PART I

Introduction and Setting the Stage for a Law of Algorithms

1

An Introduction to Law and Algorithms

Woodrow Barfield and Jessica Barfield

INTRODUCTION

A body of law is currently being developed in response to algorithms which are designed to control increasingly smart machines, to replace humans in the decision-making loops of complex systems, or to account for the actions of algorithms that make decisions which affect the legal rights of people. Such algorithms are ubiquitous; they are being used to guide commercial transactions, evaluate credit and housing applications, by courts in the criminal justice system, and to control self-driving cars and robotic surgeons. However, while the automation of decisions typically made by humans has resulted in numerous benefits to society, the use of algorithms has also resulted in challenges to established areas of law. For example, algorithms may exhibit the same human biases in decision-making that have led to violations of people's constitutional rights, and algorithms may collectively collude to price fix, thus violating antitrust law.

When algorithmic-driven systems are used in varying ways, algorithms become not just a method of filtering data, but a way of outsourcing decision-making from human to machine or to a software bot. And crucially, despite claims from some that the digital, computerized nature of algorithms means they are free of bias, in fact, we now know the opposite may be true. In situations where humans code algorithms, consciously or not, they may seed them with their own flawed perspectives. And in the case where algorithms learn, they do so by drawing on existing information to make decisions, as a result, if the training data is flawed in some way, algorithms have the potential to exacerbate or replicate human bias.

In this chapter, we review some of the fundamental characteristics of algorithms which we believe are essential to establishing a law of algorithms and we discuss different areas of law that are being challenged when algorithms are used in contexts which traditionally have required human judgment. Legislators in the United States, European Union, and Asia are beginning to discuss how to regulate increasingly smart systems controlled by algorithms,[1] and to some extent cases have been litigated which involve the use of algorithms;[2] however, we are still in the early stages of developing a law of algorithms and determining how to regulate their use. In our view, a "law of algorithms" is necessary because they are a fundamental building block of biological and non-biological systems, and in one form or another, algorithms have been the subject of litigation in patent, First Amendment, civil rights, employment, and criminal law.

[1] In 2007, the South Korean government proposed a Robot Ethics Charter; in 2011, the UK Research Council EPSRC published five ethical principles for industry; and in 2017, the Association for Computing Machinery published seven principles for algorithmic transparency and accountability.

[2] *In re. Facebook Biometric Info. Privacy Litig.*, 185 F. Supp. 3d 1155 (ND Cal. 2016).

So, what are algorithms? Algorithms are a set of rules or instructions that are followed when performing calculations, or more generally, a set of problem-solving procedures which when followed produce a certain output.[3] In *Gottschalk* v. *Benson*, the Court used a narrow definition of an algorithm and defined an algorithm as a procedure for solving a given type of *mathematical problem*.[4] They are now most familiar as instructions embodied within computer programs, such as those which make artificial intelligence (AI) possible. But before algorithms were driving advances in AI and resulting in challenges to established areas of law, for hundreds of millions of years algorithms have silently been playing a key role in the process of evolution leading to complex forms of life.[5] In fact, we humans can be described as an organism consisting of trillions of cells each of which computes at the molecular level using algorithms.[6] But, of course, there are limits to legal rights associated with algorithms, for example, in *Gottschalk*, the Court discussed the long-established principle that "[p]henomena of nature, though just discovered . . . are not patentable . . ."[7] However, as Andrew Chin discusses in his chapter in this volume, not all algorithms are a product of nature and thus, if the requirements under the patent law are met, algorithms may receive patent protection. At this point, the reader should be at ease as the authors' digression to biology is temporary, and will not be the focus of this chapter, but done sparingly to point out that algorithms are not a recent phenomenon, but a feature of evolutionary processes which have occurred over hundreds of millions of years. Of course, from the perspective of law, there are essential differences between the algorithms which are a process of nature, the algorithms written by programmers, and the algorithms derived from machine-learning techniques. Among others, the differences involve the purposes for which the algorithms are designed and the legal consequences of using algorithms that are independent from human input and control.

As algorithms proliferate into society, legal scholars, courts, and legislators are beginning to ask timely questions about their use, such as the role that algorithms play in expressing bias toward a constitutionally protected group, or the role that an algorithm has in contributing to an accident involving an autonomous car[8] or robotic surgeon,[9] or for purposes of copyright or patent law, the role of an algorithm in writing a story, composing music, or serving as an inventor. And from a constitutional law perspective, as algorithms contribute to the speech output of different forms of technology, a question being discussed is whether algorithms should be considered a form of speech and therefore receive protection in the United States under the First Amendment.[10] We argue that for each of these questions, it will be important for courts to look carefully at the specific algorithm(s) in question in order to determine which aspect of the algorithm (or algorithms) is creative, culpable, "speaks," or, for purposes of assigning liability, is independent from human supervisory control. As we begin to provide answers to these questions, and as courts litigate disputes involving algorithms, we take the

3 R. Sedgewick and K. Wayne, *Algorithms*, 4th edn. (Addison-Wesley, 2011).
4 *Gottschalk* v. *Benson*, 409 US 63, 93 S.Ct. 253, 34 L.Ed.2d 273, 175 USPQ 673 (1972).
5 W.-K. Sung, *Algorithms in Bioinformatics: A Practical Introduction*, 1st edn. (Chapman & Hall/CRC Mathematical and Computational Biology, 2009).
6 C. Calude and G. Paun, *Computing with Cells and Atoms: An Introduction to Quantum, DNA and Membrane Computing* (CRC Press, 2000).
7 *Gottschalk*, above note 4.
8 S. Beiker and R. Calo, Legal Aspects of Autonomous Driving (2012) 52 *Santa Clara Law Rev.* 1145, https://ssrn.com/abstract=2767899.
9 *Taylor* v. *Intuitive Surgical, Inc.*, 355 P.3d 309 (2015).
10 T. Wu, Machine Speech (2012–13) 161 *Univ. Pa. Law Rev.* 1495, https://scholarship.law.upenn.edu/penn_law_review/vol161/iss6/2.

view that we are simultaneously developing a law of algorithms and creating legal precedent for future cases involving algorithms.

As an exercise in legal analysis, consider an algorithmic-controlled system which results in damages, in this example, we can ask: Which aspects of the algorithm has importance for the law? Let me note here that there are many different types of algorithms, and some may be more likely to result in outcomes that may lead to legal disputes and thus court actions than others. For this reason, we argue that an examination of the specific features of an algorithm is important to those who wish to develop a law of algorithms and to courts resolving disputes. With these thoughts in mind, consider the following algorithm which we provide here for discussion purposes (but without reference to a specific legal dispute), as it has general value for the points we wish to make:

```
def find_max (L):b
if lenL) == 1:
return L[0]
v1 = L[0]
v2 = find_max(L[1:])
if v1 > v2:
return v1
else:
return v2
```

Like the process of statutory construction, in which courts are called upon to interpret the language of statutes and legislation, so too can an analysis of the lines of code comprising an algorithm be used to determine the language and purpose of the algorithm. In fact, disputes involving patents typically turn on claim construction, in which the claims are listed step by step, thus representing an algorithm (specific examples are presented below). In the above algorithm, the set of instructions are unambiguous, so it would be relatively easy for a court, for example, through expert testimony, to determine the purpose of the algorithm. For those whose programming experience is rusty, the algorithm solves the following problem: given a list of positive numbers, it returns the largest number on the list. To accomplish this, the algorithm has input which consists of a list L of positive numbers. This list must contain at least one number and the algorithm has the following output: a number, which will be the largest number of the list. Of course, examining how an algorithm works for purposes of litigation assumes that an algorithm can be admissible as evidence in a court proceeding – a topic discussed by Andrea Roth in Chapter 19 in this volume. Knowing the purpose of an algorithm is essential for determining the extent to which an algorithm contributes to a violation of the law. However, as algorithms become mathematically complex, it will become difficult for litigators to explain their reasoning to a judge or jury; this illustrates a growing problem in resolving disputes which involve systems whose performance is guided by algorithms.

Another issue involving algorithms that will surely confront courts is that systems will be controlled by not one, but many, algorithms; thus, it may be the collective performance of algorithms that is of interest in a court proceeding. This, of course, adds to the complexity of resolving disputes which involve algorithms. Additionally, if a programmer who wrote an algorithm can be identified, examining the features of an algorithm could produce important evidence in determining whether the programmer intended for a certain outcome to occur.

In contrast, if the algorithm(s) was derived from techniques such as deep learning, in which no human was involved in writing the algorithm(s) or determining the systems output would it still be relevant for a court to examine the lines of code to determine the "intent" of the non-human entity which created the algorithm(s)? As some have argued, the answer is only to the extent that the "algorithmic-based system" has been granted legal person status, because without such status afforded to an algorithmic entity, there is a gray area which currently exists in the law in which there may be no legal person to hold liable when damages have occurred.[11]

Returning to the above algorithm, note that each step of the algorithm is easily translated into a programming language. A programming language as opposed to machine language (or object code) provides a judge and jury with a readable transcription of the actions of an algorithm that may have led to an outcome contrary to the law. Further, the above algorithm has defined inputs and outputs, which surely would be of interest to a trier of fact determining the extent to which an algorithm led to damages. We should note here that just as with human behavior which may result from the collective action of many thousands (or millions) of neurons, as stated above, so too does "algorithmic performance" for a given system rely not just on one but quite often the combined output of many algorithms;[12] this, of course, complicates matters for a court determining the extent to which algorithms contribute to liability and that are tasked with explaining the role of algorithms in a dispute (this is referred to among legal scholars as the lack of transparency problem).

Referring back to the previous algorithm, it is guaranteed to terminate (thus is not open loop) if L is of length 1. If L has more than one element, find_max() is called with a list that is one element shorter and the result is then used in a computation. In addition, the nested call to find_max() always terminates, each time find_max() is called with a list that is shorter by one element. Eventually, the list will be of length 1 and the nested calls will end. The fact that the algorithm ends based on the instructions provided by its code provides a definite and unambiguous action for courts to consider – this is essential in litigating disputes. So, just based on thinking about the above simple algorithm with only nine lines of code, we can surmise that there are several aspects of an algorithm that could be of interest to a judge or jury considering the actions of algorithms, or to legislators tasked with regulating algorithmic-based systems. To add another layer of complexity to our thinking about algorithms and law, we argue that the aspects of an algorithm that will be of interest to the law will be dependent on the type of legal dispute involved. For example, if a certain intent were required for a violation of a criminal law statute, then any algorithm written by a defendant controlling the actions of an artificially intelligent entity that would "mathematically model" the necessary intent would be important evidence to produce in a criminal proceeding.

Clearly, the result (or output) of an algorithm if used for decision-making purposes could implicate the law if the decision resulted in behavior contrary to established legal principles. For example, in criminal law, if the use of algorithms for sentencing purposes resulted in bias toward members of a constitutionally protected group, this would violate the Fourteenth Amendment to the US Constitution. In our view, this raises the question: Where in the lines of code is the bias manifested? Among major categories of algorithms, there are those that operate by techniques such as brute force, divide and conquer, decrease and conquer, dynamic programming, transform and conquer, or backtracking. For the example just

[11] C. E. A. Karnow, *Future Codes: Essays in Advanced Computer Technology and the Law* (Intellectual Property Series, Computing Library), 1st edn. (Artech House, 1997).
[12] C. C. Aggaswal, *Neural Networks and Deep Learning: A Textbook*, 1st edn. (Springer, 2018).

mentioned, which algorithmic technique (if any) is more likely to lead to biased intent toward a member of a constitutionally protected group?

But examining the code will be more involved than just noting the high-level algorithmic technique used in a program. For example, if property is damaged by the end effectors of a robot that delivered excessive force, then the algorithms modeling force and the algorithms controlling the robot's visual system and behavior would be contributors to the excessive force used. However, this particular code, with matrix calculations, coordinate transformations, and control theory models is mathematically complex and thus difficult for a judge or member of a jury not trained in engineering or computer science to comprehend. But communicating this information to a judge or jury will be essential to resolving a dispute. Perhaps in response to the complexity of algorithms, a court which focuses on litigation involving algorithms will need to be developed. This is not a novel idea as in the United States there is a Tax Court, a Court of Federal Claims, a Foreign Intelligence Surveillance Court, and a Bankruptcy Court, and formerly there was a Board of Patent Appeals and Interferences, to name just a few courts with specialized subject matter. Just as Ryan Calo has advocated for a federal agency to regulate robotics,[13] the challenge to our legal system in resolving disputes involving systems controlled by algorithms, especially for increasingly autonomous entities, seems to call for a specialized court with expertise in litigating disputes which involve algorithms (and other AI techniques).

Considering another area where algorithms may challenge established law, under intellectual property law the question of whether algorithms are simply a process of nature and thus not patentable subject matter is another timely debate receiving attention in different jurisdictions. And for copyright law, the question of whether the musical compositions, paintings, and other works of authorship generated by algorithms constitute copyrightable subject matter has given rise to another important debate occurring within the legal community. But, in each of these situations, we can ask: Which aspect of the algorithm itself is to be considered a process of nature or independently creating a work of authorship? Further, from a constitutional law perspective, if algorithms are embedded within systems which communicate using natural language, under the US Constitution is such algorithmic speech protected under the First Amendment?[14] And, if so, which lines of code contribute to speech and thus should receive protection, or is an analysis at the level of an algorithm too detailed, and thus the Court should focus only on the output or spoken speech of the algorithmic entity? And more fundamentally, is an algorithm itself a form of expression and thus speech? (See Chapters 25, 27, and 28 in this volume.) And for commercial law, it is important to ask whether an algorithm is considered a product or a service. This is an important distinction because a major factor in determining liability for software (and algorithms are often embedded within software) is the classification of the software. In the United States, under the Uniform Commercial Code (UCC), software can be classified as a product or a service.[15]

[13] R. Calo, The Case for a Federal Robotics Commission, Brookings Institution Center for Technology Innovation (September 2014), https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2529151.

[14] T. M. Massaro and H. Norton, Siri-ously? Free Speech Rights and Artificial Intelligence (2016) 110 *Nw. Univ. Law Rev.* 1169.

[15] For a discussion of the concept of software as a product or service within the UCC, see R. Raysman and P. Brown, Applicability of the UCC to Software Transactions; Technology Today, *NY Law J. Online* (March 8, 2011), www.newyorklawjournal.com/id=1202484668508/Applicability-of-theUCC-to-Software-Transactions (acknowledging that art. 2 does not explicitly mention software) (on file with the *Washington & Lee Law Review*); see *Olcott Int'l & Co.* v. *Micro Data Base Sys., Inc.*, 793 NE.2d 1063, 1071 (Ind. Ct. App. 2003) (applying art. 2 to contracts to purchase pre-existing software modules); see also *Advent Sys. Ltd.* v. *Unisys Corp.*, 925 F.2d 670, 676 (3rd Cir. 1991) (identifying the benefits of applying the UCC to computer software transactions).

If classified as a product, strict product liability is imposed.[16] If classified as a service, professional misconduct is imposed, so the distinction is important for a law of algorithms, especially as algorithms become more embedded within society in decision-making roles through different software platforms.

For algorithms, the above examples indicate that assigning liability when damages occur is a major topic of discussion within the legal community and for courts to consider when litigating disputes.[17] As an example, if a programmer writes an algorithm and its use results in damage to property or harm to humans, the responsible party can be traced back to a human who wrote the code. In this case, the lines of code will reveal (at least as circumstantial evidence) the thinking, or intent, of the programmer and once the rules of civil procedure for proffering evidence are satisfied, the code could be used as evidence in a court proceeding. But if the algorithm is generated using deep learning techniques, a person may not have been directly involved in writing the algorithm that ultimately resulted in damages, who then should courts hold liable? The above examples represent just a few of the fundamental legal issues involving algorithms that the law must address as systems controlled by algorithms become more common in society and weave their way into our judicial system. Other chapters in this volume contain additional examples covering a wide range of applications.

SOME EXAMPLE DISPUTES INVOLVING ALGORITHMS

As mentioned previously, there are disputes that have already been litigated which "one way or another" involve algorithms; from this, we argue that we are in the beginning stages of witnessing a law of algorithms develop. As a major classification of algorithms challenging established law, perhaps we can distinguish between algorithms that are derived from biological processes (for example, the instructions provided by DNA) versus algorithms designed to control "human-made" systems (for example, autonomous cars). When "biology" is involved, a common question is whether it is possible to patent the biological process, which is often describable as an algorithm.[18] When human-made systems are involved, the legal issues normally revolve around commercial and tort law which applies to the algorithm as a decision-maker. We present a few examples of statutes and case law below involving algorithms for illustrative purposes, but before that, we next summarize the work being done in the general area of "biological algorithms."

For years, scientists have been working to discover the "computer-like" computations performed by cells.[19] For example, it is thought that cells store information in a way roughly approximating computer memory and that cells use rule-based expressions similar to a programming language in response to stimuli.[20] Each cell contains enough physical complexity to theoretically be quite a powerful computing unit all on its own, but each is also small enough to be packed by the millions into tiny physical spaces. Researchers at MIT and Caltech have been designing cellular machines that are currently able to perform simple

[16] The elements of a strict liability tort are similar to the elements of a negligent tort (duty, breach, and injury), except that in a strict liability case, the victim doesn't need to prove negligence.

[17] W. Barfield, Liability for Autonomous and Artificially Intelligent Robots (2018) 9 *Paladyn* 193.

[18] *Association for Molecular Pathology* v. *Myriad Genetics, Inc.*, 133 S.Ct. 2107 (2013).

[19] G. Templeton, How MIT's New Biological "Computer" Works, and What It Could Do in the Future, *Extreme Tech* (2016), www.extremetech.com/extreme/232190-how-mits-new-biological-computer-works-and-what-it-could-do-in-the-future.

[20] J. Windmiller, Molecular Scale Biocomputing: An Enzyme Logic Approach (thesis, University California, San Diego, June 2012).

computational operations and store, then recall, memory. One major advantage of biological computing compared to modern computers is power efficiency.[21] Running AI algorithms takes many gigawatt-hours of electricity, but extremely long and complex problems could end up being vastly more affordable to solve using biology-based computers. So even though a biological computer is orders of magnitude slower than a supercomputer, supercomputers cost millions of dollars in energy every year, while a bio-computer runs on far less energy. In addition, biological computers differ from non-biological computers in what exactly serves as the output signal. In bio-computers, the presence or concentration of certain chemicals serves as the output signal. Further, biologically inspired computers rely on the nature of specific molecules to adopt certain physical configurations under certain chemical conditions. As systems using biological-based computers become more prevalent, it will be interesting to see what laws are challenged and how the courts respond to such challenges.

A current statute which applies to algorithms, albeit indirectly, is the Biometric Information Privacy Act (BIPA), which was passed by the Illinois General Assembly in 2008.[22] The BIPA guards against the unlawful collection and storing of biometric information. What's particularly relevant for this chapter is that the collection and analysis of biometric information is done by algorithms. So, while algorithms are not directly mentioned in the language of the statute, if algorithms were not directly involved in collecting and analyzing biometric data, the statute would not be necessary. The BIPA remains the only state law in the United States dealing with biometric data that allows private individuals to file a lawsuit for damages stemming from a violation.[23] Because of this damages provision, the BIPA has spawned many class action lawsuits; a few examples follow.

With *In re. Facebook Biometric Info. Privacy Litig.*,[24] Illinois Facebook users alleged that the social media platform violated the BIPA when it used algorithms to scan images of their faces, without consent, in order to run its Tag Suggestions feature. Additionally, in *Monroy* v. *Shutterfly, Inc.*,[25] Shutterfly users claimed that the company violated the BIPA when it scanned uploaded digital photos using algorithmic-based facial recognition software. And in *Rivera* v. *Google, Inc.*,[26] Google users sued the company for violating the BIPA, alleging that it created and stored scans of users' faces on its Google Photos service, without user consent; however, in 2018, the lawsuit was dismissed for lack of standing. In a state case, *Rosenbach* v. *Six Flags Entm't Corp.*,[27] Six Flags was sued for collecting park-goers' thumbprints without informed consent. The Illinois Court of Appeals ruled that a mere technical violation of the BIPA was insufficient to maintain an action, because it did not necessarily mean a party was "aggrieved," as required by the statute. This was reversed by the Illinois Supreme Court, which ruled that users do not need to prove an injury (such as identity fraud or physical harm) in order to sue; the mere violation of the act was sufficient to collect damages.

There are a number of cases which have sought, inter alia, to determine whether an algorithm is a product of nature and thus not patentable subject matter. In *Mackay Radio & Telegraph Co.* v. *Radio Corp. of America*,[28] the applicant sought a patent on a directional

[21] R. T. Gonzalez, This New Discovery Will Finally Allow Us to Build Biological Computers, *Gizmodo* (March 29, 2013), https://io9.gizmodo.com/this-new-discovery-will-finally-allow-us-to-build-biolo-462867996.
[22] Codified as 740 ILCS/14 and Public Act 095-994.
[23] The Act prescribes $1,000 per violation, and $5,000 per violation if the violation is intentional or reckless.
[24] See above note 2.
[25] *Monroy* v. *Shutterfly, Inc.*, No. 16 C 10984, 2017 WL 4099846 (ND Ill. September 15, 2017).
[26] *Rivera* v. *Google, Inc.*, 238 F. Supp. 3d 1088 (ND Ill. 2017).
[27] *Rosenbach* v. *Six Flags Entm't Corp.*, 2017 IL App. (2d) (May Term 2018).
[28] *Mackay Radio & Telegraph Co.* v. *Radio Corp. of America*, 306 US 333, 1938.

antenna system in which the wire arrangement was determined by the application of a mathematical formula. Putting the question of patentability to one side as a preface to his analysis of the infringement issue, Mr. Justice Stone, writing for the Court, explained: "While a scientific truth, or the mathematical expression of it, is not a patentable invention, a novel and useful structure created with the aid of knowledge of scientific truth may be."[29] *Funk Bros. Seed Co.* v. *Kalo Co.*[30] expressed a similar approach: "He who discovers a hitherto unknown phenomenon of nature has no claim to a monopoly of it which the law recognizes. If there is to be invention from such a discovery, it must come from the application of the law of nature to a new and useful end."[31] *Mackay Radio* and *Funk Bros.* point to the proper analysis of patent cases involving algorithms: the process itself, not merely the mathematical algorithm, must be new and useful. Indeed, the novelty of the mathematical algorithm is not a determining factor at all. Whether the algorithm was in fact known or unknown at the time of the claimed invention, as one of the "basic tools of scientific and technological work,"[32] is treated as though it were a familiar part of the prior art.

As an example of a case involving algorithms based on the patentability of biological processes, consider *Association for Molecular Pathology* v. *Myriad Genetics, Inc.*[33] Before we present the case, we will make a short digression to discuss the idea that the instructions provided by DNA can be described as an algorithm. We start by noting that proteins are essential building blocks for life and the instructions to create proteins are encoded in DNA sequences. These sequences can be written as algorithms and in fact operate like algorithms. To learn from nature, reverse engineering the adaptable operating systems of biological organisms has long been a goal of computer scientists.[34] Neural nets, genetic algorithms, and cellular automata all attempt to reproduce the elegance of biological systems in silicon. Using the quaternary logic of DNA base-pairing, in 1994, Leonard Adleman showed how a computationally "hard" problem could be solved using techniques from molecular biology.[35] While conventional computers attack problems *via* large calculations in series, properly encoded "molecular computers" might quickly solve the same problems by simultaneously carrying out billions of operations in parallel. We expect algorithms which originate from biological processes to become more involved in controlling systems and thus will have the potential to challenge established areas of law. We note again that such a development will motivate the need for a specialized court to resolve disputes involving entities controlled by algorithms.

Returning to *Association for Molecular Pathology*, respondent Myriad Genetics, Inc. (Myriad) discovered the precise location and sequence of two human genes, mutations of which can increase the risks of breast and ovarian cancer. Myriad obtained a number of patents based upon its discovery. The case involved claims from three of the patents and required the Court to resolve whether a naturally occurring segment of DNA (which we discuss as an algorithm) is patent-eligible under 35 USC § 101 by virtue of its isolation from the rest of the human genome. Section 101 of the US Patent Act provides: "Whoever invents or discovers any new and useful . . . composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title." The Court also addressed the patent eligibility of synthetically created DNA known as

29 *Ibid.*
30 *Funk Bros. Seed Co.* v. *Kalo Co.*, 333 US 127,1948.
31 *Ibid.*
32 See *Gottschalk* v. *Benson*, 409 US 63, 67 (1972).
33 See above note 18.
34 B. R. Donald, *Algorithms in Structural Molecular Biology, Computational Molecular Biology* (MIT Press, 2011).
35 L. M. Adleman, Molecular Computation of Solutions to Combinatorial Problems (1994) 266 *Science* 1021.