International Journal of Pure and Applied Mathematics
Volume 114 No. 3 2017, 583-592
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
doi: 10.12732/ijpam.v114i3.13

MODELLING AND VERIFICATION ANALYSIS OF A TWO SPECIES ECOSYSTEM VIA A FIRST ORDER LOGIC APPROACH

Zvi Retchkiman Königsberg
Instituto Politécnico Nacional, CIC
Mineria 17-2, Col. Escandon, Mexico D.F 11800, MEXICO

Abstract: Consider the interaction of populations, in which there are exactly two species, one of which the *predators* eat the *preys* thereby affecting each other. In the study of this interaction Lotka-Volterra models have been used. This paper proposes a formal modelling and verification analysis methodology, which consists in representing the interaction behavior by means of a formula of the first order logic. Then, using the concept of logic implication, and transforming this logical implication relation into a set of clauses, called Skolem standard form, qualitative methods for verification (satisfiability) as well as performance issues, for some queries, are applied.

AMS Subject Classification: 08A99, 93D35, 93D99, 39A11
Key Words: ecosystem, predator-prey system, first order logic, model, verification, unsatisfiability, refutation methods

1. Introduction

Consider the interaction of populations, in which there are exactly two species, one of which the *predators* eat the *preys* thereby affecting each other. Such pairs exist throughout nature: fish and sharks, lions and gazelles, birds and insects, to mention some. In the study of this interaction Lotka-Volterra models have

Received: February 6, 2017
Revised: March 15, 2017
Published: May 23, 2017

been used [1]. This paper proposes a well defined syntax modeling and verification analysis methodology which consists in representing the predator-prey interaction system as a formula of the first order logic, (where the quantifiers are chosen according to a video showing how a group of lions attack a zebra). Then, using the concept of logic implication, and transforming this logical implication relation into a set of clauses, called Skolem standard form, qualitative methods for verification (satisfiability) as well as performance issues, for some queries, are addressed. The method of Putnam-Davis based on Herbrand theorem for testing the unsatisfiability of a set of ground clauses as well as the resolution principle due to Robinson, which can be applied directly to any set of clauses (not necessarily ground clauses), are invoked. The paper is organized as follows. In section 2, a first order background summary is given. In section 3, the Putnam-Davis rules and the resolution principle for unsatisfiability, are recalled. In section 3, the predator-prey problem is addressed. Finally, the paper ends with some conclusions.

2. First Order Logic Background

This section presents a summary of the first order logic theory. The reader interested in more details is encouraged to see [2, 3, 4].

Definition 1. A first-order language $\mathcal{L}$ is an infinite collection of distinct symbols, no one of which is properly contained in another, separated into the following categories: parentheses, connectives, quantifiers, variables, equality symbol, constant symbols,function symbols and predicate symbols.

Definition 2. Terms are defined recursively as follows: (i). A constant is a term,(ii). A variable is a term.(iii). If f is an nth-place function symbol, and $t_1, t_2, \ldots, t_n$ are terms, then $f(t_1, t_2, ..., t_n)$ is a term.(iv). All terms are generated by applying the above rules.

Definition 3. If P is an nth-place predicate symbol, and $t_1, t_2, \ldots, t_n$ are terms,then $p(t_1, t_2, \ldots, t_n)$ is an atom. No other expressions can be atoms.

Definition 4. An occurrence of a variable in a formula is bound if and only if the occurrence is within the scope of a quantifier employing the variable, or is the occurrence in that quantifier. An occurrence of a variable in a formula is free if and only if this occurrence of the variable is not bound.

Definition 5. A variable is free in a formula if at least one occurrence of it is free in the formula. A variable is bound in a formula if at least one occurrence of it is bound.

Definition 6. Well-formed formulas, or formulas for short, in the first-order logic are defined recursively as follows:(i). An atom is a formula, (ii). If F and G are formulas then, $\sim (F)$, $(F \vee G)$, $(F \wedge G)$, and $(F \leftrightarrow G)$ are formulas.(iii). If F is a formula and x is a free variable in F, then $(\forall x)F$ and $(\exists x)F$ are formulas.(iv). Formulas are generated only by a finite number of applications of (i),(ii), and (iii).

Definition 7. An interpretation I of a formula F in the first-order logic consists of a nonempty domain D, and an assignment of "values" to each constant,function symbol, and predicate symbol occurring in F as follows: (1). To each constant, we assign an element in D,(2). To each nth-place function symbol, we assign a mapping from D^n to D,(3). To each nth-place predicate symbol, we assign a mapping from D^n to T, F, where T means true and F means false.

Remark 8. Sometimes to emphasize the domain D, we speak of an interpretation of the formula over D. When we evaluate the truth value of a formula in an interpretation over the domain D, $(\forall x)$ will be interpreted as "for all elements in D," and $(\exists x)$ as "there is an element in D. For every interpretation of a formula over a domain D, the formula can be evaluated to T or F according to the following rules: (1). If the truth values of formulas G and H are evaluated, then the truth values of the formulas $\sim (F)$, $(F \vee G)$, $(F \wedge G)$, $(F \rightarrow G)$, and $(F \leftrightarrow G)$ are evaluated according to the well known formulas of propositional calculus ([2]. $(\forall x)G$ is evaluated to T if the truth value of G is evaluated to T for every $d \in D$; otherwise, it is evaluated to F, (3). $(\exists x)G$ is evaluated to T if the truth value of G is T for at least one $d \in D$; otherwise, it is evaluated to F.We note that any formula containing free variables cannot be evaluated.

Definition 9. A formula G is consistent (satisfiable) if and only if there exists an interpretation I such that G is evaluated to T in I. If a formula G is T in an interpretation I, we say that I is a model of G and I satisfies G.

Definition 10. A formula G is inconsistent (unsatisfiable) if and only if there-exists no interpretation I that satisfies G.

Definition 11. A formula G is valid if and only if every interpretation of G satisfies it.

Definition 12. A formula G is a logical implication of formulas $F_1, F_2, \ldots, F_n$ if and only if for every interpretation I, if $F_1, F_2, \ldots, F_n$ is true in I, G is also true in I.

The following characterization of logical implication plays a very important role as will be shown in the rest of the paper.

Theorem 13. *Given formulas $F_1, F_2, \ldots, F_n$ and a formula G, G is a logical implication of $F_1, F_2, \ldots, F_n$ if and only if the formula $((F_1 \wedge F_2 \wedge \ldots, \wedge F_n) \rightarrow G)$is valid if and only if the formula $(F_1 \wedge F_2 \wedge \ldots \wedge F_n \wedge \sim (G))$ is inconsistent.*

Definition 14. A formula F in the first-order logic is said to be in a prenex normal if and only if is in the form of $(Q_1x_1)(Q_2x_2)\ldots(Q_nx_n)(M)$ where every $Q_ix_i, i = 1, 2, \ldots, n$ is either $\forall x_i$ or $\exists x_i$, and M is a formula containing no quantifiers. $(Q_1x_1)(Q_2x_2)\ldots(Q_nx_n)$ is called the prefix and M is called the matrix of the formula F.

Next, Given a formula F, the following procedure transforms F into a prenex normal form. (1) Eliminate $\rightarrow$ and $\leftrightarrow$, (2) Move $\sim$, (3) Rename variables and (4) Pull quantifiers.(details are provided in [3].)

Let a formula F be already in a prenex normal form i.e., $(Q_1x_1)(Q_2x_2)\ldots(Q_nx_n)(M)$, where M is in a conjunctive normal form CNF (a finite conjunction of clauses, see next definition) . Suppose Q_i is an existential quantifier in the prefix. If no universal quantifier appears before Q_i, we choose a new constant c different from other constants occurring in M, replace all x_i appearing in M by c and delete Q_ix_i from the prefix. If $(Q_1x_1)(Q_2x_2)\ldots(Q_kx_k)$ $(1 \leq k < i)$ are all the universal quantifiers appearing before Q_ix_i, we choose a new k-place function symbol f different from other function symbols in M, replace all x_i in M by $f(x_1, x_2, \ldots, x_k)$ delete Q_ix_i from the prefix. After the above process is applied to all the existential quantifiers in the prefix, the last formula we obtain is called a universal form, or Skolem standard form, of the formula F. The constants and functions used to replace the existential variables are called Skolem functions.

Remark 15. It is important to point out that universal forms are not unique.

Definition 16. A clause is a finite disjunction of zero or more literals (atoms or negation of atoms).

When it is convenient, we shall regard a set of literals as synonymous with a clause. A clause consisting of r literals is called an r-literal clause. A one-literal clause is called a unit clause. When a clause contains no literal, we call it the empty clause, denoted by $\square$. Since the empty clause has no literal that can be satisfied by an interpretation, the empty clause is always false. The importance of transforming a formula F in to its universal form results evident, thanks to the next result.

Theorem 17. *Let S be a set of clauses that represents a universal form of a formula F. Then F is inconsistent if and only if S is inconsistent.*

By definition, a set S of clauses is unsatisfiable if and only if it is false under all interpretations over all domains. Since it is inconvenient and impossible to consider all interpretations over all domains, it would be nice if we could fix on one special domain H such that S is unsatisfiable if and only if S is false under all the interpretations over this domain. Fortunately, there does exist such a domain, which is called the Herbrand universe of S, defined as follows.

Definition 18. Let H_0 be the set of constants appearing in S. If no constant appears then, H_0 is to consist of a single constant, say $H_0 = a$. For $i = 0, 1, 2, \ldots$ let H_{i+1} be the union of H_i, and the set of all terms of the form $f(t_1, t_2, \ldots, t_n)$ for all n-place functions f occurring in S, where $t_j = 1, 2, \ldots n$ are members of the set H_i. Then each H_i is called the i-level constant set of S, and H_∞, is called the Herbrand universe of S.

Definition 19. Let S be a set of clauses. The set of ground atoms of the form $P(t_1, t_2, \ldots, t_n)$ for all n-place predicates P occurring in S, where $t_1, t_2, \ldots, t_n$ are elements of the Herbrand universe of S, is called the atom set, or the Herbrand base of S. A ground instance of a clause C of a set S of clauses is a clause obtained by replacing variables in C by members of the Herbrand universe of S.

We have seen that the problem of logical implication is reducible to the problem of satisfiability, which in turn is reducible to the problem of satisfiability of universal sentences. Next, Herbrand's theorem is presented, which states that to test whether a set S of clauses is unsatisfiable, we need consider only interpretations over the Herbrand universe of S. This can be used together with algorithms for unsatisfiability (Davis Putnam rules discussed in section 3) to develop procedures for this purpose.

Theorem 20. *Let a formula F be already in a prenex normal form i.e., $(Q_1x_1)(Q_2x_2)\ldots(Q_nx_n)(M)$, where M is in a conjunctive normal form CNF and contains no quantifiers, i.e., is universal. Let H_∞ be the Herbrand universe of S (with S the set of clauses that represents the universal form of F). Then F is unsatisfiable if and only there is a finite unsatisfiable set $\acute{S}$ of ground instances of clauses of S.*

Remark 21. Herbrand's theorem suggests a refutation procedure: that is, given an unsatisfiable set S of clauses to prove, if there is a mechanical procedure that can successively generate sub-sets $S_1, S_2 \ldots$ of ground instances of clauses in S and can successively test $S_1, S_2 \ldots$ for unsatisfiability, then, as

guaranteed by Herbrand's theorem, this procedure can detect a finite n such that S_n is unsatisfiable, otherwise it will continue forever i.e., it is undecidable.

3. Unsatisfiability Methods

3.1. Davis and Putnam rules

Davis and Putnam introduced a method for testing the unsatisfiability of a set of ground clauses, therefore it is immediately applicable to a set of clauses S considering interpretations over the Herbrand universe. Their method consists of the following rules: (1) Delete all the ground clauses from S that are tautologies. The remaining set $\acute{S}$ is unsatisfiable if and only if S is, (2) If there is a unit ground clause L in S, obtain $\acute{S}$ from S by deleting those ground clauses in S containing L. If $\acute{S}$ is empty then, S is satisfiable, otherwise obtain a set $\acute{\acute{S}}$ by deleting $\sim (L)$ from $\acute{S}$. $\acute{\acute{S}}$ is unsatisfiable if and only if S is, (3) A literal L in a ground clause of S is said to be pure in S if and only if the literal $\sim (L)$ does not appear in any ground clause in S. If a literal L is pure in S, delete all the ground clauses containing L. The remaining set $\acute{S}$ is unsatisfiable if and only if S is, (4) If the set S can be written as: $(A_1 \vee L) \wedge (A_2 \vee L) \ldots (A_m \vee L) \wedge (B_1 \vee \sim L) \wedge (B_2 \vee \sim L) \ldots (B_m \vee \sim L) \wedge R$ where A_i, B_i and R are free of L and $\sim L$ then, obtain the sets $S_1 = A_1 \wedge A_2 \ldots A_m \wedge R$ and $S_2 = B_1 \wedge B_2 \ldots B_m \wedge R$. S is unsatisfiable if and only if both, $S_1 \cup S_2$ are.

3.2. The Resolution Principle

The procedure introduced by Davis and Putnam relies on Herbrand's theorem which has one major drawback: It requires the generation of sets $S_1, S_2 \ldots$ of ground instances of clauses. For most cases, this sequence grows exponentially. We shall next introduce the resolution principle due to Robinson, a more efficient method than Davis and Putnam procedure. It can be applied directly to any set S of clauses (not necessarily ground clauses) to test the unsatisfiability of S. Resolution is a sound and complete algorithm i.e., a formula in clausal form is unsatisfiable if and only if the algorithm reports that it is unsatisfiable. Therefore it provides a consistent methodology free of contradictions. However, it is not a decision procedure because the algorithm may not terminate.

Definition 22. A substitution is a finite set of the form $\{t_1/v_1, t_2/v_2, \ldots, t_n/v_n\}$, where every v_i is a variable, every t_i, is a term different from v_i. When

the t_i are ground terms, the substitution is called a ground substitution. The substitution that consists of no elements is called the empty substitution and is denoted by ϵ.

Definition 23. Let

$$\theta = \{t_1/x_1, t_2/x_2, \ldots, t_n/x_n\}$$

and

$$\lambda = \{u_1/y_1, u_2/y_2, \ldots, u_m/y_m\}$$

be two substitutions. Then the composition of θ and λ is the substitution, denoted by $\theta \circ \lambda$, that is obtained from the set

$$\{t_1\lambda/x_1, t_2\lambda/x_2, \ldots, t_n\lambda/x_n, u_1/y_1, u_2/y_2, \ldots, u_m/y_m\}$$

by deleting any element $t_j\lambda/x_j$ for which $t_j\lambda = x_j$, and any element u_i/y_i such that y_i is among $x_1, x_2, \ldots, x_n$.

Definition 24. A substitution θ is called a unifier for a set $E_1, E_2, \ldots, E_n$ if and only if $E_1\theta = E_2\theta = \ldots, E_n\theta$ The set $\{E_1, E_2, \ldots, E_n\}$ is said to be unifiable if there is a unifier for it.

Definition 25. A unifier σ for a set $E_1, E_2, \ldots, E_n$ of expressions is a most general unifier if and only if for each unifier θ for the set there is a substitution λ such that $\theta = \sigma \circ \lambda$.

Definition 26. If two or more literals (with the same sign) of a clause C have a most general unifier σ, then $C\sigma$ is called a factor of the clause C. If $C\sigma$ is a unit clause, it is called a unit factor of C.

Definition 27. Let C_1 and C_2 be two clauses (called parent clauses) with no variables in common. Let L_1 and L_2 be two literals in C_1 and C_2, respectively. If L_1 and $\sim (L_2)$ have a most general unifier σ, then the clause $(C_1\sigma - L_1\sigma) \cup (C_2\sigma - L_2\sigma)$ is called a binary resolvent of C_1 and C_2. The literals L_1 and L_2 are called the literals resolved upon.

Definition 28. A resolvent of (parent) clauses C_1 and C_2 is one of the following binary resolvents: (1) a binary resolvent of C_1 and C_2, (2) a binary resolvent of C_1 and a factor of C_2, (3) a binary resolvent of a factor of C_1 and C_2, (4) a binary resolvent of a factor of C_1 and a factor of C_2.

Definition 29. Given a set S of clauses, a deduction of C from S is a finite sequence of clauses $C_1, C_2, \ldots, C_n$ such that each C_i, either is a clause in S or a resolvent of clauses preceding C_i,, and $C_k = C$. A deduction of $\square$ from S is called a refutation, or a proof of S.

The following result, called lifting lemma, plays a key role in the proof of the soundness and completeness theorem for the resolution procedure.

Lemma 30. *If $\acute{C}_1$ and $\acute{C}_2$ are instances of C_1 and C_2, respectively, and if $\acute{C}$ is a resolvent of $\acute{C}_1$ and $\acute{C}_2$ then there is a resolvent C of C_1 and C_2 such that $\acute{C}$ is an instance of C.*

The main result of this subsection, the soundness and completeness theorem for the resolution procedure, is next presented.

Theorem 31. *A set S of clauses is unsatisfiable if and only if there is a deduction of the empty clause $\square$ from S.*

Theorem 32. *The set of unsatisfiable sentences is undecidable.*

4. Predator-Prey System

Consider the interaction of populations, in which there are exactly two species, one of which the *predators* eats the other the *preys* thereby affecting each other's growth rates. Such pairs exist throughout nature: fish and sharks, lions and gazelles, birds and insects, to mention some. It is assumed that, the predator species is totally dependent on a single prey species as its only food supply, the prey has unlimited food supply, and that there is no threat to the pray other than the specific predator. The predator-prey system behavior is described as follows: (1) States: S: preys are safe, D: the preys are in danger, B: the preys are being eaten, I: the predators are idle, L: the predators are in search for a prey, CL: the predators continue searching for a prey, A: the predators attack a prey, F: the predator has finished eating the prey, P: the predator dies; (2) Rules of Inference: (a) if S and L then CL, (b) if S and CL then P, (c) if D and (L or CL) then A, (d) if A then B, (e) if B then F (f) if F then I, (g) if I then L. Therefore, by associating variables to the states, we can define the following predicates: $S(x)$: x is a safe prey, $D(x)$: the prey x is in danger, $B(x, y)$: the prey x is being eaten by predator y, $I(x)$: the predator x is idle, $L(x, y)$: the predator y is in search for a prey x, $CL(x, y)$: the predator y continues searching for a prey x, $A(x, y)$: the predator y attacks prey x, $F(x, y)$: the predator y has finished eating prey x, $P(x)$: the predator x passed away.

Remark 33. The main idea consists of: the predator-prey behavior is expressed by a formula of the first order logic, (where the quantifiers are chosen according to a video showing how a group of lions attack a zebra), some query is expressed as an additional formula. The query is assumed to be a logical

implication of the predator-prey formula (see theorem 13). Then, transforming this logical implication relation into a set of clauses by using the techniques given in section 2, its validity can be checked. Even more using the resolution principle, unifications done during the procedure provide answers to some specific queries. The domain D of the interpretation will be considered to be formed by a set of predators and a set of preys .

The formula that models the predator-prey behavior turns out to be:

$$\begin{aligned}[(\forall x)(\forall y)(S(x) \wedge L(x.y) \rightarrow CL(x,y))] \wedge [(\forall x)(\forall y)(S(x) \wedge CL(x.y) \rightarrow P(y))] \\ \wedge [(\exists x)(\forall y)(D(x) \wedge (L(x,y) \vee CL(x,y))) \rightarrow A(x,y))] \wedge [(\exists x)(\forall y)(A(x,y) \\ \rightarrow B(x,y))] \wedge [(\exists x)(\forall y)(B(x,y) \\ \rightarrow F(x,y))] \wedge [(\exists x)(\forall y)(F(x,y) \rightarrow I(y))] \wedge [(\exists x)(\forall y)(I(y) \rightarrow L(x,y))]. \quad (1)\end{aligned}$$

We are interested in verifying, the following statements:
(S1) Claim: If D and (L or CL) then B. Specifically, we want to know if there is prey p such that the following formula is a logical implication of equation 1: $(\exists p)(\forall q)(D(p) \wedge (L(p,q) \vee CL(p,q))) \rightarrow B(p,q))$. The set of clauses for this case is given by:

$$\begin{aligned}S = \{(\sim S(x) \vee \sim L(x.y) \vee CL(x.y)), (\sim S(x) \vee \sim CL(x,y) \vee P(y)), \\ (\sim D(c_1) \vee \sim L(c_1,z) \vee A(c_1,z), (\sim D(c_2) \vee \sim CL(c_2,w) \\ \vee A(c_2,w)), (\sim A(c_3,u) \vee B(c_3,u)), (\sim B(c_4,v) \vee F(c_4,v)), \\ (\sim F(c_5,r) \vee I(r)), (\sim I(s) \vee L(c_6,s)), (D(p)), (L(p,f(p)) \\ \vee CL(p,f(p)), (\sim B(p,f(p)))\}.\end{aligned}$$

Then a resolution refutation proof, with its required substitutions, is as follows:

(a) $p = c_1, (\sim L(c_1,z) \vee A(c_1,z)) \rightarrow z = u, c_3 = c_1, (\sim L(c_1,z) \vee B(c_1,z)) \rightarrow p = c_1, z = f(p), (\sim L(c_1,f(p))) \rightarrow p = c_1, (CL(c_1,f(p)))$

(b) $p = c_2, (\sim CL(c_2,w) \vee A(c_2,w)) \rightarrow w = u, c_2 = c_3, (\sim CL(c_2,w)) \vee B(c_2,w)) \rightarrow p = c_2, z = f(p), (\sim CL(c_2,f(p))))$.

Now, from the last two equations of (a) and (b), setting $c_2 = c_1$, we get a proof of S i.e., $\square$. Therefore we can conclude that: we not only have proved that the claim is true, but we have computed a value for p, $p = c_1 = c_2 = c_3$. which tell us that the same prey that has been attacked, it has to be the same that is being eaten, and not another one, otherwise, the refutation procedure fails. This result is consistent with reality.

(S2) Claim: if D and (L or CL) then I. Specifically, we want to know if there is prey p such that the following formula is a logical implication of

equation 1: $(\exists p)(\forall q)(D(p) \wedge (L(p,q) \vee CL(p,q))) \rightarrow I(q))$. The set of clauses for this case is given by:

$$S = \{(\sim S(x) \vee \sim L(x.y) \vee CL(x.y)), (\sim S(x) \vee \sim CL(x,y) \vee P(y)), (\sim D(c_1) \vee \sim L(c_1,z) \vee A(c_1,z), (\sim D(c_2) \vee \sim CL(c_2,w) \vee A(c_2,w)), (\sim A(c_3,u) \vee B(c_3,u)), (\sim B(c_4,v) \vee F(c_4,v)), (\sim F(c_5,r) \vee I(r)), (\sim I(s) \vee L(c_6,s)), (D(p)), (L(p,f(p)) \vee CL(p,f(p)), (\sim I(q))\}.$$

Then a resolution refutation proof, with its required substitutions, is as follows:

(a) $r = q, (\sim F(c_5,q)) \rightarrow c_5 = c_4, q = v, (\sim B(c_4,v))$ from (a) of (S1) we know $(\sim L(c_1,z) \vee B(c_1,z))$ therefore $c_1 = c_4, v = z, (\sim L(c_1,z)) \rightarrow p = c_1, z = f(p), (CL(c_1,f(p)))$

(b) $r = q, (\sim F(c_5,q)) \rightarrow c_5 = c_4, q = v, (\sim B(c_4,v))$ from (b) of (S2) we know $(\sim CL(c_2,w)) \vee B(c_2,w))$ therefore $c_2 = c_4, v = w, (\sim CL(c_2,w))$.

Now, from the last two equations of (a) and (b), setting $c_2 = c_1, w = f(p)$, we get a proof of S i.e., $\square$.

Therefore, the claim holds for $p = c_1 = c_2 = c_3 = c_4 = c_5$, and the same conclusion given in (S1) extrapolates for this case.

5. Conclusions

The main contribution of the paper consists in the study of the predator-pray system by means of a formal reasoning deductive methodology based on first order logic theory. The results obtained are consistent with how the predator-prey system behaves in real life.

References

[1] R. Haberman, *Mathematical Models in mechanical vibrations, population dynamics, and traffic flow*, Prentice Hall (1977).

[2] Chin-Liang Chang and Richard Char-Tung Lee,*Symbolic Logic and Mechanical Theorem Proving*. Academic Press (1973).

[3] M. Davis, R. Sigal and E. Weyuker, *Computability, Complexity, and Languages, Fundamentals of Theoretical Computer Science*, Academic Press (1983).

[4] M. Ben Ari, *Mathematical Logic for Computer Science*, Spinger Verlag (2012).