Competitive Models, Predictive Set Game Trees, and Data Analytics

Cyrus F Nourani* and Associates: AI TU Berlin
Acdmkrd@gmail.com

August 2017

Abstract: We can examine random sets as a basis to carry structures modeling towards a competitive culmination problem where models "compete" based on modeling game trees. Competitive game tree modeling and predictive game logic is briefed. The techniques are developed on descriptive game models and compatibility is characterized. Specific game models are presented to illustrate the techniques. A model rank is higher when a on game trees with a higher game tree degree, satisfies goals, hence realizing specific models where the plan goals are satisfied. Characterizing Competitive Model Degrees on Random Sets is a basis area to explore. A model is a competing model iff at each stage the model is compatible with the goal tree satisfiability criteria. Compatibility is defined on Random Sets where the correspondence between compatibility on random sets and game tree degrees are applied to present random model diagrams. Random diagram game degrees are applied and model ranks based on satisfiability computability to optimal ranks are examined.
Keywords: Competitive Model Computing, Game Degrees, Random sets and model compatibility, Random Model Diagrams, Model Rank Computability, Predictive Data Analytics

* Affiliation AI-TU Berlin, DE & SFU
Acdmkrd@gmail.com : On Singularity University Hub
University E-mail: cyrusfn@alum.mit.edu

1. Introduction

Games play an important role as a basis to economic theories. Here the import is brought forth onto decision tree planning with agents. The author had presented specific agent decision tree computing theories since 1994. and can be applied to present precise strategies and prove theorems on multiplayer games. Game tree degree with respect to models is defined and applied to prove soundness and completeness. The game is viewed as a multiplayer game with only perfect information between agent pairs. Upper bounds on determined games were presented on first authors publications stated. A technique for modeling game tress satisfiability is based on competitive models (Nourani 2008) and section 2. The present paper is a preliminary basis to carry on competitive model satisfiability as a basis to optimized decisions based on random sets (Martin Lof 1966). Random sets are random elements taking values as subsets of some space, are a mathematical model for set-valued observations and irregular geometrical patterns. Random sets in stochastic geometry (Kendall 1974), are examples. Besides sampling designs, confident regions, stochastic geometry and morphological problems, random sets

appear in general as set-valued observed processes. The concept of random sets has been carried onto random fuzzy sets to model perception-based information in social systems, artificial intelligence problems such as intelligent control and decisions.
Our specific paper addresses the question: how to model competitive model computing planning with random set. The paper develops a novel model diagram tchniques to carry competitive model planning to reach specific goals or to carry on game tree objectives. This section begins to develop competitive model game trees brief from the first authors recent publications. Section 3 starts to characterize competitive model degrees based on random sets, where on section form non-determinisitc diagrams are applied to compatibility on models. Computations geometry on random algorithms is previewed to projections on Boolean valued maps to product random sets. Model ranks are presented based on random model diagrams. Section 5 defined random diagram game trees where computability questions on model compatibly are addressed and model ranking complexity is examined.

2. Competitive Models and Game TREES

Planning is based on goal satisfaction at models. We can examine random sets as a basis to carry structures modeling as a competitive culmination problem where models "compete" based on modeling game trees, where the model rank is higher when a on game trees with a higher game tree degree, satisfies goals, hence realizing specific models where the plan goals are satisfied. When a specific player group "wins" to satisfy a goal the group has presented a model to the specific goal, presumably consistent with an intended world model. For example, if there is a goal to put a spacecraft at a specific planet's orbit, there might be competing agents with alternate micro-plans to accomplish the goal. While the galaxy model is the same, the specific virtual worlds where a plan is carried out to accomplish a real goal at the galaxy via agents are not. Therefore, Plan goal selections and objectives are facilitated with competitive agent learning. The intelligent trees (Nourani 1994,1996) are ways to encode plans with agents and compare models on goal satisfaction to examine and predict via model diagrams why one plan is better than another or how it could fail. Virtual model planning is treated in the author's publications where plan comparison can be carried out at VR planning (Nourani 1999b).

3. Characterizing Competitive Model Degrees on Random Sets

Let us start with certain basic premises.
A model is a competing model iff at each stage the model is compatible with the goal tree satisfiability criteria. Compatibility defined on Random Sets In computational geometry, a standard technique to build a structure like a convex hull is to randomly permute the input points and then insert them one by one into the existing structure. The randomization ensures that the expected number of changes to the structure caused by an insertion is small, and so the expected running time of the algorithm can be upper bounded. This technique is called randomized incremental construction[3]. Graph problems are another area that Randomized algorithms are applied, for example, a randomized minimum cut algorithm:
find_min_cut(undirected graph G) { while there are more than 2 nodes in G do { pick an edge (u,v) at random in G contract the edge, while preserving multi-edges remove all loops } output the remaining edges } *There* are various notions of algorithmic

randomness. Relativized randomness is where. A set is random if it is Martin-Löf random relative to (n-1). For example (Nielsen et all 2005) show that a set is 2-random if and only if there is a constant c such that infinitely many initial segments x of the set are c-incompressible. *Let us develop model compatibility on random sets and game tree degrees. We begin with model* computing on structures that are definable with certain basic functions and constants. That is an abstract computing that is carried on with functions that are computable, thus structures that are defined by computable functions. When you have such structures, you can check what is true on the structure with respect to statements stated with first order logic, equational, or Horn clauses, that can take free assignment to variables to logical formulas that can be stated on tree terms defined with the computable functions. That brings us to two definitions.

Definition 3.1 A generic model diagram is a diagram with which a model is characterized with specific functions, e.g. specific Skolem functions.

The computing specifics are based on creating models from generic model diagram functions where basic models can be piece-meal designed and diagrams completed starting from incomplete descriptions at times. Models uphold to a deductive closure of the axioms modeled and some rules of inference, depending on the theory. By the definition of a diagram they are a set of atomic and negated

4. Random Sets and Nonderminism

4.1 Computational Model Specifics

The computing specifics are based on creating models from generic model diagram functions where basic models can be piece-meal designed and diagrams completed starting from incomplete descriptions at times. Models uphold to a deductive closure of the axioms modeled and some rules of inference, depending on the theory. By the definition of a diagram they are a set of atomic and negated atomic sentences. Thus, the diagram might be considered as basic for a model, provided we can by algebraic extension, define the truth value of arbitrary formulas instantiated with arbitrary terms. Thus all compound sentences build out of atomic sentences then could be assigned a truth-value, handing over a model. This will be made clearer in the following subsections. The following examples would run throughout the paper.
Computing models are structures where certain computation properties stated on a logic with the functions defining the structures can be ascertained true or false. Let us consider computing on structures on a specific signature that can take assignments to variables from a random set. We can rank formulas with ranking based on a measure rank: Formulas with ranking less than a specific value would be assigned 'T' and the other formulas would be assigned `F.' Corresponding to Possible Worlds we can define Random worlds where we can consider a formula to be true iff it is true with all such random assignments. The first author had defined the notion of a plausible diagram, which can be constructed to define plausible models for revised theories. In practice, one may envision planning with plausible diagrams such that certain propositions are deliberately left indeterminate to allow flexibility in planning. Such extensions to the usual notion of diagram in model theory are put forth around 1987. That approach was

one method of avoiding the computational complexity and computability problems of having complete diagrams. Truth maintenance and model revision can all be done by a simple reassignment to the diagram. The canonical model of the world is defined directly from the diagram. Corresponding to the above we have: Random Model diagrams.

4.2 Competitive Models and Compatibility

Now let us examine the definition of situation and view it in the present formulation.

Definition 4.1 A situation consists of a nonempty set D, the domain of the situation, and two mappings: g,h. g is a mapping of function letters into functions over the domain as in standard model theory. h maps each predicate letter, pn, to a function from Dn to a subset of {t,f}, to determine the truth value of atomic formulas as defined below. The logic has four truth values: the set of subsets of {t,f}.{{t},{f}, {t,f},0}. the latter two is corresponding to inconsistency, and lack of knowledge of whether it is true or false.

Due to the above truth values, the number of situations exceeds the number of possible worlds. The possible worlds being those situations with no missing information and no contradictions. From the above definitions the mapping of terms and predicate models extend as in standard model theory. Next, a compatible set of situations is a set of situations with the same domain and the same mapping of function letters to functions. In other words, definition 5.2 has a proper definition by specific function symbols. Remark: The functions above are those by which a standard model could be defined by inductive definitions. What it takes to have an algebra and model theory of epistemic states, as defined by generalized diagram of possible worlds is exactly what (Nourani 98, 91) had accomplished To decide compatibility of two situations we compare their generalized diagrams. Thus, we have the following Theorem.

Theorem 4.1 Two situations are compatible iff their corresponding generalized diagrams are compatible with respect to the Boolean structure of the set to which formulas are mapped (by the function h above, defining situations).

Proof e.g. (Nourani 2015).

The generic diagrams, definitions above encode possible worlds and since we can define a one- one correspondence between possible worlds and truth sets for situations, computability is definable by the generic-diagrams.

Proposition 4.1 Computational geometry with random sets can be characterized as a projection to a Boolean valued function on random diagrams on the corresponding geometry to a product random set.

4.3 Predictive Competitive Model Game Trees

Minimal prediction is a technique defined since the authors model- theoretic planning project. It is a cumulative nonmonotonic approximation attained with completing model diagrams on what might be true in a model or knowledge base. A predictive diagram for a theory T is a diagram D (M), where M is a model for T, and for any formula q in M, either the function f: q → 0,1 is defined, or there exists a formula p in D(M), such that T ∪ p proves q; or that T proves q by minimal prediction. A generalized predictive diagram is a predictive diagram with D (M) defined from a minimal set of functions. The predictive diagram could be minimally represented by a set of functions f1,...,fn that inductively define the model. The free trees we had defined by the notion of provability implied by the definition, could consist of some extra Skolem functions g1,...,gl that appear at free trees. The f terms and g terms, tree congruences, and predictive diagrams then characterize fragment deduction with free trees. The predictive diagrams are applied to discover models for game trees.

Theorem A set of first order definable game tree goals G is attainable iff there exists a predictive diagram for the logical consequences to G on the game tree model.

5. Ranks, Models, and Random Diagrams

Based on game trees on competitive models : AND/OR trees Nilsson(1969) are game trees defined to solve a game from a player's standpoint. Formally a node problem is said to be solved if one of the following conditions hold. 1. The node is the set of terminal nodes (primitive problem- the node has no successor). 2. The node has AND nodes as successors and the successors are solved. 3. The node has OR nodes as successors and any one of the successors is solved.

A solution to the original problem is given by the subgraph of AND/OR graph sufficient to show that the node is solved. A program which can play a theoretically perfect game would have task like searching and AND/OR tree for a solution to a one-person problem to a two-person game. An intelligent AND/OR tree is and AND/OR tree where the tree branches are intelligent trees. The branches compute a Boolean function via agents. The Boolean function is what might satisfy a goal formula on the tree. An intelligent AND/OR tree is solved iff the corresponding Boolean functions solve the AND/OR trees named by intelligent functions on the trees.

Thus node m might be f(a1,a2,a3) & g(b1,b2), where f and g are Boolean functions of three and two variables, respectively, and ai's and bi's are Boolean valued agents satisfying goal formulas for f and g.

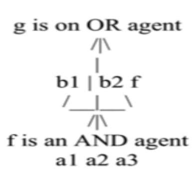

Figure 1 Agent and/or trees

The chess game trees can be defined by agent augmenting AND/OR trees (Nilsson 69). For the intelligent game trees and the problem-solving techniques defined, 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-cupboard agent computation is depicted on a tree. Whereas the satespace trees for each agent is determined by the computation sequence on its side of the board-cupboard. Thus a tree node m might be f(a1,a2,a3) & g(b1,b2), where f and g are Boolean functions of three and two variables, respectively, and ai's and bi's are Boolean valued agents satisfying goal formulas for f and g. A tree game degree is the game state a tree is at with respect to a model truth assignment, e.g. to the parameters to the Boolean functions above. Let generic diagram or G-diagrams be diagrams definable by specific functions. We can then rank models based on game-tree satisfiability on a specific game tree degree. Thus we have the model closest to a win when ranks higher on satisfiability.

Definition 5.1 A random diagram game tree is a game tree where assignments to variables is defined on a Boolean function on a specified random set. We can then rank models based on game-tree satisfiability on a specific game tree degree. Thus we have the model closest to a win when ranks higher on satisfiability. Based on the above we can state basic theorems that

Proposition 5.1 A model has optimal rank ifff the model satisfies every plan goal and has the lowest highest game tree degree.

Theorem 5.1 There are computable models where optimal ranks can be determined.

Theorem 5.2 Based on computable models with computable diagrams model compatibility is effectively computable.

5.1 Random Models and Big Data Heuristics

Let us now view the deductive methods, for example the proof- theoretic example :SLDNF- resolution, a well-known deductive heuristic. A SLDNF-proof can be considered as the unfolding of an AND/OR-tree, which is rooted in the formula to be proven, whose branches are determined by formulas of the theory, and whose leaves are determined by atomic formulas which are true in a world. Partial deduction from our view point (Nourani-Hoppe 1995) usually computes from a formula and a theory an existential quantified diagram. In these papers and (Nourani 1995,2005) we also instantiate proof tree leaves with free Skolemized trees, where free trees are substituted for the leaves. By a free Skolemized tree we intend a term built with constant symbols and Skolem functions terms. Dropping the assumption that proof-tree leaves get instantiated with atomic formulas only yields an abstract and general notion of proof trees. The mathematical formalization that allows us to apply the method of free proof trees is based on the first author’s 1995-2005. In the present approach, as we shall further define, leaves could be free Skolemized trees. By a free Skolemized tree we intend a term made of constant symbols and Skolem function terms. Like models and diagrams, which where generalized above in different ways, we can generalize the notion of a proof.

First author had developed free proof tree techniques since projects at TU Berlin, 1994. Free proof trees allow us to carry on Skolemized tress on game tree computing models, for example, that can have unassigned variables. The techniques allow us to carry on predictive model diagrams. Reverse Skolemization (Nourani 1986) that can be carried on with generic model diagrams corresponding to what since Genesereth (2011) is applying on game tree "stratified" recursion to check game tree computations. The free trees defined by the notion of provability implied by the definition, could consist of some extra Skolem functions{g1,...,gm},that appear at free trees. The f terms and g terms, tree congruences, and predictive diagrams then characterize partial deduction with free trees. To compare recursive stratification on game trees on what Geneserth calls recursive stratification we carry on models that are recursive on generic diagram functions where goal satisfaction is realized on plans with free proof trees (Nourani 1994-2007).

Thus essentially the basic heurstics here is satisfying nodes on agent AND./OR game trees. The general heuristics to accomplish that are a game tree deductive technique based on computing game tree unfoldings projected onto predictive model diagrams. The soundness and completeness of these techniques, e,g, heuristics as a computing logic is published since (Nourani 1994) at several events e.g. AISB 1995, and Systems and Cybernectics 2005), (Nourani 2015). In computer science, specifically in algorithms related to pathfinding, a heuristic function is said to be admissible if it never overestimates the cost of reaching the goal, i.e. the cost it estimates to reach the goal is not higher than the lowest possible cost from the current point in the path. An admissible heuristic is also known as an optimistic heuristic.

An admissible heuristic is used to estimate the cost of reaching the goal state in an informed search algorithm. In order for a heuristic to be admissible to the search problem, the estimated cost must always be lower than or equal to the actual cost of reaching the goal state. The search algorithm uses the admissible heuristic to find an estimated optimal path to the goal state from the current node.

An admissible heuristic can be derived from a relaxed version of the problem, or by information from pattern databases that store exact solutions to subproblems of the problem, or by using inductive learning methods. Here we apply the techniques on goal satisfiability on competive models brifed on the preceding section. While all consistent heuristics are admissible, not all admissible heuristics are consistent. For tree search problems, if an admissible heuristic is used, the A* search algorithm will never return a suboptimal goal node.

The heuristic nomenclature indicates that a heuristic function is called an admissible-heuristic if it never overestimates the cost of reaching the goal, i.e. the cost it estimates to reach the goal is not higher than the lowest possible cost from the current point in the path. An admissible heuristic is also known as an optimistic heuristic (Russell and Norvig 2002). What is the cost estimate on satisfying a goal on an unfolding projection to model diagrams, for example with SLNDF, to satisfy a goal? Our heuristics are based on satisfying nondeterministic Skolemized trees. The heuristics aims to decrease the unknown assignments on the trees. Since at least one path on the tree must have all assignments defined to T, or F, and at most one such assignment closes the search, the "cost estimate," is no more than the lowest.

6. Areas to further Explore

Practical AI systems are designed by modeling AI with facts, rules, goals, strategies, knowledge bases. Patterns, schemas, AI frames and viewpoints are the micro to aggregate glimpses onto the database and knowledge bases were masses of data and their relashionships-represenations, respectively, are stored. Schemas and frames are what might be defined with objects, the object classes, the object class inheritances, user-defined inheritance relations, and specific restrictions on the object, class, or frame slot types and behaviors to design analytics interfaces.

Example Scheme: Intelligent Forecasting

IS-A Stock Forecasting Technique
Portfolios Stock, bonds, corporate assets
Member Management Science Techniques

References

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. 1999c, “Infinitary Multiplayer Games,” Summer Logic Colloquium, Utrecht, August.

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

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

Gale, D. and F.M. Stewart,1953 "Infinite Games with Perfect Information," in Contributions to the Theory of Games, Annals of Mathematical Studies, vol. 28, Princeton. Nourani,C.F. 1995" Free Proof Trees and Model-theoretic Planning, February 23,1995, Automated Reasoning AISB, England, April 1995.

Nourani,C.F. 2003,’KR, Data Modeling, DB and KB Predictive Scheme Completion,” A version published at International Systems and Cybernetics, Florida, July 2003.

Nourani,C.F.1996,"Slalom Tree Computing," 1994, AI Communications, December Nourani, C.F. 2008, Business Planning and Cross Organizational Models (2008), http://mc.informatik.uni - hamburg.de/konferenzbaende/mc2005/workshops/WS4_B2.pdf

Berenbrink, P and Schulte, O: Evolutionary equilibrium in Bayesian routing games: Specialization and niche formation.TCS. 411(7-9): 1054-1074 (2010).

O. Schulte and J. Delgrande (2004), Representing von Neumann-Morgenstern Games in the Situation Calculus. Annals of Mathematics and Artificial Intelligence 42 (1-3): 73-101. (Special Issue on Multiagent Systems and Computational Logic). A shorter version of this paper appeared in the 2002 AAAI, Workshop on Decision and Game Theory.

O. Schulte (2003). Iterated Backward Inference: An Algorithm for Proper Rationalizability. Proceedings of TARK IX (Theoretical Aspects of Reasoning About Knowledge), Bloomington, Indiana, pp. 15--28. ACM, New York. Expanded version with full proofs.

N. Cressie and G.M. Laslett (1987) Random set theory and problems of modeling. SIAM Review (29) : 557-574.

Nourani,C.F.,”Descriptive Definability: A Model Computing Perspective on the Tableaux.”,CSIT 2000.: Ufa, Russia,msu.jurinfor.ru/CSIT2000.

Nourani,C.F. 2009, A Descriptive Computing, Information Forum, Liepzig, Germany, March 2009. SIWN2009 Program, Liepzig, Germany, March 2009.SIWN2009 Program. The Foresight Academy of Technology Press International Transactions on Systems Science and Applications, Vol. 5, No. 1, June 2009, pp. 60-69 .

D.G. Kendall (1974) Foundations of a theory of random sets. In Stochastic Geometry (E.F. Harding and D.G. Kendall, editors) : 322-376. J.Wiley, New York.

L.A. Zadeh (1978) Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets and Systems (1) : 3-28. Internal references.

H.T. Nguyen (2006). An Introduction to Random Sets. Chapman and Hall/CRC Press, Boca Raton, Florida. Martin-Löf. "The Definition of Random Sequences." Information and Control, 9(6): 602-619, 1966. Wolfgang Merkle and Nenad Mihailović, On the Construction of Effectively Random Sets The Journal of Symbolic Logic, Vol. 69, No. 3 (Sep., 2004), pp. 862-878. Published by: Association for Symbolic Logic.

N. Cressie and G.M. Laslett (1987) Random set theory and problems of modeling. SIAM Review (29) : 557-574.

Nourani,C.F. 2009, Competitive Models, Game Tree Degrees, and Projective Geometry on Random Sets - A Preliminary, ASL, Wisconsin.
Nourani,C.F. and O.Schulte, “Competitive Models, Descriptive Computing, and Nash Games,” by AMS, Oxford, Mississipi, April 2013. Reference: 1087-91-76.
Nourani, Cyurs F and Oliver Schulte "Multiagent Decision Trees, Competitive Models, and Goal Satisfiability,,"DICTAP, Ostrava, Czeck Republic, July 2013..
Cyrus F Nourani*, Johannes Fähndrich+, A Formal Approach to Agent planning with inference trees, NAEC, Treiste, August 2015.
Cyrus Nourani & Johannes Fähndrich, Decision Trees, Competitive Models, and Big Data Heuristics: DAI TU Berlin, September 2016: Written for Virtual Business Proceedings, Slovakia, Thompson Publishers. Heuristics, December 2016, Slovakia
Cyrus F Nourani, and Oliver Schulte, 2014.Multiplayer Games, Competitive Models, Descriptive Computing submitted for the EUROPMENT EUROPMEN, Prague, Czech Republic, April 2014. Computers, Automatic Control, Signal Processing and Systems Science, ISBN: 978-1-61804-233-0.
Nourani, C.F. “Planning and Plausible Reasoning in Artificial Intelligence, Diagrams, Planning, and Reasoning”, Proc. Scandinavian Conference on Artificial Intelligence, Denmark, May 1991, IOS Press.
Cyrus F. Nourani*, Codrina Lauth**, Rasmus Uslev Pedersen Merging Process-Oriented Ontologies with Cognitive Agent, Planning and Attention Spanning, Open Journal of Business Model Innovations, April 2015.
Nourani, C.F. 1999a, “Agent Computing, KB For Intelligent Forecasting, and Model Discovery for Knowledge Management,” in the Proceedings *AAAI- Workshop on Agent Based Systems in the Business Context*, Orlando, Florida, July 18-July 22,1999, AAAI Press.
Wilkins,D.,“Domain Independent Planning: Representation and Plan Generation,” AI, 22(3): 269-301, 1984.