Communications in Applied Analysis, **21**, No. 3 (2017), 337-351 ISSN: 1083-2564

MODELLING AND VERIFICATION ANALYSIS OF THE BIOLOGICAL COOPERATIVE COMPETITION PROBLEM VIA A FIRST ORDER LOGIC APPROACH

ZVI RETCHKIMAN KÖNIGSBERG

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

ABSTRACT: This paper addresses the biological cooperative competition problem among organisms of the same or different species associated with the need for a common resource that occurs in a limited supply relative to demand. If two competitors try to occupy the same realized niche, one species will try to eliminate the other. Therefore, there is a need to cooperate sharing part of the resource so that both organisms will benefit from it. In this work, the biological cooperative competition problem is modelled as 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 as well as performance issues, for some queries, are applied.

AMS Subject Classification: 08A99, 93D35, 93D99, 39A11

Key Words: biological cooperative competition system, first order logic, model, verification, unsatisfiability, refutation methods

Received: January 13, 2017 ; **Accepted:** April 8, 2017 ;
Published: April 11, 2017. **doi:** 10.12732/caa.v21i3.2
Dynamic Publishers, Inc., Acad. Publishers, Ltd. http://www.acadsol.eu/caa

1. INTRODUCTION

Consider the biological competition problem among organisms of the same or different species associated with the need for a common resource that occurs in a limited supply relative to demand. In other words, competition better defined as interaction occurs when the capability of the environment to supply resources is smaller than the potential biological requirement so that organisms interfere with each other. More concisely we can say that competition is the interaction between individuals of the same or different species leading to an increase of the fitness of one at the expense of the other. Plants, for example, often compete for access to a limited supply of nutrients, water, sunlight, and space. Therefore, two species cannot indefinitely coexist if they are limited by the same resource. If two competitors try to occupy the same realized niche, one species will try to eliminate the other [1]. Therefore, there is a need to cooperate sharing part of the resource so that both organisms will benefit from it. In the study of this type of problems Lotka-Volterra models as well as evolutionary game theory concepts have been used [2, 3]. This paper proposes a well defined syntax modeling and verification analysis methodology which consists in representing the biological cooperative competition system as 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 (validity) 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 biological cooperative competition 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 [4, 5, 6].

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 ([4]. $(\forall x)G$ 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 [5]).

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 $\Box$ 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 $\Box$ from S.*

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

4. THE BIOLOGICAL COOPERATIVE COMPETITION SYSTEM

Consider the biological cooperative competition problem among organisms of the same or different species associated with the need for a common resource that occurs in a limited supply relative to demand. In other words, competition better defined as interaction occurs when the capability of the environment to supply resources is smaller than the potential biological requirement so that organisms interfere with each other. Plants, for example, often compete for access to a limited supply of nutrients, water, sunlight, and space. Therefore, two species cannot indefinitely coexist if they are limited by the same resource. If two competitors try to occupy the same realized niche, one species will try to eliminate the other [1]. Therefore, there is a need to cooperate sharing part of the resource so that both organisms will benefit from it. The biological cooperative competition system behavior is described as follows: (1) States: S: resources are safe, D: the resources are in danger, B: the resources are being eaten, I_1, I_2: the organisms are inactive, L_1, L_2: the organisms are in search for a resource, CL_1, CL_2: the organisms continue searching for a resource, A_1, A_2: the organisms attack the resource, F_1, F_2: the organisms have finished eating the resource, P_1, P_2: the organisms die; (2) Rules of Inference: (a) if S and L_1 and L_2 then CL_1 and CL_2, (b) if S and CL_1 and CL_2 then P_1 and P_2, (c) if D and ((L_1 or CL_1) and not(L_2 or CL_2)) then A_1 and not(A_2), (d) if D and (not(L_1 or CL_1) and (L_2 or CL_2)) then not(A_1) and A_2, (e) if A_1 and not(A_2) then B_1 and not(B_2),(f) if not(A_1) and A_2 then not(B_1) and B_2, (g) if B_1 and not(B_2) then F_1 and not(F_2), (h) if not(B_1) and B_2 then not(F_1) and F_2, (i) if F_1 and not(F_2) then I_1 and not(I_2), (j) if not(F_1) and F_2 then not(I_1) and I_2,(k) if I_1 and not(I_2) then L_1 and not(L_2), (l) if not(I_1) and I_2 then not(L_1) and L_2.

Remark 33. It important to underline that the inference rules express the cooperative property of the biological competition system over the resource, where one organism takes control over part of the resource while the other one takes control over part of the rest. As a result there is no possible contradiction when two complementary rules execute at the same time. This cooperative competitive behavior differs from the strictly competitive where there exists just one of the organisms (the winner) who takes completely control of the resource.

Therefore, by associating variables to the states, we can define the following predicates $i = 1, 2$: $S(x)$: x is a safe resource, $D(x)$: the resource x is in danger, $B_i(x, y)$: the resource x is being eaten by the organisms y, $I_i(x)$: the organisms x are inactive, $L_i(x, y)$: the organisms y is in search for a resource x, $CL_i(x, y)$: the organisms y continue searching for a resource x, $A_i(x, y)$: the organisms y attack the resource x, $F_i(x, y)$: the organisms y have finished eating the resource x, $P_i(x)$: the organisms x passed away.

Remark 34. The main idea consists of: the biological cooperative competition system behavior is expressed by a formula of the first order logic. Then, after doing skolemitization i.e., obtaining a Skolem standard form, some query is expressed as an additional formula. The query is assumed to be a logical implication of the biological cooperative competition 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 the two organisms and the resources.

The formula that models the biological cooperative competition system behavior turns out to be:

$$[(\forall x)(\forall y)(S(x) \wedge L_1(x.y) \wedge L_2(x.y) \rightarrow CL_1(x, y) \wedge CL_2(x.y))], \tag{1}$$

$$\begin{gathered}\wedge [(\forall x)(\forall y)(S(x) \wedge CL_1(x.y) \wedge CL_2(x.y) \rightarrow P_1(y) \wedge P_2(y))] \\ \wedge [(\exists x)(\forall y)(D(x) \wedge (L_1(x, y) \vee CL_1(x, y)) \wedge \sim (L_2(x, y) \vee CL_2(x, y)) \\ \rightarrow A_1(x, y) \wedge \sim A_2(x, y))] \wedge [(\exists x)(\forall y)(D(x) \wedge \sim (L_1(x, y) \vee CL_1(x, y)) \\ \wedge (L_2(x, y) \vee CL_2(x, y)) \rightarrow \sim A_1(x, y) \wedge A_2(x, y))] \\ \wedge [(\exists x)(\forall y)(A_1(x, y) \wedge \sim A_2(x, y) \rightarrow \\ B_1(x, y)) \wedge \sim B_2(x, y))] \wedge [(\exists x)(\forall y)(\sim A_1(x, y) \wedge A_2(x, y) \\ \rightarrow \sim B_1(x, y) \wedge B_2(x, y))] \wedge [(\exists x)(\forall y)(B_1(x, y)\end{gathered}$$

$$\wedge \sim B_2(x,y) \to F_1(x,y) \wedge \sim F_2(x,y))] \wedge [(\exists x)(\forall y)(\sim B_1(x,y)$$
$$\wedge B_2(x,y) \to \sim F_1(x,y) \wedge F_2(x,y))] \wedge [(\exists x)(\forall y)(F_1(x,y)$$
$$\wedge \sim F_2(x,y) \to I_1(y) \wedge \sim I_2(y))] \wedge [(\exists x)(\forall y)(\sim F_1(x,y)$$
$$\wedge F_2(x,y) \to \sim I_1(y) \wedge I_2(y))] \wedge [(\exists x)(\forall y)(I_1(y)$$
$$\wedge \sim I_2(y) \to L_1(x,y) \wedge \sim L_2(x,y))] \wedge [(\exists x)(\forall y)(\sim I_1(y)$$
$$\wedge I_2(y) \to \sim L_1(x,y) \wedge L_2(x,y))].$$

We are interested in verifying, the following statements:

(S1) Claim: If D and ((L_1 or CL_1) and not(L_2 or CL_2)) then B_1 and not(B_2). Specifically, we want to know if there is resource m such that the following formula is a logical implication of equation 1: $(\exists m)(\forall q)(D(m) \wedge (L_1(m,q) \vee CL_1(m,q)) \wedge \sim (L_2(m,q) \vee CL_2(m,q)) \to B_1(m,q) \wedge (\sim B_2(m,q)))$. The set of clauses for this case is given by:
$S = \{(\sim S(x) \vee \sim L_1(x.y) \vee \sim L_2(x.y) \vee CL_1(x.y)), (\sim S(x) \vee \sim L_1(x.y) \vee \sim L_2(x.y) \vee CL_2(x.y)), (\sim S(x) \vee \sim CL_1(x,y) \vee \sim CL_2(x,y) \vee P_1(y)), (\sim S(x) \vee \sim CL_1(x,y) \vee \sim CL_2(x,y) \vee P_2(y)), (\sim D(c_1) \vee \sim L_1(c_1,z) \vee L_2(c_1,z) \vee CL_2(c_1,z) \vee A_1(c_1,z)), (\sim D(c_1) \vee \sim L_1(c_1,z) \vee L_2(c_1,z) \vee CL_2(c_1,z) \vee \sim A_2(c_1,z)), (\sim D(c_1) \vee \sim CL_1(c_1,z) \vee L_2(c_1,z) \vee CL_2(c_1,z) \vee A_1(c_1,z)), (\sim D(c_1) \vee \sim CL_1(c_1,z) \vee L_2(c_1,z) \vee CL_2(c_1,z) \vee \sim A_2(c_1,z)), (\sim D(c_2) \vee \sim L_2(c_2,t) \vee L_1(c_2,t) \vee CL_1(c_2,t) \vee \sim A_1(c_2,t)), (\sim D(c_2) \vee L_1(c_2,t) \vee \sim L_2(c_2,t) \vee CL_1(c_2,t) \vee A_2(c_2,t)), (\sim D(c_2) \vee CL_1(c_2,t) \vee L_1(c_2,t) \vee \sim CL_2(c_2,t) \vee \sim A_1(c_2,t)), (\sim D(c_2) \vee CL_1(c_2,t) \vee L_1(c_2,t) \vee \sim CL_2(c_2,t) \vee A_2(c_2,t)), (\sim A_1(c_3,s) \vee A_2(c_3,s) \vee B_1(c_3,s)), (\sim A_1(c_3,s) \vee A_2(c_3,s) \vee \sim B_2(c_3,s)), (\sim A_2(c_4,d) \vee A_1(c_4,d) \vee \sim B_1(c_4,d)), (\sim A_2(c_4,d) \vee A_1(c_4,d) \vee B_2(c_4,d)), (\sim B_1(c_5,j) \vee B_2(c_5,j) \vee F_1(c_5,j)), (\sim B_1(c_5,j) \vee B_2(c_5,j) \vee \sim F_2(c_5,j)), (\sim B_2(c_6,h) \vee B_1(c_6,h) \vee \sim F_1(c_6,h)), (\sim B_2(c_6,h) \vee B_1(c_6,h) \vee F_2(c_6,h)), (\sim F_1(c_7,r) \vee F_2(c_7,r) \vee I_1(c_7)), (\sim F_1(c_7,r) \vee F_2(c_7,r) \vee \sim I_2(c_7)), (\sim F_2(c_8,k) \vee F_1(c_8,k) \vee \sim I_1(c_8)), (\sim F_2(c_8,k) \vee F_1(c_8,k) \vee I_2(c_8)), (\sim I_1(w) \vee I_2(w) \vee L_1(c_9,w)), (\sim I_1(w) \vee I_2(w) \vee \sim L_2(c_9,w)), (\sim I_2(p) \vee I_1(p) \vee L_1(c_{10},p)), (\sim I_2(p) \vee I_1(p) \vee L_2(c_{10},p)), (D(m)), (L_1(m,f(m)) \vee CL_1(m,f(m)), (\sim L_2(m,f(m))), (\sim CL_2(m,f(m))), (\sim B_1(m,f(m)) \vee B_2(m,f(m)))\}$. Where, due to the cooperation behavior the following conditions must be impossed: $c_1 \neq c_2, c_3 \neq c_4, c_5 \neq c_6, c_7 \neq c_8, c_9 \neq c_{10}$.

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

(a) $m = c_3, s = f(c_3), (\sim A_1(c_3, f(c_3)) \vee A_2(c_3, f(c_3)) \vee B_1(c_3, f(c_3)))(\sim B_1(c_3, f(c_3)) \vee B_2(c_3, f(c_3))) \to (\sim A_1(c_3, f(c_3)) \vee A_2(c_3, f(c_3)) \vee B_2(c_3, f(c_3)))$.

(b) $(\sim A_1(c_3, f(c_3)) \vee A_2(c_3, f(c_3)) \vee B_2(c_3, f(c_3)))(\sim A_1(c_3, s) \vee A_2(c_3, s) \vee \sim B_2(c_3, s)) \to (\sim A_1(c_3, f(c_3)) \vee A_2(c_3, f(c_3)))$.

(c) $m = c_1, s = f(c_1), (\sim D(c_1) \vee \sim CL_1(c_1, f(c_1)) \vee L_2(c_1, f(c_1)) \vee CL_2(c_1, f(c_1)) \vee \sim A_2(c_1, f(c_1)))(D(c_1))(\sim L_2(c_1, f(c_1)))(\sim CL_2(c_1, f(c_1))) \to (\sim CL_1(c_1, f(c_1)) \vee \sim A_2(c_1, f(c_1)))$.

(d) $(m = c_1, s = f(c_1), (\sim D(c_1) \vee \sim L_1(c_1, f(c_1)) \vee L_2(c_1, f(c_1)) \vee CL_2(c_1, f(c_1)) \vee \sim A_2(c_1, f(c_1)))(D(m))(\sim L_2(c_1, f(c_1)))(\sim CL_2(c_1, f(c_1))) \to (\sim L_1(c_1, f(c_1)) \vee \sim A_2(c_1, f(c_1)))$.

(e) $(m = c_1)(\sim L_1(c_1, f(c_1)) \vee \sim A_2(c_1, f(c_1)))(L_1(c_1, f(c_1)) \vee CL_1(c_1, f(c_1)) \to CL_1(c_1, f(c_1)) \vee \sim A_2(c_1, f(c_1)))$.

(f) $(\sim CL_1(c_1, f(c_1)) \vee \sim A_2(c_1, f(c_1)))(CL_1(c_1, f(c_1)) \vee \sim A_2(c_1, f(c_1))) \to (\sim A_2(c_1, f(c_1)))$.

(g) $m = c_1, s = f(c_1), (\sim D(c_1) \vee \sim CL_1(c_1, f(c_1)) \vee L_2(c_1, f(c_1)) \vee CL_2(c_1, f(c_1)) \vee A_1(c_1, f(c_1)))(D(m))(\sim L_2(c_1, f(c_1)))(\sim CL_2(c_1, f(c_1))) \to (\sim CL_1(c_1, f(c_1)) \vee A_1(c_1, f(c_1)))$.

(h) $m = c_1, s = f(c_1), (\sim D(c_1) \vee \sim L_1(c_1, f(c_1)) \vee L_2(c_1, f(c_1)) \vee CL_2(c_1, f(c_1)) \vee A_1(c_1, f(c_1)))(D(m))(\sim L_2(c_1, f(c_1)))(\sim CL_2(c_1, f(c_1))) \to (\sim L_1(c_1, f(c_1)) \vee A_1(c_1, f(c_1)))$.

(i) $(\sim L_1(c_1, f(c_1)) \vee A_1(c_1, f(c_1)))(L_1(c_1, f(c_1)) \vee CL_1(c_1, f(c_1)) \to (CL_1(c_1, f(c_1) \vee A_1(c_1, f(c_1)))$.

(j) $(\sim CL_1(c_1, f(c_1)) \vee A_1(c_1, f(c_1)))(CL_1(c_1, f(c_1) \vee A_1(c_1, f(c_1))) \to (A_1(c_1, f(c_1))$.

Now, from (b) and (j) setting $c_1 = c_3$, we get:

(k) $(\sim A_1(c_1, f(c_1)) \vee A_2(c_1, f(c_1)))(A_1(c_1, f(c_1)) \to (A_2(c_1, f(c_1))$.

Therefore, from the conclusion of (f) and (k), we get a proof of S i.e., $\Box$.

Therefore we can conclude that: we not only have proved that the claim is valid, but we have computed a value for m, $m = c_1 = c_3$, which tell us that the same resource that has been attacked, it has to be the same that is being eaten, and not another one, otherwise, the refutation procedure fails.

Remark 35. It is also true that the claim: If D and not(L_1 or CL_1) and (L_2 or CL_2) then not(B_1) and B_2 i.e., that the following formula is a logical implication of equation 1: $(\exists w)(\forall \chi)(D(w) \wedge \sim (L_1(w, \chi) \vee CL_1(w, \chi)) \wedge$

$(L_2(w,\chi) \vee CL_2(w,\chi)) \rightarrow (\sim B_1(w,\chi)) \wedge B_2(w,\chi))$, getting $w = c_2 = c_4$, The proof follows the same steps as the one provided, just changing names.

(S2) Claim: If D and $((L_1$ or $CL_1)$ and not$(L_2$ or $CL_2))$ then B_1 and not(B_2) and If D and not$(L_1$ or $CL_1)$ and $(L_2$ or $CL_2)$ then not(B_1) and B_2. Specifically, we want to show that the cooperative behavior of the organisms over the resource holds. Therefore, we want to prove that there exist m and w $m \neq w$ such that the following formula is a logical implication of equation 1: $(\exists m)(\forall q)(D(m) \wedge (L_1(m,q) \vee CL_1(m,q)) \wedge \sim (L_2(m,q) \vee CL_2(m,q)) \rightarrow B_1(m,q) \wedge (\sim B_2(m,q))) \wedge (\exists w)(\forall \chi)(D(w) \wedge \sim (L_1(w,\chi) \vee CL_1(w,\chi)) \wedge (L_2(w,\chi) \vee CL_2(w,\chi)) \rightarrow (\sim B_1(w,\chi)) \wedge B_2(w,\chi))$. The set of clauses for this case is given by: $S = \{(\sim S(x) \vee \sim L_1(x.y) \vee \sim L_2(x.y) \vee CL_1(x.y)), (\sim S(x) \vee \sim L_1(x.y) \vee \sim L_2(x.y) \vee CL_2(x.y)), (\sim S(x) \vee \sim CL_1(x,y) \vee \sim CL_2(x,y) \vee P_1(y)), (\sim S(x) \vee \sim CL_1(x,y) \vee \sim CL_2(x,y) \vee P_2(y)), (\sim D(c_1) \vee \sim L_1(c_1,z) \vee L_2(c_1,z) \vee CL_2(c_1,z) \vee A_1(c_1,z)), (\sim D(c_1) \vee \sim L_1(c_1,z) \vee L_2(c_1,z) \vee CL_2(c_1,z) \vee \sim A_2(c_1,z)), (\sim D(c_1) \vee \sim CL_1(c_1,z) \vee L_2(c_1,z) \vee CL_2(c_1,z) \vee A_1(c_1,z)), (\sim D(c_1) \vee \sim CL_1(c_1,z) \vee L_2(c_1,z) \vee CL_2(c_1,z) \vee \sim A_2(c_1,z)), (\sim D(c_2) \vee \sim L_2(c_2,t) \vee L_1(c_2,t) \vee CL_1(c_2,t) \vee \sim A_1(c_2,t)), (\sim D(c_2) \vee L_1(c_2,t) \vee \sim L_2(c_2,t) \vee CL_1(c_2,t) \vee A_2(c_2,t)), (\sim D(c_2) \vee CL_1(c_2,t) \vee L_1(c_2,t) \vee \sim CL_2(c_2,t) \vee \sim A_1(c_2,t)), (\sim D(c_2) \vee CL_1(c_2,t) \vee L_1(c_2,t) \vee \sim CL_2(c_2,t) \vee A_2(c_2,t)), (\sim A_1(c_3,s) \vee A_2(c_3,s) \vee B_1(c_3,s)), (\sim A_1(c_3,s) \vee A_2(c_3,s) \vee \sim B_2(c_3,s)), (\sim A_2(c_4,d) \vee A_1(c_4,d) \vee \sim B_1(c_4,d)), (\sim A_2(c_4,d) \vee A_1(c_4,d) \vee B_2(c_4,d)), (\sim B_1(c_5,j) \vee B_2(c_5,j) \vee F_1(c_5,j)), (\sim B_1(c_5,j) \vee B_2(c_5,j) \vee \sim F_2(c_5,j)), (\sim B_2(c_6,h) \vee B_1(c_6,h) \vee \sim F_1(c_6,h)), (\sim B_2(c_6,h) \vee B_1(c_6,h) \vee F_2(c_6,h)), (\sim F_1(c_7,r) \vee F_2(c_7,r) \vee I_1(c_7)), (\sim F_1(c_7,r) \vee F_2(c_7,r) \vee \sim I_2(c_7)), (\sim F_2(c_8,k) \vee F_1(c_8,k) \vee \sim I_1(c_8)), (\sim F_2(c_8,k) \vee F_1(c_8,k) \vee I_2(c_8)), (\sim I_1(w) \vee I_2(w) \vee L_1(c_9,w)), (\sim I_1(w) \vee I_2(w) \vee \sim L_2(c_9,w)), (\sim I_2(p) \vee I_1(p) \vee L_1(c_{10},p)), (\sim I_2(p) \vee I_1(p) \vee L_2(c_{10},p)), (D(m)), (L_1(m,f(m)) \vee CL_1(m,f(m)), (\sim L_2(m,f(m))),$ $(\sim CL_2(m,f(m))), (\sim B_1(m,f(m)) \vee B_2(m,f(m))), (D(w)), (L_2(w,f(w)) \vee CL_2(w,f(w)), (\sim L_1(w,f(w))), (\sim CL_1(w,f(w))), (\sim B_2(w,f(w)) \vee B_1(w,f(w))\}$. Where, due to the cooperation behavior the following conditions must be impossed: $c_1 \neq c_2, c_3 \neq c_4, c_5 \neq c_6, c_7 \neq c_8, c_9 \neq c_{10}$.

Corollary 36. *The proof follows from what was discussed in claim (S1) getting: $m = c_1 = c_3$, and $w = c_2 = c_4$, and since $c_1 \neq c_2$ and $c_3 \neq c_4$ i.e.,*

$m \neq w$ claim (S2) results to be valid.

5. CONCLUSIONS

The main contribution of the paper consists in the study of the biological cooperative competition system by means of a formal reasoning deductive methodology based on first order logic theory. Verification (validity) as well as performance issues, for some queries were addressed.

REFERENCES

[1] G.F. Gause, *The Struggle for Existence*, Baltimore, Williams and Wilkins, 1934.

[2] J. Weibull, *Evolutionary Game Theory*, The MIT Press, 1977.

[3] R. Haberman, *Mathematical Models in Mechanical Vibrations, Population Dynamics, and Traffic Flow*, Prentice Hall, 1977.

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

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

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