Michal S. (2018): "*Algorithms as Illegal Agreements*", Berkeley Technology Law Journal, (*forthcoming*).
- Gata, Joao E. (1995): "*Infinite Regression in Strategic Decision Making: An Application of Rice's Theorem*", Discussion Paper No. 95/38, DERS/University of York, UK.
- Gilboa, Itzhak (1988): "*The Complexity of Computing Best-Response Automata in Repeated Games*," Journal of Economic Theory, Vol. 45, pp. 342-352.
- Harrington, J. E. (2018): "*Developing Competition Law for Collusion by Autonomous Artificial Agents*", WP, Wharton School/University of Pennsylvania, April.
- Ivaldi, Mark et al. (2003): "*The Economics of Tacit Collusion*", Final Report for the DG COMP/EC, March.
- Kalai, Ehud (1990): "*Bounded Rationality and Strategic Complexity in Repeated Games*," in Tatsuro Ichiishi, Abraham Neyman & Yari Tauman (Eds.), Game Theory and Applications, pp. 131-157, Academic Press.
- Kaplow, Louis (2011): "*On the Meaning of Horizontal Agreements in Competition Law*", California Law Review, Vol. 99, pp. 683-808.
- Klein, Joel I. (1999): "*Competition in the Airline Industry*", Testimony as Assistant Attorney General, Antitrust Division, US Department of Justice, before the Committee on Commerce, Science and Transportation, US Senate, March 12.
- Klein, Timo (2018): "*Assessing Autonomous Algorithmic Collusion: Q-Learning under Sequential Pricing*", Amsterdam Law School Legal Studies Research Paper No. 2018-15, June.
- Kovacic, W. E., Marshall, R. C., Marx, L. M. & White, H. L. (2011): "*Plus Factors and Agreement in Antitrust Law*", Michigan Law Review, Vol. 110, pp. 393-436.
- Kroll, Joshua et al. (2017): "*Accountable Algorithms*", University of Pennsylvania Law Review, Vol. 165, pp. 633-705.
- Lewis, David (1969): Convention: a philosophical study. Reprint 2002. Blackwell Publishers.

- Lewis, Harry & Papadimitriou, Christos (1981): Elements of the Theory of Computation. Prentice-Hall.
- McSweeny, Terrell (2017): "*Algorithms and Coordinated Effects*". Presentation given at University of Oxford Center for Competition Law and Policy, Oxford, UK, May 22nd.
- Mehra, Salil K. (2016): "*Antitrust and the Robo-Seller Competition in the Time of Algorithms*", Minnesota Law Review, Vol. 100, pp. 1323-1375.
- OECD (2017): "*Algorithms and Collusion: Competition Policy in the Digital Age*". Paris.
- Ohlhausen, Maureen (2017): "*Should We Fear the Things That Go Beep In the Night? Some Initial Thoughts on the Intersection of Antitrust Law and Algorithmic Pricing*", Remarks from the Concurrences Antitrust in the Financial Sector Conference, New York, May 23rd.
- Papadimitriou, Christos (1992): "*On Players with a Bounded Number of States*," Games and Economic Behavior, Vol. 4(1), pp. 122-131.
- Petit, Nicolas (2017): "*Antitrust and Artificial Intelligence: A Research Agenda*," Journal of European Competition Law and Practice, Vol. 8(6), pp. 361-362.
- Rogers, Hartley (1987): Theory of Recursive Functions and Effective Computability. MIT Press.
- Rubinstein, Ariel (1986): "*Finite Automata Play the Repeated Prisoner's Dilemma*", Journal of Economic Theory, Vol. 39, pp. 83-96.
- Rubinstein, Ariel (1998): Modeling Bounded Rationality. MIT Press.
- Russell, Stuart & Norvig, Peter (2016): Artificial Intelligence: a Modern Approach (3rd edition). Pearson Education Ltd.
- Salcedo, Bruno (2015): "*Pricing Algorithms and Tacit Collusion*", WP, Pennsylvania State University.
- Schwalbe, Ulrich (2018): "*Algorithms, Machine Learning, and Collusion*", WP Institute of Economics, University of Hohenheim, Germany.
- Vestager, Margrethe (2017): "*Algorithms and Competition*". Presentation at the Bundeskartellamt 18th Conference on Competition, March 16th.