Artificial Intelligence for Ecological and Evolutionary Synthesis

Philippe Desjardins-Proulx[0,1,2], Timothée Poisot[2], and Dominique Gravel[1]

[0] *email: philippe.d.proulx@gmail.com*
[1] *Université de Sherbrooke, Canada.*
[2] *Université de Montréal, Canada.*

September 21, 2019

Abstract

The grand ambition of theorists studying ecology and evolution is to discover the logical and mathematical rules driving the world's biodiversity at every level from genetic diversity within species to differences between populations, communities, and ecosystems. This ambition has been difficult to realize in great part because of the complexity of biodiversity. Theoretical work has led to a complex web of theories, each having non-obvious consequences for other theories. Case in point, the recent realization that genetic diversity involves a great deal of temporal and spatial stochasticity forces theoretical population genetics to consider abiotic and biotic factors generally reserved to ecosystem ecology. This interconnectedness may require theoretical scientists to adopt new techniques adapted to reason about large sets of theories. Mathematicians have solved this problem by using formal languages based on logic to manage theorems. However, theories in ecology and evolution are not mathematical theorems, they involve uncertainty. Recent work in Artificial Intelligence in bridging logic and probability theory offers the opportunity to build rich knowledge bases that combine logic's ability to represent complex mathematical ideas with probability theory's ability to model uncertainty. We describe these hybrid languages and explore how they could be used to build a unified knowledge base of theories for ecology and evolution.

case study you explore using the Salix tritrophic system.

Keywords*: Artificial Intelligence, Theoretical Biology, Theoretical Ecology, Evolution, Theoretical Population Genetics, Machine learning, Knowledge Representation.*

0 Introduction

Almost four decades ago, Ralph W. Lewis argued for the formalization of evolutionary theory and the recognition of evolution as a system of theories. In his words, "when theories are partially formalized [...] the intra- and interworkings of theories become more clearly visible, and the total structure of the discipline becomes more evident" [45]. Supporting Lewis' point, Queller recently showed how Fisher's fundamental theorem of natural selection, Price's theorem, the Breeder equation of quantitative genetics, and other key formulas in ecology and evolution were related [62]. In the same vein, Rice formulated an axiomatic theory of evolution based on a stochastic version of Price's theorem [63]. These projects fall under the scope of automated theorem proving, one of the oldest and most mature branches of Artificial Intelligence [31]. Theories can be written in some formal language, such as first-order logic or type theory, and then algorithms are used to ensure the theories can be derived from a knowledge base of axioms and existing results. In the last few decades, mathematicians have built knowledge bases with millions of helper theorems to assist the discovery of new ideas [39]. For example, the Mizar Mathematical Library is a growing library of theorems, which are added after new candidate theorems are approved by the proof checker and peer-reviewed for style. Such libraries help mathematicians juggle with a growing body of knowledge and offers a concrete answer to the issue of knowledge synthesis. Mizar uses a language powerful

enough for the formalization of evolutionary theories envisioned by Lewis and the result of Queller on Price's theorem and its relationship to other theories. It is also expressive enough to build a knowledge base out of Rice's axiomatic theory of evolution. Doing so would force us to think more clearly about the theoretical structure of evolution, with theoretical ecology facing a similar state of disorganization [45]. Case in point: theoretical community ecologists have been criticized for focusing on a single prediction for theories capable of making several [49]. An example of this is Hubbell's neutral theory of biodiversity [36], which uses an unrealistic point-mutation model that does not fit with our knowledge of speciation, leading to odd predictions [23, 19, 20]. In logic-based (also called symbolic) systems like Mizar, all formulas involving speciation would be implicitly linked together. Storing ecological theories in such knowledge base would automatically prevent inconsistencies and highlight the consequences of theories.

Despite the importance of formalization, it remains somewhat divorced from an essential aspect of theories in ecology and evolution: their probabilistic and fuzzy nature. As a few examples: a surprisingly common idea found in ecological theories is that predators are generally larger than their prey, a key assumption of the food web model of Williams and Martinez [76]; deviations from the Hardy–Weinberg principle are not only common but tend to give important information on selective pressures; and nobody expects the Rosenzweig–MacArthur predator-prey model to be exactly right. In short, important ideas in ecology and evolution do not fit the true/false epistemological framework of systems like Mizar, and ideas do not need to be derived from axiomatic principles to be useful. We are often less concerned by whether a formula can be derived from axioms than in how it fits a particular dataset. In the 1980s, Artificial Intelligence experts developed probabilistic graphical models to handle large probabilistic systems [57]. While probabilistic graphical models are capable of answering probabilistic queries for large systems of variables, they cannot represent or reason with sophisticated mathematical formulas. Alone, neither logic nor probability theory is enough to elucidate the structure of theories in ecology and evolution.

For decades, researchers have tried to unify probability theory with rich logics to build knowledge bases both capable of the sophisticated mathematical reasoning found in automated theorem provers and the probabilistic reasoning of graphical models. Recent advances moved us closer to that goal [64, 27, 74, 54, 35, 70, 4]. Using these systems, it is possible to check if a mathematical formula can be derived from existing results and also possible to ask probabilistic queries about theories and data. The probabilistic nature of these representations is a good fit to learn complex logical and mathematical formulas from data [41]. Within this framework, there is no longer a sharp distinction between *theory* and *data*, since the knowledge base defines a probability distribution over all objects, including logical relationships and mathematical formulas.

For this article, we introduce key ideas on methods at the frontier of logic and probability, beginning with a short survey of knowledge representations based on logic and probability. First-order logic is described, along with how it can be used in a probabilistic setting with Markov logic networks [64]. We detail how theories in ecology and evolution can be represented with Markov logic networks, as well as highlighting some limitations. We present a case study involving a tritrophic system to demonstrate the strengths and weaknesses of Markov logic networks. Synthesis in ecology and evolution has been made difficult by the sheer number of theories involved and their complex relationships [59]. Practical representations to unify logic and probability are relatively new, but we argue they could be used to achieve greater synthesis by allowing the construction of large, flexible knowledge bases with a mix of mathematical and scientific knowledge.

1 Knowledge representations

Traditional scientific theories and models are mathematical, or logic-based. Einstein's $e = mc^2$ established a relationship between energy e, mass m, and the speed of light c. This mathematical knowledge can be reused: in any equation with energy, we could replace e with mc^2. This ability of mathematical theories to establish precise relationships between concepts, which can then be used as foundations for other theories, is fundamental to how science grows and forms an interconnected corpus of knowledge. The formula is implicitly connected to other formulas involving the same symbol, such that if we were to establish a different but equivalent way to represent the speed of light c, it could automatically substitute

c in $e = mc^2$.

Artificial Intelligence researchers have long been interested in expert systems capable of scientific discoveries, or simply capable of storing scientific and medical knowledge in a single coherent system. *Dendral*, arguably the first expert system, could form hypotheses to help identify new molecules using its knowledge of chemistry [46]. In the 1980s, *Mycin* was used to diagnose blood infections (and did so more accurately than professionals) [14]. Both systems were based on logic, with *Mycin* adding a "confidence factor" to its rules to model uncertainty. These expert systems generally relied on a simple logic system not powerful enough to handle uncertainty. With few exceptions, the rules were hand-crafted by human experts. After the experts established the logic formulas, the systems acted as static knowledge bases and unable to discover new rules. Algorithms have been developed to learn new logic rules from data [52, 53], but the non-probabilistic nature of the resulting knowledge base makes it difficult to handle real-world uncertainty. In addition to expert systems, logic systems are used to store mathematical knowledge and perform automatic theorem proving [31]. Pure logic has rarely been used in ecology and evolution, but recent studies have shown its ability to reconstruct food webs from data [10, 72].

There are many different logics for expert systems and automatic theorem proving [31, 61, 56]. We will focus on first-order logic, the most commonly used logic in efforts to unify logic with probability. A major reason for adopting rich logics, whether first-order or higher-order, is to allow for the complex relationships found in ecology and evolution to be expressed in concise formulas. Stuart Russell noted that "the rules of chess occupy 10^0 pages in first-order logic, 10^5 pages in propositional logic, and 10^{38} pages in the language of finite automata" [65]. Similarly, first-order logic allows us to directly express complex ecological ideas in a simple but formal language.

In mathematics, a function f maps terms $\mathbf{X}$ (its domain) to other terms $\mathbf{Y}$ (its codomain) $f : \mathbf{X} \rightarrow \mathbf{Y}$. The number of arguments of a function, $|\mathbf{X}|$, is called its arity. The atomic element of first-order logic is the **predicate**: a function that maps 0 or more terms to a truth value: false or true. In first-order logic, terms are either variables, constants, or functions. A **variable** ranges over a domain, for example x could range over integers, p over a set of species, and *city* over a set of cities. **Constants** represent values such as 42, $Manila$, π. Lastly, **functions** map terms to other terms such as multiplication, integration, sin, $CapitalOf$ (mapping a country to its capital). Variables have to be **quantified** either universally with $\forall$ (forall), existentially with $\exists$ (exists), or uniquely with $\exists!$. $\forall x : p(x)$ means $p(x)$ must hold true for all possible values of x. $\exists x : p(x)$ means it must hold for at least one value of x while $\exists! x : p(x)$ means it must hold for exactly one value of x. Using this formal notation, we can write the relationship between the basal metabolic rate (BMR) and body mass ($Mass$) for mammals [2]:

$$\forall m \in Mammal : BMR(m) = 4.1 \times Mass(m)^{0.75}. \tag{1}$$

This formula has one variable m which is universally quantified: $\forall m \in Mammal$ reads "for all m in the set $Mammal$". It has two constants: the numbers 4.1 and 0.75, along with four functions (BMR, $Mass$, multiplication, exponentiation). The equal sign $=$ is the sole predicate.

A first-order logic **formula** is either a lone predicate or a complex formula formed by linking formulas using the unary connective $\neg$ (negation) or binary connectives (*and* $\wedge$, *or* $\vee$, *implication* $\Rightarrow$, see table 1). For example, $PreyOn(s_x, s_y)$ is a predicate that maps two species to a truth value, in this case whether the first species preys on the second species, and $IsParasite(s)$ is a predicate that is true if species s is a parasite. We could also have a function $Mass(s_x)$ mapping a species to its body mass. We can build more complex formulas from there, for example:

$$\forall s_x : \neg PreyOn(s_x, s_x). \tag{2a}$$

$$\forall s_x, s_y : PreyOn(s_x, s_y) \Rightarrow Mass(s_x) > Mass(s_y). \tag{2b}$$

$$\forall s_x, s_y : PreyOn(s_x, s_y) \wedge \neg IsParasite(s_x) \Rightarrow Mass(s_x) > Mass(s_y). \tag{2c}$$

The first formula says that species don't prey on themselves. The second formula says that predators are larger than their prey ($>$ is a shorthand for the *greater than* predicate). The third formula refines the second one by adding that predators are larger than their prey unless the predator is a parasite. None of

these rules are expected to be true all the time, which is where mixing probability with logic will come in handy. The Rosenzweig-MacArthur equation can also easily be expressed with first-order logic:

$$\forall x, y : \dot{x} = r_0 \left(1 - \frac{x}{K}\right) - \frac{Cxy}{D+x} \wedge \dot{y} = X\frac{Cxy}{D+x} - \delta_0 y. \tag{3}$$

This formula has four functions: the time differential $\dot{x} \equiv dx/dt$, multiplication, addition, and subtraction. Prey x and predator y are universally quantified variables while $r_0, K, C, D, X, \delta_0$ are constants. The formula has only one predicate, $=$, and both sides of the formula are connected by $\wedge$, the symbol for conjunction ("and").

A **knowledge base** $\mathcal{K}$ in first-order logic is a set of formulas $\mathcal{K} = \{f_0, f_1, ..., f_{|\mathcal{K}|-1}\}$. First-order logic is expressive enough to represent and manipulate complex logic and mathematical ideas. It can be used for simple ideas such that predators are generally larger than their prey (eq. 2b), mathematical formulas for predator-prey systems equation (eq. 3), and also to establish the logical relationship between various predicates. We may want a $PreyOn$ predicate to tell us whether s_x preys on s_y, but also a narrower $PreyOnAt(s_x, s_y, l)$ predicate to model whether s_x preys on s_y at a specific location l. In this case, it would be a good idea to have the formula $\forall s_x, s_y, l : PreyOnAt(s_x, s_y, l) \Rightarrow PreyOn(s_x, s_y)$. Given this formula and the data point $PreyOnAt(Wolverine, Rabbit, Quebec)$, we do not need $PreyOn(Wolverine, Rabbit)$ to be explicitly stated, ensuring the larger metaweb [60] is always consistent with information from local food webs.

An **interpretation** defines which object, predicate, or function is represented by which symbol, e.g., it says $PreyOnAt$ is a predicate with three arguments, two species and one location. The process of replacing variables with constants is called **grounding**, and we talk of ground terms / predicates / formulas when no variables are present. Together with an interpretation, a **possible world** assigns truth values to each possible ground predicate, which can then be used to assign truth values to a knowledge base's formulas. $PreyOn(s_x, s_y)$ can be neither true nor false until we assign constants to the variables s_x and s_y. Constants are typed, so a set of constants $\mathcal{C}$ may include two species $\{Gulo\ gulo, Orcinus\ orca\}$ and three locations $\{Quebec, Fukuoka, Arrakis\}$. The constants $\mathcal{C}$ yield $2^2 \times 3$ possible ground predicates for $PreyOnAt(s_x, s_y, l)$:

$$PreyOnAt(Gulo\ gulo, Gulo\ gulo, Quebec)$$
$$PreyOnAt(Gulo\ gulo, Orcinus\ orca, Quebec)$$
$$PreyOnAt(Orcinus\ orca, Orcinus\ orca, Quebec)$$
$$PreyOnAt(Orcinus\ orca, Gulo\ gulo, Quebec)$$
$$PreyOnAt(Gulo\ gulo, Gulo\ gulo, Fukuoka)$$
$$\dots$$

and only two possible ground predicates for $IsParasite$:

$$IsParasite(Gulo\ gulo)$$
$$IsParasite(Orcinus\ orca)$$

We say a possible world **satisfies** a knowledge base (or a single formula) if all the formulas are true given the ground predicates. A basic question in first-order logic is to determine whether a knowledge base $\mathcal{K}$ **entails** a formula f, or $\mathcal{K} \models f$. Formally, the entailment $\mathcal{K} \models f$ means that for all possible worlds in which all formulas in $\mathcal{K}$ are true, f is also true. More intuitively, it can be read as the formula *following from* the knowledge base [66]. A **proof** in first-order logic can be derived using **inference rules** such as **Modus Ponens**:

$$\frac{\alpha \Rightarrow \beta \quad \alpha}{\beta}. \tag{6}$$

Name	Common	Symbol	Truth table			
			T x T	T x F	F x T	F x F
Conjunction	and	$\wedge$	T	F	F	F
Disjunction	or	$\vee$	T	T	T	F
Implication	implies	$\Rightarrow$	T	F	T	T
Material equivalence	iff	$\Leftrightarrow$	T	F	F	T
Exclusive disjunction	xor	$\veebar$	F	T	T	F

Table 1: Common binary connectives. The table shows the resulting truth value (T: True, F: False) for all possible combinations. *iff* is read *if and only if*. Implication is one of the most common connective and may have surprising behavior. In particular, it will always return true when the left-side is false. While this may seem odd, it allows us to make statements such as $\forall x \in \mathbb{R} : x \geq 0 \Rightarrow \sqrt{x^2} = x$. This formula holds for all real numbers, including negative ones, since with $x = -1$, $x \geq 0$ is false and $F \Rightarrow F$ returns true.

This notation reads: infer β if $\alpha \Rightarrow \beta$ is true and α is true. See [31] for a detailed look at inference rules in first-order logic.

Probabilistic graphical models, which combine graph theory with probability theory to represent complex probability distributions, can provide an alternative to logic-based representations [42, 6]. There are primarily two motivations behind probabilistic graphical models. First, even for binary random variables, we need to learn $2^n - 1$ parameters for a distribution of n variables. This is unmanageable on many levels: it is computationally difficult to do inference with so many parameters, requires a large amount of memory, and makes it difficult to learn parameters without an unreasonable volume of data [42]. Second, probabilistic graphical models provide important information about independences and the overall structure of the distribution. Probabilistic graphical models were also used as expert systems: *Munin* had a network of more than 1000 nodes to analyze electromyographic data [22], while *PathFinder* assisted medical professionals for the diagnostic of lymph-node pathologies [33] (Figure 1).

The two key inference problems in probabilistic machine learning are finding the most probable joint state of the unobserved variables (maximum a posteriori, or MAP) and computing conditional probabilities (conditional inference). In a simple presence/absence model for 10 species $(s_0, s_1, ..., s_9)$, given that we know the state of species $s_0 = Present, s_1 = Absent, s_2 = Absent$, MAP inference would tell us the most likely state for species $s_3, ..., s_9$, while conditional inference could answer queries such as $P(s_4 = Absent | s_0 = Present)$.

2 Markov logic

At this point we have first-order logic, which is capable of manipulating complex logic and mathematical formulas but cannot handle uncertainty, and probabilistic graphical models, which cannot be used to represent mathematical formulas (and thus theories in ecology and evolution) but can handle uncertainty. The limit of first-order logic can be illustrated with our previous example: predators generally have a larger body weight ($Mass$) than their prey, which we expressed in predicate logic as $\forall s_x, s_y : PreyOn(s_x, s_y) \Rightarrow Mass(s_x) > Mass(s_y)$, but this is obviously false for some assignments such as $s_x : grey\ wolf$ and $s_y : moose$. However, it is still useful knowledge that underpins many ecological theories [76]. When our domain involves a great number of variables, we should expect useful rules and formulas that are not always true.

A core idea behind many efforts to unify rich logics with probability theory is that formulas can be weighted, with higher values meaning we have greater certainty in the formula. In pure logic, it is impossible to violate a single formula. With weighted formulas, an assignment of concrete values to variables is only *less likely* if it violates formulas, and how much less likely will depend on the weight assigned to the violated formula. The higher the weight of the formula violated, the less likely the assignment is. It is conjectured that all perfect numbers are even ($\forall x : Perfect(x) \Rightarrow Even(x)$), thus, if we were to find a single odd perfect number, that formula would be refuted. It makes sense for

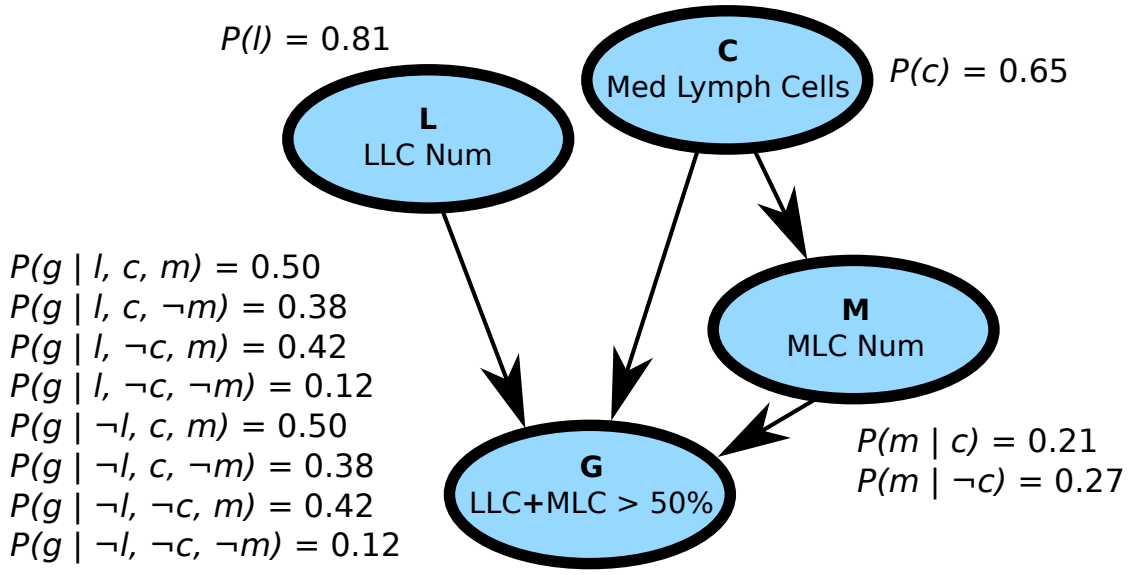

Figure 1: A Bayesian network with four binary variables (the vertices) and possible conditional probability tables. Bayesian networks encode the distribution as directed acyclic graphs such that $P(\mathbf{X} = \mathbf{x}) = \prod_i P(x_i|Pa(x_i))$, where $Pa(x_i)$ is the set of parents of variable x_i. Because no cycles are allowed, the variables form an ordering so the set $Pa(x_i)$ can only involve variables already seen on the left of x_i. Thus, $P(a)P(b|a)P(c)$ is a valid Bayesian networks but not $P(a)P(b|c)P(c|b)$. The four vertices represented here were extracted from *PathFinder*, a Bayesian network with more than 1000 vertices used to help diagnose blood infections [33]. The vertices represent four variables related to blood cells and are denoted by a single character (in bold in the figure): C, M, L, G. We denote a positive value with a lowercase letter and a negative value with $\neg$ (e.g.: $C = c$, $M = \neg m$). Since $P(\neg x|\mathbf{y}) = 1 - P(x|\mathbf{y})$, we need only $2^{|Pa(x)|}$ parameters per vertex, with $|Pa(x)|$ being the number of parents of vertex x. The structure of Bayesian networks highlights the conditional independence assumptions of the distribution and reduces the number of parameters for learning and inference. As a example query: $P(l, \neg c, m, \neg g) = P(l)P(\neg c)P(m|\neg c)P(\neg g|l, \neg c, m) = 0.81 \times (1 - 0.65) \times 0.27 \times (1 - 0.42) = 0.044$. See [17] for a detailed treatment of Bayesian networks and [42] for a more general reference on probabilistic graphical models.

mathematics but for many disciplines, such as biology, important principles are only expected to be true *most* of the time. If we were to find a single predator smaller than its prey, it would definitely not make our rule useless.

The idea of weighted formulas is not new. Markov logic networks (or just Markov logic), invented a decade ago, allows for logic formulas to be weighted [64, 21]. Similar efforts also use weighted formulas [4, 35]. Markov logic supports algorithms to add weights to existing formulas given a dataset, learn new formulas or revise existing ones, and answer probabilistic queries (MAP or conditional). As a case study, Yoshikawa et al. used Markov logic to understand how events in a document were time-related [78]. Their research is a good case study of interaction between traditional theory-making and artificial intelligence. The formulas they used as a starting point were well-established logic rules to understand temporal expressions. From there, they used Markov logic to weight the rules, adding enough flexibility to their system to beat the best approach of the time. Brouard et al. [13] used Markov logic to understand gene regulatory networks, noting how the resulting model provided clear insights, in contrast to more traditional machine learning techniques. Markov logic greatly simplifies the process of growing a base of knowledge: two research labs with different knowledge bases can simply put all their formulas in a single knowledge base. The only steps required to merge two knowledge bases is to put all the formulas in a single knowledge base and reevaluate the weights.

In a nutshell, a knowledge base in Markov logic $\mathcal{M}$ is a set of formulas $f_0, f_1, f_2, ...$ along with their weights $w_0, w_1, w_2, \ldots$:

$$\mathcal{M} = \{(f_0, w_0), (f_1, w_1), ..., (f_{|\mathcal{M}|-1}, w_{|\mathcal{M}|-1})\}. \tag{7}$$

Given constants $\mathcal{C} = \{c_0, c_1, \ldots, c_{|\mathcal{C}|-1}\}$, $\mathcal{M}$ defines a Markov network (an undirected probabilistic graphical model) which can be used to answer probabilistic queries. **Weights** are real numbers in the $-\infty, \infty$ range. The intuition is: the higher the weight associated with a formula, the greater the penalty for violating it (or alternatively: the less likely a possible world is). The **cost** of an assignment is the sum of the weights of the unsatisfied formulas (those that are false). The higher the cost, the less likely the assignment is. Thus, if a variable assignment violates 12 times a formula with a weight of 0.1 and once a formula with a weight of 1.1, while another variable assignment violates a single formula with a weight of 5, the first assignment has a higher likelihood (cost of 2.3 vs 5). Formulas with an infinite weight act like formulas in pure logic: they cannot be violated without setting the probabilities to 0. In short, a knowledge base in pure first-order logic is exactly the same as a knowledge base in Markov logic where all the weights are infinite. In practice, it means mathematical ideas and axioms can easily be added to Markov logic as formulas with an infinite weight. Formulas with weights close to 0 have little effect on the probabilities and the cost of violating them is small. A formula with a negative weight is expected to be false. It is often assumed that all weights are positive real numbers without loss of generality since $(f, -w) \equiv (\neg f, w)$. See Jain [38] for a detailed treatment of knowledge engineering with Markov logic. Markov logic can answer queries of complex formulas of the form:

$$P(f_0|f_1, \mathcal{M}, \mathcal{C}) = \frac{P(f_0 \wedge f_1|\mathcal{M}, \mathcal{C})}{P(f_1|\mathcal{M}, \mathcal{C})}, \tag{8}$$

where f_0 and f_1 are first-order logic formulas while $\mathcal{M}$ is a weighted knowledge base and $\mathcal{C}$ a set of constants. It's important to note that neither f_0 nor f_1 need to be in $\mathcal{M}$. Logical entailment $\mathcal{M} \models f$ is equivalent to finding $P(f|\mathcal{M}) = 1$ [21].

We build a small knowledge base for an established ecological theory: the niche model of trophic interactions [76]. The first iteration of the niche model posits that all species are described by a niche position N (their body size for instance) in the $[0, 1]$ interval, a diet D in the $[0, N]$ interval, and a range R such that a species preys on all species with a niche in the $[D - R/2, D + R/2]$ interval. We can represent these ideas with three formulas:

$$\forall x, y : \neg PreyOn(x, y), \tag{9a}$$

$$\forall x : D(x) < N(x), \tag{9b}$$

Connective	Łukasiewicz	Logic Gödel-Dummett	Product
$x \wedge y$	$max(0, x+y-1)$	$min(x, y)$	$x \times y$
$x \vee y$	$min(1, x+y)$	$max(x, y)$	$x + y - x \times y$
$x \Rightarrow y$	$min(1, 1-x+y)$	1 if $x \leq y$, y otherwise	1 if $x \leq y$, y/x otherwise
$\neg x$	$1-x$	0 if $x > 0$, 1 otherwise	0 if $x > 0$, 1 otherwise

Table 2: Definitions of logic connectives for the three main fuzzy logics. These three logics are said to be normal, meaning they behave exactly like classical logic when restricted to truth values of 0 (false) and 1 (true). When truth values are between 0 and 1, these logics will often behave differently than classical logic. For example, in both classical and Łukasiewicz logics, $\neg\neg x \equiv x$, but it is not the case for Gödel-Dummett and Product logics (unless $x \in \{0, 1\}$). Another example is that conjunction and disjunction are idempotent in classical and Gödel-Dummett logics, meaning $x \wedge x \equiv x$ and $x \vee x \equiv x$, but it is not the case for Łukasiewicz and Product logics. See [7] for a detailed explanation of how the connectives are defined.

$$\forall x, y : PreyOn(x, y) \Leftrightarrow D(x) - R(x)/2 < N(y) \wedge N(y) < D(x) + R(x)/2, \tag{9c}$$

As pure logic, this knowledge base makes little sense. Formula 9a is obviously not true all the time but often is since most pairs of species do not interact. In Markov logic, it is common to have a formula for each lone predicate, painting a rough picture of its marginal probability [21, 38]. We could also add that cannibalism is rare $\forall x : \neg PreyOn(x, x)$ and that predator-prey relationships are generally asymmetrical $\forall x, y : PreyOn(x, y) \Rightarrow \neg PreyOn(y, x)$ (although this formula is redundant with the idea that predators are generally larger than their prey). Formulas that are often wrong are assigned a lower weight but can still provide useful information about the system. The second formula says that the diet is smaller than the niche value. The last formula is the niche model: species x preys on y if and only if species y's niche is within the diet interval of x. Using Markov logic and a dataset, we can learn a weight for each formula in the knowledge base. This step alone is useful and provides insights into which formulas hold best in the data. With the resulting weighted knowledge base, we can make probabilistic queries and even attempt to revise the theory automatically. We could find, for example, that the second rule does not apply to parasites or some group and get a revised rule such as $\forall x : \neg IsParasite(x) \Rightarrow D(x) < N(x)$.

3 Fuzziness

First-order logic provides a formal language for expressing mathematical and logical ideas while probability theory provides a framework for reasoning about uncertainty. A third dimension often found in discussions on unifying logic with probability is fuzziness. A struggle with applying logic to ecology is that all predicates are either true or false. Even probabilistic logics like Markov logic define a distribution over *binary* predicates. Going back to Rosenzweig-MacArthur (eq. 3), this formula's weight in Markov logic is almost certainly going to be zero since it's never *exactly* right. If the Rosenzweig-MacArthur equation predicts a population size of 94 and we observe 93, the formula is false. Weighted formulas help us understand *how often a formula is true*, but in the end the formula has to give a binary truth value: true or false, there is no place for nuance. Logicians studied more flexible logics where truth is a real number in the $[0, 1]$ range. These logics are said to be "infinitely many-valued" or "fuzzy". In this setting: 0 is false, 1 is true, and everything in-between is used to denote nuances of truth [79, 7]. Predicates returning truth values in the $[0, 1]$ range are called **fuzzy predicates**, while standard predicates returning $false, true$ are said to be bivalent. To show fuzziness in action, let's look at a simple formula that says that small populations experience exponential growth:

$$\forall s, l, t : SmallPopSize(s, l, t) \Rightarrow N(s, l, t+1) = R(s) \times N(s, l, t). \tag{10}$$

Variables s, l, t respectively denote a species, a location, and time. Function N returns the population size of a species at a specific location and time while function R returns the growth rate of the species.

The predicates $SmallPopSize$ and $=$ are both problematic from a bivalent perspective. Equality poses problem for the same reason it did with the Rosenzweig-MacArthur example: we do not expect this formula to be exactly right. The notion of a small population size should also be flexible, yet logic forces us to determine a strict threshold where $SmallPopSize$ will change from true to false. Using truth values in the $[0, 1]$ range makes it possible to have a wide range of nuances for both $SmallPopSize$ and equality. See table 2 for the definitions of fuzzy logic connectives.

Fuzzy logic is not a replacement for probability theory. The most interesting aspect of fuzzy logic is how it interacts with probability theory to form truly flexible languages. For examples, fuzzy predicates are used in both probabilistic soft logic [40, 4] and deep learning approaches to predicate logic [80, 35]. Hybrid Markov logic [74, 21] extends Markov logic by allowing not only weighted formulas but terms like soft equality, which applies a Gaussian penalty to deviations from equality. While not exactly a full integration of fuzzy logic into Markov logic, soft equality behaves in a similar matter and is a good fit for formulas like the Rosenzweig-MacArthur system or our previous example with exponential growth. Hybrid Markov logic is not as well-developed as standard Markov logic, for example there are no algorithms to learn new formulas from data. On the other hand, Hybrid Markov logic solves many of the problems caused by bivalent predicates while retaining the ability to answer conditional queries. In the next section we'll explore hybrid Markov logic and its application to an ecological dataset. Several languages for reasoning have combined fuzziness with probability or logic (Figure 2). It has been argued that, in the context of Bayesian reasoning, fuzziness plays an important role in bridging logic with probability [55, 37]. However, how to effectively combine rich logics with probability theory remains an open question, as is the role of fuzziness.

4 Markov logic and the Salix tritrophic system

The primary goal of unifying logic and probability is to be able to grow knowledge bases of formulas in a clear, precise language. For Markov logic, it means a set of formulas in first-order logic. For this example, we used Markov logic to build a knowledge base for ecological interactions around the Salix dataset [43]. The Salix dataset has 126 parasites, 96 species of gallers (insects), and 52 species of salix, forming a tritrophic ecological network ($Parasite \rightarrow Galler \rightarrow Salix$). Furthermore, we have partial phylogenetic information for the species, their presence/absence in 374 locations, interactions, and some environmental information on the locations. To fully illustrate the strengths and limits of Markov logic in this setting, we will not limit ourselves to the data available for this particular dataset (e.g. we do not have body mass for all species).

Data in first-order logic can be organized as a set of tables (one for each predicate). For our example, we have a table named $PreyOnAt$ with three columns (its arguments) and a table named $IsParasitoid$ with only one column. This format implies the closed-world assumption: if an entry is not found, it is false (see table 4 for an example). For this problem we defined several functions and predicates to describe everything from predator-prey relationships, whether pairs of species often co-occurred, along with information on locations such as humidity, precipitation, and temperature (see table 3). We ran the basic learning algorithm from Alchemy-2 [64], which is used both to learn new formulas and weight them. The weights are listed at the end of each formula. We use the "?" character at the end of the formula involving data that were unavailable for this dataset (and thus, we could not learn the weight). Here's a sample of a knowledge base where the first three formulas were learned directly from our dataset and the last two serve as example for Hybrid Markov logic:

$$\forall s_0, s_1 : IsGaller(s_0) \wedge PreyOn(s_0, s_1) \Rightarrow IsSalix(s_1), 4.15. \tag{11a}$$

$$\forall s_0, s_1 : IsParasitoid(s_0) \wedge PreyOn(s_0, s_1) \Rightarrow IsGaller(s_1), 3.49. \tag{11b}$$

$$\forall s_0, s_1 : PreyOn(s_0, s_1) \Rightarrow HighCooccurrence(s_0, s_1), 1.57. \tag{11c}$$

$$\forall s_0, s_1, \exists \alpha : PPreyOn(s_0, s_1) \approx \alpha \exp\left(-2(N(s_1) - C(s_0))^2 / R(s_0)\right)? \tag{11d}$$

$$\forall s_0, s_1 : CloselyRelated(s_0, s_1) \wedge T(Occ(s_0)) > T(Occ(s_1)) \Rightarrow Mass(s_0) > Mass(s_1)? \tag{11e}$$

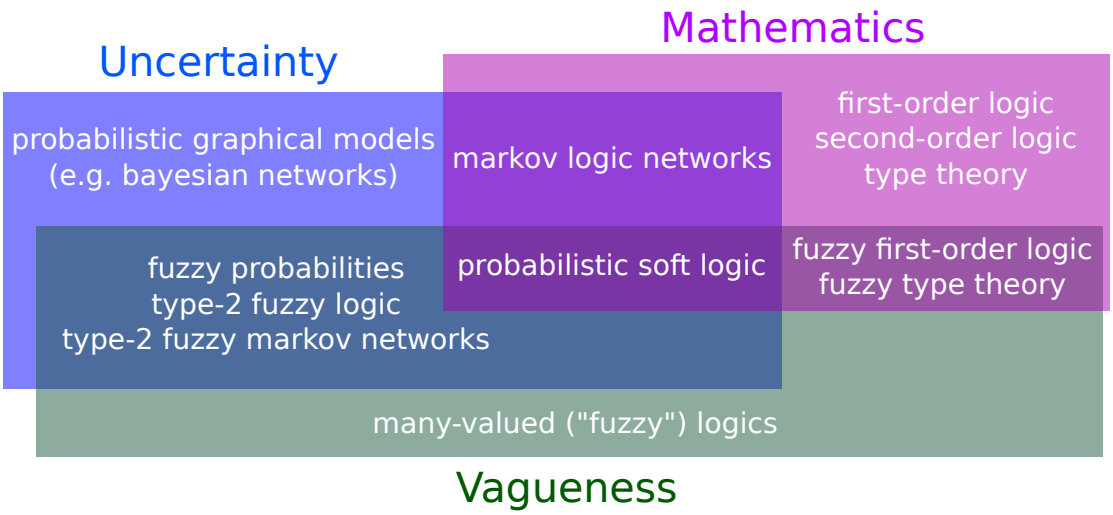

Figure 2: Various languages and their ability to model uncertainty, vagueness, and mathematics (the size of the rectangles has no meaning). **In the blue rectangle**: languages capable of handling uncertainty. Probabilistic graphical models combine probability theory with graph theory to represent complex distributions [42]. Alternatives to probability theory for reasoning about uncertainty include possibility theory and Dempster-Shafer belief functions, see [30] for an extended discussion. **In the green rectangle**: Fuzzy logic extends standard logic by allowing truth values to be anywhere in the $[0, 1]$ interval. Fuzziness models vagueness and is particularly popular in linguistics, engineering, and bioinformatics, where complex concepts and measures tend to be vague by nature. See [44] for a detailed comparison of probability and fuzziness. **In the purple rectangle**: languages capable of modelling mathematical formulas. It is important to note that while first-order logic is expressive enough to express a large class of mathematical ideas, many languages rely on a restricted from of first-order logic without functions. Alone, these languages are not powerful enough to express scientific ideas, we must thus focus on what lies at their intersection. Type-2 Fuzzy Logic is a fast-expanding [67, 50] extension to fuzzy logic, which, in a nutshell, models uncertainty by considering the truth value itself to be fuzzy [51, 81]. Markov logic networks [64, 21] extends predicate logic with weights to unify probability theory with logic. Probabilistic soft logic [40, 3] also has formulas with weights, but allows the predicates to be fuzzy, i.e. have truth values in the $[0, 1]$ interval. Some recent deep learning studies also combine all three aspects [25, 35].

Functions	Meaning
$PPreyOn : species \times species \mapsto [0,1]$	Probability that a species preys on another
$PreyOn : species \times species \mapsto bool$	Predator-prey relationship
$PreyOnAt : species \times species \times location \mapsto bool$	Predator-prey relationship at a given location
$PresenceAt : species \times location \mapsto bool$	Presence of a species at a location
$IsParasite : species \mapsto bool$	Whether the species is a parasite
$IsGaller : species \mapsto bool$	Whether the species is a galler
$IsSalix : species \mapsto bool$	Whether the species is a salix
$CloselyRelated : species \times species \mapsto bool$	Whether two species are closely related
$Occ : species \mapsto \{location\}$	Set of locations where a species is found
$Cooccurrence : species \times species \mapsto \mathbb{R}^+$	Proportion of locations where the species co-occur
$HighCooccurrence : species \times species \mapsto bool$	Pair of species with high co-occurence
$HighTemperature : location \mapsto bool$	Location with above-average temperature
$T : \{location\} \mapsto \mathbb{R}$	Mean temperature for a set of locations
$Mass : species \mapsto \mathbb{R}^+$	Mean adult body mass for a species
$FoodWeb : location \mapsto Graph$	Food web at a given location
$Connectance : Graph \mapsto \mathbb{R}^+$	$Edges/Vertices^2$
$SpeciesRichness : Graph \mapsto \mathbb{N}$	Number of species in the food web
$N : species \mapsto \mathbb{R}^+$	Niche of species per [75]
$C : species \mapsto \mathbb{R}^+$	Diet of the species per [75]
$R : species \mapsto \mathbb{R}^+$	Range of species' diet per [75]

Table 3: Predicates and functions used for the Salix example. A predicate is simply a function mapping to a boolean value (false or true, denoted *bool*). $\mathbb{N}$ stands for natural numbers (0, 1, 2, ...) while $\mathbb{R}$ stands for real numbers, and $[0,1]$ is a shorthand for a real number in the $[0,1]$ range. We must often force continuous values into boolean values. For example, $HighTemperature$ and $CloselyRelated$ both require arbitrary cutoffs, often the line between $true$ and $false$ is set at the mean. Recent languages push for greater integration with fuzziness, which would allow predicates to take any values in the $[0,1]$ range.

$PreyOnAt$		
Amorri	Ovesic	Site060
Chalci	Halien	Site116
Ireuni	Hpolit	Site291
Eacicu	Ovimin	Site121
...	...	...

$IsParasitoid$
Ppecti
Psoemi
Tspone
Tsptwo
...

$HighTemperature$
Site006
Site311
Site296
Site183
...

Table 4: A sample of three tables for the Salix dataset [43]. Species are denoted by the first six letters of their names while sites are numbered from 1 to 374. Data in first-order logic is often organized in tables with one table per predicate and where entries represent true values while absent combinations are assumed to be false. For example, given this sample, $HighTemp(Site006)$ is true while $HighTemp(Site001)$ would be false. The full data formatted for Alchemy-2 [64] is provided as supplementary material.

The first two formulas correctly define the tritrophic relationship between parasites, galler and salix, while the third shows a solid, but not as strong, relationship between predation and co-occurence. Formula 11d would require hybrid Markov logic and a fuzzy predicate $\approx$.

Integration of macroecology and food web ecology may rely on a better understanding of macroecological rules [5]. These rules are easy to express with first-order logic, for example equation 11e is a formulation of Bergmann's rule. We also used the learning algorithm to test whether closely related species had similar prey, but the weight attributed to the formula was almost zero, telling us the formula was right as often as it was wrong:

$$\forall s_0, s_1 : CloselyRelated(s_0, s_1) \wedge PreyOn(s_0, s_2) \Rightarrow PreyOn(s_1, s_2), 0.00. \tag{12}$$

This example shows both the promise and the current issues with hybrid logic-probabilistic techniques. Many of the predicates would benefit from being fuzzy, for example, $PreyOn$ should take different values depending on how often predation occurs. We also had to use arbitrary cut-offs for predicates like $CloselyRelated$ and $HighTemperature$. Fortunately, many recent approaches integrate logic with both fuzziness and probability theory [1, 35, 4]. Weights are useful to understand which relationship is strong in the data, and this example shows the beginning of a knowledge base for food web ecology. The next step would be to discover new formulas, whether manually or using machine learning algorithms, and add data to revise the weights. If a formula involves a predicate operating on food webs and we want to apply our knowledge base to a dataset without food webs, this formula will simply be ignored (because it won't have grounded predicates to evaluate it; see section 1). This is a strong advantage of this knowledge representation: our little knowledge base here can be used as a basis for any other ecological datasets even if they quite different. With time, it's possible to grow an increasingly connected knowledge base, linking various ideas from different fields together.

5 Bayesian Higher-Order Probabilistic Programming

Artificial Intelligence has a long history with first-order logic [66] but type theory (or higher-order logic), a more expressive logic, is currently more popular both as a tool to formalize mathematics and as foundation for programming languages. We explored hybrid approaches based on first-order logic and, for this section, we'll briefly discuss Bayesian Higher-Order Probabilistic Programming (BHOPP) along with its relationship with type theory. Probabilistic programming languages are programming languages built to describe probabilistic models and simplify the inference process. Stan [15] and BUGS [47] are two popular examples of probabilistic programming languages used for Bayesian inference, but even more flexible languages for Bayesian probabilistic programming have recently emerged. These languages, like Church [29] and Anglican [77], accept higher-order constructs (that is: functions accepting other functions as arguments). The ambition is that "ultimately we would like simply to be able to do probabilistic programming using any existing programming language as the modeling language" [73].

First-order logic allowed us to model intricate theories but, in practice, almost all modern systems used to formalize mathematics are based on type theory (higher-order logic) [56]. The "first" in first-order logic refers to the limitation that quantification can only be done on individual elements of a set, but not on higher-order structures like sets, predicates, or functions. As a consequence, several important concepts in mathematics cannot be formalized directly with first-order logic. Since type theory supports higher-order quantification, it is used as a foundation to reason about mathematics. Coq, HOL, HOL Light, and Microsoft's LEAN are all popular languages for automated theorem proving based on different forms of type theory [48, 32, 18]. Programming languages in general, not just those targeted at mathematicians, tend to also rely on type theory as foundation [58]. See [24, 56] for an introduction to type theory. Here is where it gets confusing: the *higher* in higher-order logic has a different meaning than in *higher-order probabilistic programming* and yet, Bayesian higher-order probabilistic programming languages (BHOPPL) may hold the key to sound inference mixed with type theory. In BHOPPL, *higher-order* means functions can take functions as arguments, a common capability of modern programming languages. This is necessary for higher-order logic but not sufficient. Where it gets exciting is that a lot

of progress is being made in framing BHOPPL in the language of type theory [12]. In effect, it would bring Bayesian and higher-order logic reasoning together.

Furthermore, software-wise, BHOPPLs are well ahead of the approaches described in previous sections such as hybrid Markov logic networks. Current higher-order probabilistic programming languages operate on variants of well-known languages: Anglican is based on Clojure [77], Pyro is based on Python [9], Turing.jl uses Julia [26]. Many BHOPPLs have been designed to exploit the high-performance architecture developed for deep learning such as distributed systems of GPUs (graphics cards). GPUs have been important in the development of fast learning and inference in deep learning [28]. Pyro [9] is a BHOPPL built on top of PyTorch, one of the most popular frameworks for deep learning, allowing computation to be distributed on systems of GPUs. In contrast, there are no open-source implementations of Markov logic networks running on GPUs. The main downside of BHOPPLs is that, while in theory they may support the richer logics used to formalize modern mathematics, in practice higher-order probability theory is itself not well understood. This is an active research topic [73] but formalization faces serious issues. For one, there are incompatibilities with the standard measure-theoretic foundation of probability theory, which may require rethinking how probability theory is formulated [11, 70, 69, 34, 68]. First-order logic is among the most studied formal languages, making it easy to use a first-order knowledge base with various software. The current informal nature of BHOPPLs make them hard to recommend for the synthesis of knowledge in ecology and evolution, even though they may very well hold the the most potential.

6 Where's our unreasonably effective paradigm?

Legitimate abstractions can often obfuscate how much various subfields are related. Natural selection is a good example. Many formulas in population genetics rely on fitness. Nobody disputes the usefulness of this abstraction, it allows us to think about changes in populations without worrying whether selection is caused by predation or climate change. On the other hand, fitness has also allowed the development of theoretical population genetics to evolve almost independently of ecology. There is a realization that much of the complexity of evolution is related to how selection varies in time and space, which puts evolution in ecology's backyard [8]. Achieving Lewis' goal of formalization would not prevent the use of fitness, but having formulas with fitness cohabiting with formulas explaining the components of fitness would implicitly link ecology and evolution. This goes in both directions: what are the consequences of new discoveries on speciation and adaptive radiations on the formation of metacommunities? How can community dynamics explain the extinction and persistence of new species? If there isn't a single theory of biodiversity, the imperative is to understand biodiversity as a system of theories. Given the scope of ecology and evolution and the vast number of theories involved, it seems difficult to achieve a holistic understanding without some sort of formal system to see how the pieces of the puzzle fit together. Connolly et al. noted how theories for metacommunities were divided between those derived from first principles and those based on statistical methods [16]. In systems unifying rich logics with a probabilistic representation, this distinction does not exist, theories are fully realized as symbolic and statistical entities. Efforts to bring theories in ecology and evolution into a formal setting should be primarily seen as an attempt to put them in context, to force us to be explicit about our assumptions and see how our ideas interact [71].

Despite recent progress at the frontier of logic and probability, there are still practical and theoretical issues to overcome to make a large database of knowledge for ecology and evolution possible. Inference can be difficult in rich knowledge representations, not all methods have robust open-source implementations, and some approaches such as Bayesian higher-order probabilistic programming are themselves not well understood. Plus, while mathematicians benefit from decades of experience making large databases of theorems, there have been no such efforts for ecology and evolution. Lewis' case for the formalization is worth repeating: "when theories are partially formalized [...] the intra- and interworkings of theories become more clearly visible, and the total structure of the discipline becomes more evident" [45]. This vision might soon become reality thanks to increased access to data in evolution and evolution and recent advances at the frontier of logic and probability. Given the pressing need to understand a declining

biodiversity, ecologists and evolutionary biologists should be at the forefront of the efforts to organize theories in unified knowledge bases.

7 Acknowledgements

We thank three anonymous reviewers for their helpful comments. PDP has been funded by an Alexander Graham Bell Graduate Scholarship from the National Sciences and Engineering Research Council of Canada, an Azure for Research award from Microsoft, and benefited from the Hardware Donation Program from NVIDIA. DG is funded by the Canada Research Chair program and NSERC Discovery grant. TP is funded by an NSERC Discovery grant and an FQRNT Nouveau Chercheur grant.

References

[1] R Adams and B Jacobs. A type theory for probabilistic and bayesian reasoning. *CoRR*, abs/1511.09230, 2015.

[2] BK Ahlborn. *Zoological Physics: Quantitative Models of Body Design, Actions, and Physical Limitations of Animals*. Springer, 2004.

[3] SH Bach, M Broecheler, B Huang, and L Getoor. Hinge-loss markov random fields and probabilistic soft logic. arXiv: 1505.04406, 2015.

[4] SH Bach, M Broecheler, B Huang, and L Getoor. Hinge-loss markov random fields and probabilistic soft logic. *Journal of Machine Learning Research*, 18:1–67, 2017.

[5] B Baiser, D Gravel, AR Cirtwill, JA Dunne, AK Fahimipour, LJ Gilarranz, JA Grochow, D Li, ND Martinez, A McGrew, T Poisot, TN Romanuk, DB Stouffer, LB Trotta, FS Valdovinos, RJ Williams, SA Wood, and JD Yeakel. Ecogeographical rules and the macroecology of food webs. *Global Ecology and Biogeography*, 0(0).

[6] D Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

[7] L Behounek, P Cintula, and P Hájek. Introduction to mathematical fuzzy logic. In P Cintula, P Hájek, and C Noguera, editors, *Handbook of Mathematical Fuzzy Logic volume 1*, chapter 1, pages 1–101. College Publications, London, 2011.

[8] G Bell. Fluctuating selection: the perpetual renewal of adaptation in variable environments. *Phil. Trans. R. Soc. B*, 365:87–97, 2010.

[9] E Bingham, JP Chen, M Jankowiak, F Obermeyer, N Pradhan, T Karaletsos, R Singh, P Szerlip, P Horsfall, and ND Goodman. Pyro: Deep universal probabilistic programming. *Journal of Machine Learning Research*, 20(1):973–978, 2019.

[10] DA Bohan, G Caron-Lormier, S Muggleton, A Raybould, and A Tamaddoni-Nezhad. Automated discovery of food webs from ecological data using logic-based machine learning. *PLoS ONE*, 6(12):e29028, 2011.

[11] J Borgströ, AD Gordon, M Greenberg, J Margetson, and J Van Gael. Measure transformer semantics for bayesian machine learning. In G Barthe, editor, *Programming Languages and Systems*, pages 77–96. Springer Berlin Heidelberg, 2011.

[12] J Borgström, U Dal Lago, AD Gordon, and M Szymczak. A lambda-calculus foundation for universal probabilistic programming. In *Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming*, ICFP 2016, pages 33–46, 2016.

[13] C Brouard, C Vrain, J Dubois, D Castel, MA D, and F d'Alche Buc. Learning a markov logic network for supervised gene regulatory network inference. *BMC Bioinformatics*, 14(1):273, 2013.

[14] B Buchanan and EH Shortliffe. *Rule-based Expert Systems: The Mycin experiments of the Stanford Heuristic Programming Project*. Addison-Wesley, 1984.

[15] B Carpenter, A Gelman, M Hoffman, D Lee, B Goodrich, M Betancourt, M Brubaker, J Guo, P Li, and A Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software*, 76(1):1–32, 2017.

[16] SR Connolly, SA Keith, RK Colwell, and C Rahbek. Process, mechanism, and modeling in macroecology. 32(11):835–844, 2017.

[17] A Darwiche. *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press, 2009.

[18] L de Moura, S Kong, J Avigad, F van Doorn, and J von Raumer. The lean theorem prover. In *25th International Conference on Automated Deduction (CADE-25), Berlin, Germany*, 2015.

[19] P Desjardins-Proulx and D Gravel. A complex speciation-richness relationship in a simple neutral model. *Ecology and Evolution*, 2(8):1781–1790, 2012.

[20] P Desjardins-Proulx and D Gravel. How likely is speciation in neutral ecology? *The American Naturalist*, 179(1):137–144, 2012.

[21] P Domingos and D Lowd. *Markov Logic: An Interface Layer for Artificial Intelligence*. Morgan & Claypool Publishers, 2009.

[22] S Andreassen et al. Evaluation of the diagnostic performance of the expert EMG assistant MUNIN. *Electroencephalogr Clin Neurophysiol*, 101(2):129–144, 1996.

[23] RS Etienne and B Haegeman. The neutral theory of biodiversity with random fission speciation. *Theoretical Ecology*, 4(1):87–109, 2011.

[24] WM Farmer. The seven virtues of simple type theory. *Journal of Applied Logic*, 6(3):267–286, 2008.

[25] M Garnelo, K Arulkumaran, and M Shanahan. Towards deep symbolic reinforcement learning. arXiv:1609.05518v2, 2016.

[26] H Ge, K Xu, and Z Ghahramani. Turing: A language for flexible probabilistic inference. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84, pages 1682–1690, 2018.

[27] L Getoor, N Friedman, D Koller, A Pfeffer, and B Taskar. Probabilistic relational models. In L Getoor and B Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2007.

[28] I Goodfellow, Y Bengio, and A Courville. *Deep Learning*. MIT Press, 2016.

[29] ND Goodman, VK Mansinghka, D Roy, K Bonawitz, and JB Tenenbaum. Church: a language for generative models, 2008.

[30] JY Halpern. *Reasoning about Uncertainty*. The MIT Press, 2003.

[31] J Harrison. *Handbook of Practical Logic and Automated Reasoning*. Cambridge University Press, 2009.

[32] J Harrison. Hol light: An overview. In S Berghofer, T Nipkow, C Urban, and M Wenzel, editors, *Theorem Proving in Higher Order Logics*, pages 60–66, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[33] D Heckerman and B Nathwani. An evaluation of the diagnostic accuracy of Pathfinder. *Computers and Biomedical Research*, 25(1):56–74, 1992.

[34] C Heunen, O Kammar, S Staton, and H Yang. A convenient category for higher-order probability theory, 2017.

[35] Z Hu, X Ma, Z Liu, E Hovy, and EP Xing. Harnessing deep neural networks with logic rules. arXiv:1603.06318, 2016.

[36] SP Hubbell. *The Unified Neutral Theory of Biodiversity and Biogeography*, volume 32 of *Monographs in Population Biology*. Princeton University Press, 2001.

[37] B Jacobs and F Zanasi. The logical essentials of bayesian reasoning. *CoRR*, abs/1804.01193, 2018.

[38] D Jain. Knowledge Engineering with Markov Logic Networks: A Review. In *DKB 2011: Proceedings of the Third Workshop on Dynamics of Knowledge and Belief*, 2011.

[39] C Kaliszyk and J Urban. Learning-assisted theorem proving with millions of lemmas. *Journal of Symbolic Computation*, 69:109–128, 2015.

[40] A Kimmig, SH Bach, M Broecheler, B Huang, and L Getoor. A short introduction to probabilistic soft logic. In *Proceedings of the NIPS Workshop on Probabilistic Programming*, 2012.

[41] S Kok and P Domingos. Learning markov logic network structure via hypergraph lifting. In *Proceedings of the 26nd international conference on Machine learning*, 2009.

[42] D Koller and N Friedman. *Probabilistic Graphical Models*. The MIT Press, 2009.

[43] JP Kopelke, T Nyman, K Cazelles, D Gravel, S Vissault, and T Roslin. Food-web structure of willow-galling sawflies and their natural enemies across europe. *Ecology*, 98(6):1730, 2017.

[44] B Kosko. Fuzziness vs probability. *Int J General Systems*, 17:211–240, 1990.

[45] RW Lewis. Evolution: A system of theories. *Perspectives in Biology and Medicine*, 23:551–572, 1980.

[46] RK Lindsay, BG Buchanan, EA Feigenbaum, and J Lederberg. Dendral: A case study of the first expert system for scientific hypothesis formation. *Artificial Intelligence*, 61(2):209–261, 1993.

[47] D Lunn, C Jackson, N Best, A Thomas, and DJ Spiegelhalter. *The BUGS Book – A Practical Introduction to Bayesian Analysis*. Chapman and Hall/CRC, 2012.

[48] The Coq development team. *The Coq proof assistant reference manual*. LogiCal Project, 2004. Version 8.0.

[49] BJ McGill, RS Etienne, JS Gray, D Alonso, MJ Anderson, HK Benecha, M Dornelas, BJ Enquist, JL Green, F He, AH Hurlbert, AE Magurran, PA Marquet, BA Maurer, A Ostling, CU Soykan, KI Ugland, and EP White. Species abundance distributions: moving beyond single prediction theories to integration within an ecological framework. *Ecology Letters*, 10(10):995–1015, 2007.

[50] JM Mendel. *Uncertain Rule-Based Fuzzy Systems*. Springer, 2nd edition, 2017.

[51] JM Mendel and RI Bob John. Type-2 fuzzy sets made simple. *IEEE Transactions on Fuzzy Systems*, 10(2):117–127, 2002.

[52] S Muggleton and L de Raedt. Inductive logic programming: Theory and methods. *The Journal of Logic Programming*, 19-20:629–679, 1994.

[53] S Muggleton and C Feng. Efficient induction of logic programs. In *New Generation Computing*. Academic Press, 1990.

[54] A Nath and P Domingos. Learning relational sum-product networks. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2878–2886, 2015.

[55] R Nedbal and L Serafini. Bayesian markov logic networks. In *AI*IA 2018 – Advances in Artificial Intelligence*, pages 348–361, 2018.

[56] R Nederpelt and H Geuvers. *Type Theory and Formal Proof: An Introduction*. Cambridge University Press, 2014.

[57] J Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[58] BC Pierce. *Types and Programming Languages*. MIT Press, 2002.

[59] T Poisot, R Labrie, E Larson, and A Rahlin. Data-based, synthesis-driven: setting the agenda for computational ecology. *bioRxiv*, 150128, 2018.

[60] T Poisot, DB Stouffer, and S Kéfi. Describe, understand and predict: why do we need networks in ecology? *Functional Ecology*, 30:1878–1882, 2016.

[61] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study, 2013.

[62] DC Queller. Fundamental theorems of evolution. *The American Naturalist*, 189:345–353., 2017.

[63] SH Rice and A Papadopoulos. Evolution with stochastic fitness and stochastic migration. *PLOS ONE*, 4(10):1–12, 2009.

[64] M Richardson and P Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006.

[65] S Russell. Unifying logic and probability. *Communications of the ACM*, 58(7):88–97, 2015.

[66] S Russell and P Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3rd edition, 2009.

[67] A Sadeghian, JM Mendel, and H Tahayori. *Advances in Type-2 Fuzzy Sets and Systems*. Springer, 2014.

[68] A Ścibior, O Kammar, M Vákár, S Staton, Y Hongseok, Y Cai, K Ostermann, SK Moss, C Heunen, and Z Ghahramani. Denotational validation of higher-order bayesian inference. *Proceedings of the ACM on Programming Languages*, 2(POPL):60:1–60:29, 2018.

[69] S Staton. Commutative semantics for probabilistic programming. In H Yang, editor, *Programming Languages and Systems*, pages 855–879, Berlin, Heidelberg, 2017. Springer Berlin Heidelberg.

[70] S Staton, H Yang, F Wood, C Heunen, and O Kammar. Semantics for probabilistic programming: higher-order functions, continuous distributions, and soft constraints, 2016.

[71] P Suppes. The desirability of formalization in science. *Journal of Philosophy*, 65(20):651–664, 1968.

[72] A Tamaddoni-Nezhad, G Milani, A Raybould, S Muggleton, and DA Bohan. Construction and validation of food webs using logic-based machine learning and text mining. *Advances in Ecological Research*, 49:225–289, 2013.

[73] J-W van de Meent, B Paige, H Yang, and F Wood. An introduction to probabilistic programming. 2018.

[74] J Wang and P Domingos. Hybrid markov logic networks. In *AAAI'08 Proceedings of the 23rd national conference on Artificial intelligenc*, volume 2, pages 1106–1111, 2008.

[75] RJ Williams, A Anandanadesan, and ND Martinez. The probabilistic niche model reveals the niche structure and role of body size in a complex food web. *PLOS One*, 5(8):e12092, 2010.

[76] RJ Williams and ND Martinez. Simple rules yield complex food webs. *Nature*, 404:180–183, 2000.

[77] F Wood, JW van de Meent, and V Mansinghka. A new approach to probabilistic programming inference. In *Proceedings of the 17th International conference on Artificial Intelligence and Statistics*, pages 1024–1032, 2014.

[78] K Yoshikawa, S Riedel, M Asahara, and Y Matsumoto. Jointly identifying temporal relations with markov logic. 2009.

[79] LA Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.

[80] T Zahavy, N Ben-Zrihem, and S Mannor. Graying the black box: Understanding DQNS. arXiv: 1602.02658, 2016.

[81] J Zeng and Z-Q Liu. Type-2 fuzzy markov random fields and their application to handwritten chinese character recognition. *IEEE Transactions on Fuzzy Systems*, 16(3):747–760, 2008.