Multiagent Chess Games

Cyrus F. Nourani
Project_METAAI@CompuServe.com
USA Academic cyrusfn@cs.ucsb.edu
June 1996, Revised April 1997

ABSTRACT

A Multiagent Chess playing paradigm is defined. By defining spheres and strength degrees for pieces winning strategies on games can be defined. The new Intelligent Tree Computing theories we have defined since 1994 can be applied to present precise strategies and prove theorems on games. The multiagent chess game model is defined by an isomorphism on multiboards and agents. Intelligent game trees are presented and goal directed planning is defined by tree rewrite computation on intelligent game trees. Applying intelligent game trees we define capture agents and state an overview to multiagent chess thinking. Game tree intelligence degree is defined and applied to prove model-theoretic soundness and completeness. The game is viewed as a multiplayer game with only perfect information between agent pairs. The man-machine technologies thought dilemma is dispelled in brief by addressing the thinking in the absolute versus thinking for a precisely defined area with a multiagent image for the computing mind.
Keywords Intelligent Game Trees, Multiagent A.I., Multiagent Games, Multiboard A.I., Multiagent Planning

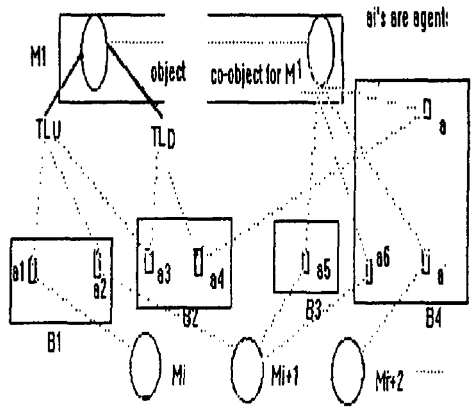
1. Introduction

A Multi Agent Chess playing paradigm is defined. By defining spheres on boards and strength degrees for pieces winning strategies on games can be defined. The new Intelligent Tree Computing theories we have defined can be applied to present precise strategies and prove theorems on games. The present computational model for multi-agent system provides a formal basis for single agent moves. For each agent function there is a way to determine mutual information content with respect to the decision trees connected to it. A single agent makes its decisions for each operation or action by computing a plausible next move set. The plausible next move set might have dynamic properties. It might consist of a set of trees bearing agent functions which compute their next move sets to update the computing trees intelligence content. Foundational questions such as: what does playing Deep Blue(Newborn 1997) mean to AI? and what the future of the chess technology (Berliner 1978) is, are alluded to in the paper. The cultural question: Why the negative emotional reaction to a very notion of artificial intelligence by many philosophers and cognitive scientists; and the ontological and epistemological questions for "sensing" intelligence for chess are addressed. We further question whether chess is a really a two person game with perfect information by offering an alternative view point. A basis for a theory of computing with *intelligent languages* and intelligent algebraic *game tree rewriting* is presented. We present *intelligent syntax* and put forth intelligent tree rewriting. We define a basis for *algebraic tree information-theoretic computing,* presenting the concepts of tree information content and mutual information amongst trees. We have defined the basis for algebraic tree rewrite computing with intelligent trees, and for computing with *decisive agents.* Intelligent tree completion theorems are presented, and techniques for generating initial intelligent models are developed for this computing method. There is soundness and completeness theorems for the theory of computing with intelligent trees and their model theory. Goal directed move generation as in (Newell,Shaw, &Simon 1963) is a planning problem which is definable by intelligent trees and we have begun to write papers for its mathematics and practice in (Nourani 94a,1995f).

2. The Multiagent Chess Model

The term "agent" has been recently applied to refer to AI constructs that enable computation on behalf of an AI activity(Genesereth&Nilsson 1987),(Nourani 1993c,96a) with definitions which are software/hardware independent, thus implementation, independent (Nourani 1993c). The present paper applies the techniques and theory of computing with intelligent trees(Nourani 1993e,96a), with signatures that bear agent functions on trees, to define intelligent game trees for chess. Our results for computability of initial models by abstract subtree replacement systems (Nourani 1984) are presented in brief as foundations for computing on trees to be applicable to intelligent free tree computing. We have presented the concepts of intelligent syntax, intelligent languages , and their applications to AI and programming in (Nourani 1994c,93e,96a). Algebraic tree rewriting is defined on intelligent trees by presenting the concepts and definition of algebraic tree information content and mutual tree information content within a forest. Let us view an abstract chess player as a pair <P,B>. The player P makes its moves based on the board B it views. <P,B> might view chess as if the pieces on the board had come alive and were autonomous agents communicating by messages, as if it were Alice in

Wonderland. For each chess piece a designating agent is defined. To have an agent model we must define the internal structure of agents as well as their external behavior. The <P,B> chess agents are what are called hysteric agents (Genesereth&Nilson 1987). A hysteric agent has an internal state set I, which the agent can distinguish its membership. The agent can trasnit from each internal state to another in a single step. Actions by chess hysteric agents are based on I and board observations. There is an external state set S, modulated to a set T of distinguishable subsets from the observation view point. An agent cannot distinguish states in the same partition defined by a game congruence relation. A sensory function $s : S \rightarrow T$ maps each state to the partition it belongs. Let A be a set of actions which can be performed by agents. A function action can be defined to characterize an agent activity action: $T \rightarrow A$ There is also a memory update fucntion mem: $I \times T \rightarrow I$. Instead of modeling the game as if it were played by something external to the board, we view it all as if <P,B> is an autonomous extension of a human player's mind. For each piece p we define its sphere S(p). S(p) is a defined by the board state from <P,B> and its situation defined by its location on the board the threat set, and the capture set. For each piece we define an activating agent. The muliagent morphisms and their ontologies are further defined by (Nourani 1993c). Pieces with nontrivial capture capability, i.e., Bishop,Knight, Rook, and Queen are designated by AF agents, abbreviating Arbitrary Force. Pawns have agents with limited force sphere, i.e. their immediate front squares. The obvious game objective is to force a win. The chess pieces transmit what is necessary from their sphere to their sphere agent group. The way the sphere group for an agent is defined might be part of a specific play strategy. The agent society for <P,B> determines which piece moves, since there is only a single move allowed at each stage. The move is determined by the king agent signaling it via the *game executive agent*.. The game executive agent is a designated agent which is impartial to the game played. When defined on a computer, the game executive agent is the computing agent issuing the move for either side. <P,B>'s external state is partitioned to many board views by each hysteric agent.The agents cooperate on problem solving based on a multiboard model depicted by the enclosed figure. There are multiboards as viewed by each agent and the object-coobject design depicts the board from the two player game view point. The mutliboard multiagent model is from Double Vision Computing 23. The object-cobject pair is an isomorphism defined corresponding the <P.B>'s external state partition to the board viewed from two players. Hence there is a board-cobroad partitioned via object-coobject pairs from each piece designating agent view and their agent computation. (see design figures in Nourani 1995b).



Mi's are (object,coobject) pairs; ai's are agents, Bi's are where ai's cooperate onboard
Design with Object Co-object Pairs

Fig 1 The Multiagent Multiboard Field

Multibaord-Multiagent computing is new since our projects from 1993. The cooperative problem solving paradigms have been applied ever since the AI methods put forth by (Hays-Roth 1985).

3. Intelligent Game Trees

3.1 Game Trees and And/Or Trees

The chess game trees can be defined by AND/Or trees(Nilsson 1969,71). For the intelligent game trees and the problem solving techniques defined 23, the same model can be applied to the game trees in the sense of two person games and to the state space from the single agent view. The two person game tree is obtained from the intelligent tree model, as is the state space tree for agents. To obtain the two-person game tree the cross-board-cobroad agent computation is depicted on a tree. Whereas the sate-space trees for each agent is determined by the computation sequence on its side of the board-coboard. We have defined an abstract notion of information on intelligent game trees corresponding to what Sahnnon might have defined for games(Shannon 1956). The way the intelligent game trees are defined, a tree information theoretic theorem presents itself, corresponding intelligent tree rewriting to tree information content preservation.

3.2 Game Tree Rewrite Computing

Next, tree rewriting with intelligent trees is formalized by defining canonical intelligent initial models and results that intelligent algebraic tree rewriting leads to intelligent initial models. Models for intelligent theories emerge from the algebraic intelligent tree rewriting. Intelligent algebraic tree completion theorems and initial model rewrite theorems are put forth for intelligent trees. To bring the techniques to a climax a soundness and completeness theorems is proved for intelligent tree rewriting as a formal model-theoretic computing technique. We further state that there are new theoretical developments for computing that are an information-theoretic formulation for computing. For goal directed move planning examples goals might be as follows. The coboard AF agents be tree rewritten to the empty string, i.e. capture. The coboard agents be rewritten to king and pawn agents only. The coabrad castle to be rewritten to a vulnerable state. Mate: the coboard king designated agent acknowledging it.

3.3 Computing On Intelligent Trees

3.3.1 Our Recent Views

Henkin style proof for Godel's completeness theorem, is implemented by constructing a model directly from the syntax of theories. The computing enterprise requires more general techniques of model construction and extension, since it has to accommodate dynamically changing world descriptions and theories. The models to be defined are for complex computing phenomena, for which we define generalized diagrams.

The techniques in (Nourani 1983,87,91,94a) for model building as applied to the problem of AI reasoning allows us to build and extend models through diagrams. It required us to define the notion of generalized diagram. We had invented G-diagrams(Nourani 1987,91,93b,94a) to build models with prespecified generalized Skolem functions. The specific minimal set of function symbols is the set with which a model can be inductively defined. We focus our attention on such models, since they are Initial and computable models defined by our papers since 1978 (Nourani 1984,91,94a). The G-diagram techniques allowed us to formulate AI world descriptions, theories, and models in a minimal computable manner. Thus models and proofs for AI and computing problems can be characterized by models computable by a set of functions.

3.3.2 Algebraic Tree Computation

The techniques in (Nourani 1991,94a) for model building as applied to the problem of AI reasoning allows us to build and extend models through diagrams. A technical example of algebraic models defined from syntax had appeared in defining initial algebras for equational theories of data types (ADJ 1973) and our research in (Nilsson 1969). In such direction for computing models of equational theories of computing problems are presented by a pair (Σ,E), where Σ is a signature (of many sorts, for a sort set S) (ADJ 1973,Nourani 1995a) and E a set of Σ-equations. Let $T<\Sigma>$ be the free tree word algebra of signature Σ. The quotient of $T<\Sigma>$, the word algebra of signature Σ, with respect to the Σ-congruence relation generated by E, will be denoted by $T<\Sigma,E>$, or $T<P>$ for presentation P. $T<P>$ is the "initial" model of the presentation P. One representation of T(P) which is nice in practice consists of an algebra of the canonical representations of the congruence classes. This is a special case of generalized standard models defined here. In what follows gt1...tn denotes the formal string obtained by applying the operation symbol g in Σ to an n-tuple t of arity corresponding to the signature of g. Furthermore, gC denotes the function corresponding to the symbol g in the algebra C. We present some definitions from our papers (Nourani 1983,84, 91) that allow us to define standard models of theories that are Σ-CTA's. The standard models are significant for tree computational theories that we had presented in (Nourani 1984,Nilsson 1969) and the intelligent tree computation theories developed by the present paper. A standard model M, with base M and functionality F, is a structure inductively defined by $<F,M>$ provided the $<F,M>$ defines an initial structure with sets disjoint except fro perhaps the constant basis. We will review these definitions in the sections to follow.

3.3.3 G-diagrams of Initial Models

The generalized diagram (G-diagram) (Nourani 1991,94a,93b) is a diagram in which the elements of the structure are all represented by a minimal set of function symbols and constants, such that it is sufficient to define the truth of formulas only for the terms generated by the minimal set of functions and constant symbols. Such assignment implicitly defines the diagram. This allows us to define a canonical model of a theory in terms of a minimal family of function symbols. The minimal set of functions that define a G-diagram are those with which a standard model could be defined by a monomorphic pair. Formal definition of diagrams are stated here, generalized to G-diagrams, and applied in the sections to follow.

Definition 3.1 A G-diagram for a structure M is a model-theoretic diagram definable by a specific function set. .

Remark: The minimal set of functions is the set by which a standard model could be defined. Thus initial models could be characterized by their G-diagrams. Further practical and the theoretic characterization of models by their G-diagrams are presented by this author in (Nourani 1994a). It builds the basis for some forthcoming formulations that follow, and the tree computation theories that we had put forth in (Nilsson 1969). Here we showed how initial models could appear out of thin air within our formulation. Initial models are defined by algebraic tree rewriting for the intelligent languages is developed from our 1979-84 papers and (Nourani 1984). We showed how initial algebras can be defined by subtree replacement and tree rewriting These are the minimal set of functions that by forming a monomorphic pair with the base set, bring forth an initial model by forming the free trees that define it. Thus an initial free model is formed. The model can be obtained by algebraic subtree replacement systems. The G-diagram for the model is also defined from the same free trees. The conditions of the theorems are what you expect them to be- that canonical subset be closed under constructor operations, and that operations outside the constructor signature on canonical terms yield canonical terms.

4. Intelligent Languages, and Models

4.1 Intelligent Syntax

By an intelligent language we intend a language with syntactic constructs that allow function symbols and corresponding objects, such that the function symbols are implemented by computing agents in the sense defined by this author in (Nourani 1993c, 96a). Sentential logic is the standard formal language applied when defining basic models. The language $\mathcal{L}$ is a set of sentence symbol closed by finite application of negation and conjunction to sentence symbols. Once quantifier logical symbols are added to the language, the language of first order logic can be defined. A Model $\mathcal{A}$ for $\mathcal{L}$ is a structure with a set A . There are structures defined for $\mathcal{L}$ such that for each constant symbol in the language there corresponds a constant in A. For each function symbol in the language there is a function defined on A; and for each relation symbol in the language there is a relation defined on A. For the algebraic theories we are defining for intelligent tree computing in the forthcoming sections the language is defined from signatures as in the logical language is the language of many-sorted equational logic. The signature defines the language $\mathcal{L}$ by specifying the function symbols' arities. The model is a structure defined on a many-sorted algebra consisting of S-indexed sets for S a set of sorts. By an intelligent language we intend a language with syntactic constructs that allow function symbols and corresponding objects, such that the function symbols are implemented by computing agents. A set of function symbols in the language, referred to by AF, is the set modeled in the computing world by AI Agents with across and/or over board capability. Thus the language $\mathcal{L}$ defined by the signature has designated function symbols called AF. The AF function symbols define signatures which have specific message paths defined for carrying context around an otherwise context free abstract syntax. A set of function symbols in the language, referred to by AF, are agents with nontrivial capability. The boards, message passing actions, and implementing agents are defined by syntactic constructs,with agents appearing as functions. The computation is expressed by an abstract language that is capable of specifying modules, agents, and their communications. We have put together the AI concepts with syntactic constructs that could run on the tree computing theories we are presenting in brief. We have to define how the syntactic trees involving functions from the AF are to be represented by algebraic tree rewriting.on trees. This is the subject of the next section, where free intelligent trees are defined. An important technical point is that the for agents there are function names on trees.

Definition 4.2 We say that a signature Σ is intelligent iff it has intelligent function symbols. We say that a language has intelligent syntax if the syntax is defined on an intelligent signature

Definition 4.3 A language L is said to be an intelligent language iff L is defined from an intelligent syntax .

4.2 Abstract Syntax Information

It is essential to the formulation of computations on intelligent trees and the notions of congruence that we define tree information content of some sort. A reason is that there could be loss of tree information content when tree rewriting because not all intelligent functions are required to be mutually informable. What we have to define, is some tree computational formulation of information content such that it applies to the intelligent computability theory proposed. Once the formulation is presented, we could start decorating the trees with it and define computation on intelligent trees. The example of intelligent languages we could present are composed from <O,A,R> triples as control structures(Nourani 1993f). The A's have operations that also consist of agent message passing. The functions in AF are the agent functions capable of message passing. The O refers to the set of objects and R the relations defining the effect of A's on objects. Amongst the functions in AF only some interact by message passing. The functions could affect objects in ways that affect the information content of a tree. There you are: the tree congruence definition thus is more complex for intelligent languages than those of ordinary syntax.trees. Let us define tree information content for the present formulation.

Definition 4.4 We say that a function f is a *string* function iff there is no message passing or information exchange except onto the board agents that are at the range set for f, reading parameters visible. Otherwise, f is said to be a *splurge* .

Remark: Nullary functions are information strings.

Definition 4.5 The tree information content, TID, is defined by induction on tree structures:
(0) the information content of a constant symbol. function f is f; (i) for a string function f, and tree f(t1,...,tn) the TID is defined by U TID (ti::f) , where (ti::f) refers to a subtree of ti visible to f;
(ii) for a splurge function f, TID is defined by U TID (f:ti), where f:ti refers to the tree resulting from ti upon information exchange by f.

The techniques for our new intelligent object level programming paradigm are implicit in the above definition, in for example, the concept of a subtree being visible to a function. The theorem below formalizes these points. Thus out of the forest of intelligent trees, not to overstate the significance, there appears a sudden information theoretic rewrite theorem. Ordering on information content is equivalent to subset ordering on TID sets. A rewrite rule on a set of trees with TID set S has information loss if the TID of the resulting rewritten tree is a subset of S. For the chess model defined the pawn agents are trivial string functions with MIC connected and known to their sphere AF agents. We refer to it by the *pawn string rule.*

There would alwaysbe an AF sphere agent defined. The king designated agent becomes the sphere agent for the pawn string rule only when AF becomes the empty set.

Theorem 4.1 Trees on intelligent syntax, rewritten guided only by what equations state, could cause a loss of information content between trees.

Proof Trees with AF functions by definition affect TID, thus a rewrite from a tree formed by a function g in AF to a tree that does not have g as a function symbol causes an information loss. For example, a trivial equation of the form f-1(f(t)) = t, where f is in AF causes an information loss to the resulting set of trees, from the left hand to the right hand tree t.

4.2.1 The Capture Agents

The positive side for theorem 4.1 can be seen by defining Capture Agents. As an example for how the agent intelligent tree computing might be applied to ordinary tree rewriting we had defined (Nourani. 1993e) *completion agents.* The tree completion agents are indicators as to when all terms called for by an. agent function are in normal form. A term of the form f(t1,...,tn) carries an agent function on each term for the terms subtree sort completion agent. When. term ti is in normal form its completion agent function is dissolved as promised by theorem 4.1. It no longer appears on the tree. We define a similar notion called capture Agents. Capture Agent are what ultimately return a tree via a completion agent, where on the normalized tree the captured piece's designated agent is no longer present. The capture agent indicates the piece has been captured.

4.2.2 Intelligent Theories

Let us formulate proof systems for intelligent equational theories.

Definition 4.6 We say that an equational theory T of signature IΣ is an intelligent IΣ theory iff for every proof step involving tree rewriting, the TID is preserved. We state T<IST> |- t=t' when T is an IΣ theory.

Definition 4.7 We say that an equational theory T is intelligent, iff T has an intelligent signature IΣ,and axioms E, with IΣ its intelligent signature. A proof of t=t' in an intelligent equational theory T is a finite sequence b of IΣ-equations ending in t=t' such that if q=q' is in b, then either q=q' in E, or q=q' is derived from 0 or more previous equations in E by one application of the rules of inference. Write T <IST>⊢t=t' for "T proves t=t' by intelligent algebraic subtree replacement system."

By definition of such theories proofs only allow tree rewrites that preserve TID across a rule. These definitions may be applied to prove rewriting theorems,to set up the foundations for what could make intelligent tree rewriting TID, and define intelligent tree computation. Thus the essence of intelligent trees will not be lost while rewriting. Next, we define a computing agent function's information content from the above definition. This is not as easy as it seems and it is a matter of the model of computation applied rather than a definition inherent to intelligent syntax. Let me make it a function of intelligent syntax only, because we are to stay with abstract model theory and build models from abstract syntax. The definition depends on the properties of intelligent trees, to be defined in the following section.

5. Intelligent Trees

5.1 Embedding Intelligence

Viewing the methods of computation on trees presented in the sections above we define intelligent trees here.

Definition 5.8 A tree defined from an arbitrary signature Σ is intelligent iff there is at least one function symbol g in Σ such that g is a member of the set of intelligent functions AF, and g is a function symbol that appears on the tree.

Definition 5.9 We define an intelligent Σ-equation, abbreviated by IΣ-equation, to be a Σ-equation on intelligent Σ-terms. An IΣ congruence is an Σ-congruence with the following conditions:

(i) the congruence preserves IΣ equations;

(ii) the congruence preserves computing agents information content of Σ-trees.

Definition 5.10 The mutual information content, MIC, of an intelligent function f, a member of the intelligent signature AF, is determined by the IΣ-congruence on T<AF> relating the functions in AF. It is union of the TID over the trees that are a member of the congruence class of the free T<AF> with respect to the IΣ-congruence defined on the T<Σ,w>, where w is the arity of f.

Let Σ be an intelligent signature. We have defined canonical term IΣ-algebra (IΣ-CTA) in (Nourani 1993e,96a) as term algebra models for intelligent signatured theories. The validity of propositions can be ascertained by checking them for canonical terms at the canonical initial model.

5.2 Intelligent Rewrite Models

Term rewrite model theorems for intelligent syntax

Lemma 5.1 Let R be a set of IΣ-equations. Let R be the set of algebraic IΣ-rewrite rules obtained by considering each equation l =r in Ro as a rule l => r, then for t,t' in T<Σ>, t => * t' iff T(R) <IST>⊢t = t'..

Recall that a presentation (Σ,E) defined an equational theory of signature Σ and axioms E. Next we show how canonical models can be constructed by algebraic subtree replacement system. A definition and what we have done thus far (Nourani 1984,Nilsson 1969) for how to represent normal forms with canonical terms and their relations to defining models gets us to where we want to go: the canonical algebraic intelligent term rewriting theorems. We say that (C,R) represents a IΣ-algebra A iff the IΣ-algebra so defined by (C,R) is IΣ-isomorphic to the algebra A presented by the axioms. We have proved (Nourani 1993e,96a) theorem for sufficient conditions for constructibility of an initial model for an IΣ equational presentation. It is the mathematical justification for the proposition that initial models with intelligent signature can be automatically implemented (constructed) by algebraic subtree replacement systems, with normal forms defined by a minimal set of functions that are Skolem functions or type constructors. The Canonical Intelligent Model Theorems are proved in (Nourani 1993e,96a). The theorems provide conditions for automatic implementation by intelligent tree rewriting to initial models.

Theorem 5.3 (The MIC Theorem) Let P be a presentation with intelligent signature IΣ for a computing theory T with intelligent syntax trees. Then T is (a) A Sound logical theory iff every axiom or proof rule in T is TID preserving;

(b) A Complete logical theory iff there is a <fucntion,set> pair defining a canonical structure C and a G-diagram, such that C with R represents T<IΣ,R>, where R is the set R of axioms for P viewed as IΣ-rewrite rules.

Proof By Definition of MIC , theorems on canonical intelligent models(Nourani 1993e, 96a), completeness theorems for the first order logic, and completeness of induction for algebraic structures (Nourani 1994b). We had called the above theorem the logical foundations MIC theorem for intelligent trees. We have begun to present MIC theorems for the information theoretic properties of the present game tree computing theories(Nourani 1993e,96a). The above with the pawn string rule stated at the last part sections 4.2 are sufficient to define the game moves.

6. Computing On Intelligent Trees

6.1 Tree Computing For AI

We present a brief overview of the applications of our methods to AI planning problems in (Nourani 1991,93c,94a). We have proposed (Nourani 1994a) methods that can be applied to planning with GF-diagrams(Nourani 1995f) with applications to some current research directions in AI. The techniques can be applied to implement planning and reasoning for goal directed move planning. While planning with GF-diagrams that part of the plan that involves free Skolemized trees is carried along with the proof tree for a plan goal. The idea is that if the free proof tree is constructed then the plan has an initial model in which the goals are satisfied (Nourani 1984,94a,93b).

...........6.2 The Single Agent Moves

The present computational model for multi-agent system provides the following basis for single agent decisions. For each agent function there is a MIC determining its mutual information content with respect to the decision trees connected to it. A single agent makes its decisions for each operation or action by computing a plausible next move set. The plausible next move set might have dynamic properties. For example, it might consist of a set of trees bearing agent functions, which compute their next move sets to update the computing trees MIC. The interplay of intelligent syntax and our intelligent syntax computability techniques (Nourani 1993d) could provide us with exciting expressive languages and techniques for trees, by desiging programs for single agent functions.

6.3 Chess Thinking Machines

To play muliagent chess the mind has to be capable to distribute and partition the board as an isomorphic image to the multiagent multiboard model. The master player man or machine'smind might be modelled by a multiagent-multiboard structure. We do not wish to enter a new debate on man-machine intelligence issue which AI has written for during the last many years (Minsky 1986). However, we might address the questions on thinking machine epistemicis and the sufficient/necessary conditions for "sensing" intelligence. The intelligent tree computing basis to chess for a machine designed with the agent capabilities defined and learning degrees, might be certified "Thinking" relative to the chess paradigm. Thinking in the absolute is a difficult proposition to establish since it is difficult to universally define for all domains. However, thinking relative to a well-defined problem such as chess is not as inconceivable. Such competence might be bestowed onto a machine via a multiagent paradigm. A chess playing machine with specific problem solving plans, offensive and defensive tactics, which can anticipate opponent's moves and game plans, is well on its way to be eligible as a thinking machine.

7. Is Chess A Two-Person PI Game

From the game tree view point for what Shannon had estimated a complete tree carried to depth 6-three moves for each player- would already have one billion tip nodes. Yet from an abstract mathematical view point only, the game is a two-person game with perfect information. However, the two-person game view is not a mathematical model for any chess playing algorithm or machine. The real chess game, from the abstract view point, might well be modeled as a multiagent game, being only a two-person game with perfect information between mutually informable agents. There is only minimal information for the multiagent plans across the board. The multiagent multiboard model is a way to come to the realization that the game is partitioned and correlated amongst agents and boards, with a cognitive anthropomorphism to human player's mind. There is an abstract two-person game model, but it does not apply to define a chess playing machine. It is there to make precise mathematical statements.

8. Concluding Comments

What do the man-machine philosophical dilemma and epistemology imply for chess machines. In papers written by Nourani93-97 a new area called descriptive computing is defined (Nourani 1993a,94d,96b,97a) where games and winning strategies are defined in terms of diagrams for computing epistemics.The emotional reaction to a very notion of artificial intelligence might be dispelled once we view the game epistemics as being a mind image for a muliagent cognition, relative to a specific defined problem, and not in the absolute. As far as Heidegger and technology are concerned I might expound to a degree to (Nourani 1993a). There is a positive side to the views to technology. Heidegger states for man it means being at home with something. The papers in (Nourani 1993a,94d) had move to the positive side of Vergessenheit, defining what the game moves might imply on a reasoning diagram by applying dynamic epistemic computing. Hence it is only defining what it means for the mind to be at home with games. Vergessenheit's negative aspects are what happens when the situation's dynamics are never defined, e.g. the chess piece's agent epistemics are not defined. A move is not well-defined for a piece for either the human mind or the machine at such Vergessenheit. From the AI view point we have defined a preliminary basis for multiagnet multiboard chess. The mathematics for intelligent game trees is defined with soundness and completeness theorems for the tree computing with agents on intelligent trees.

References

Nourani, C.F., 1984, "Positive Thinking On Trees," 1984, Brief Version Appeared at ASL and LICS, Stanford, July 1985.

Nilsson, N.J. 1969,"Searching, problem solving, and game-playing trees for minimal cost solutions." In A.J. Morell (Ed.) IFIP 1968 Vol.2, Amsterdam, North-Holland, 1556-1562, 1969.

Nourani,C.F.,1987,"Diagrams, Possible Worlds, and the Problem of Reasoning in Artificial Intelligence," **Logic Colloquium**, Padova, Italy, 1987, Proc.in Journal of Symbolic Logic.

Nourani,C.F. 1991,"Planning and Plausible Reasoning in AI,"**Proc. Scandinavian Conference in AI**, May 1991, Denmark, 150-157, IOS Press.

Nourani,C.F.1993a,"Dynamic Epistemic Computing And Machine Learning On Diagrams," 1993, Revised 1996..

Nourani,C.F. 1994a,"GF-Diagrams For Models and Free Proof Trees," March 26, 1994, Berlin Logic Colloquium, May 1994.

Nourani, C.F. 1993b, "Computation on Diagrams and Free Skolemization,"**Summer Logic Colloquium**, July 1993, England.

Nourani,C.F.1983,"Equational Intensity, Initial Models, and AI Reasoning, Technical Report, 1983, : A: Conceptual Overview, in **Proc. Sixth European Conference in Artificial Intelligence**, Pisa, Italy, September 1984, North-Holland.

ADJ 1973,Goguen, J.A., J.W. Thatcher, E.G. Wagner and J.B. Wright,"A Junction Between Computer Science and Category Theory," (parts I and II),IBM T.J. Watson Research Center, Yorktown Heights, N.Y. Research Report, RC4526,1973.

Nourani,C.F.1995a,"Linguistics Computation, Automatic Model Generation, and Intensions,"Brief Versions Appeard at Nordic and General Lingusitics and Scandinvians Linguistics, January 1995, Oslo, Norway.

Nourani,C.F,1993c "AII,"February 11, 1993, Applications to Heterogenous Design, MAAMAW-97, Ronneby, Sweden, May 1997.

Newborn, M 1997, "Kasparov versus Deep Blue:Computer Chess Comes of Age", Springer, 1997.

Minsky,M 1986, The Sociey of Mind, Simon-Schuster, 1986.

Nourani,C.F.1994b,"Induction, Completeness, and Incompleteness For Abstract Automated Proofs,"Technical Report, February 1994, EATCS.

Nourani,C.F.1993d,"A Theory of Programming With Intelligent Syntax Trees- A Preliminary View,"May 1993.

Genesereth,M.R. and N.J. Nilsson 1987, Logical Foundations of Artificial Intelligence, Morgan-Kaufmann, 1987.

Nourani,C.F. 1994c, "A Theory For Programming With Intelligent Syntax Trees And Intelligent Decisive Agents-Prelimary Report" 11th European Conference A.I., ECAI Workshop on DAI Applications to Decision Theory,Amsterdam, Augus 1994

Nourani, C.F.1995b, "Double Vision Computing", December 1993, IAS-4, Karlsruhe, Germany April 1995.

Nourani,C.F. 1993e,"Computing With Intelligent Trees" April 1993.

Nourani,C.F.1996a,"Slalom Tree Computing," 1994, AI Communications, November 1996.

Berliner, H,J.1978,"A chronology of compuer chess and its literature. AI 10, 201-214, 1978.

Hays-Roth, B 1985,"A Balckboard Architecture For Control,"*Artificial Intelligence, 26*(3), 1985, 251-321.

Newell,A, Shaw,J.C., and Simon,H.A.1963,"Chess Playing Programs and The Problem of Complexity," In Feigenbaum-Feldman, 39-70, 1963.

Shannon, C.E.1956,"A Chess Playing Machine." In J.R. Newman (ed.) The World of Mathematics (Vol.4), New York, Simon-Schuster, 2124-2133, 1956.

Nourani,C.F.1995f,"Free Proof Trees and Model-theoretic Planning, February 23, 1995, Automated Reasoning AISB, England, April 1995.

Nilsson, N.J.1971, Problem Solving Methods in Artificial Intelliegence," New York, McGraw-Hill, 1971.

Nourani,C.F.,1993f"Abstract Implementations Techniques For AI Systems By Computing Agents-A Conceptual Overview,"March 3, 1993, **Proc. SERF-93**, Orlando, FL, November 1993.

Nourani,C.F.,'1997a,"Descriptive Computing,"The Mathematics Applications For A.I. AMS, Memphis, April, 1997.

Nourani,C.F.1994d,"Towards Computational Epistemology- A Forward," Proc. Summer Logic Colloquium, Clermont-Ferrand, France, July 1994.

Nourani,C.F.1996b,"Descriptive Computing-The Preliminary Definition, Summer Logic Colloquium, July 1996, San Sebastian, Spain.