

AGENT LANGUAGES, VISUAL VIRTUAL TREES, AND MODELS

Cyrus F. Nourani

Simon Fraser University – SFU

8888, University Drive, Burnaby, B.C. CANADA, V5A 1S6

180, Stuart Street, Suite, 190762, SF, CA 94119, USA

Abstract: Linguistics knowledge representation and its relation to context abstraction are presented in brief. Nourani (e.g. Nourani 1999a) has put forth new visual computing techniques for intelligent multimedia context abstraction with linguistics components. In the present paper we also instantiate proof tree leaves with free Skolemized trees. Thus virtual trees, at times like intelligent trees, are substituted for the leaves. By a virtual tree we mean a term made up of constant symbols and named but not always prespecified Skolem function terms. In virtual planning with generic diagrams that part of the plan that involves free Skolemized trees is carried along with the proof tree for a plan goal. We can apply predictive model diagrams to compute queries and discover data knowledge from observed data and visual object images keyed with diagram functions. Model-based computing can be applied to automated data and knowledge engineering with keyed diagrams. Specific computations can be carried out with predictive model diagrams. For cognition, planning, and learning the robot's mind, a diagram grid can define state. The starting space applicable project was meant for an autonomous robots space journeys.

Key Words: context and proof abstraction, explanation-based generalization, model discovery virtual tree planning topics: context-aware explanation vs. explanation-aware context, explanations and learning, visualising explanations, intelligent syntax abstraction, Morph gentzen, Language categories, agent linguistics, meta-contextual reasoning

1. Introduction

A computational linguistics with intelligent syntax and model theory is defined by [1,4]. Intelligent functions can represent agent functions, as artificial intelligence agents, or represent languages with definitions known at syntax. Since the languages represented by the agent functions can have arbitrary grammars not known to the signatures defined amongst the agent set, nondeterministic syntax computing is definable by the present programming linguistics theory. This area is to be explored in [4]. Form and context are definable by viewing computational linguistics by agent function sets. An agent FI might be as abstract as Functors defining functions and context with respect to a set and a linguistics model. To address the issues raised the role of context in KR and NL systems, particularly in the process of reasoning is related to diagram functions defining relevant world knowledge for a particular context. The relevant world functions can transfer the axioms and relevant sentences for reasoning for a context. Further, by passing context around trees via intelligent syntax trees (Section 3 the locality burden is lifted from the deductive viewpoint. A formal computable theory can be defined based on the functions defining computable models for a context and the functions carrying context around. For the VAS context foundations it is indeed possible to decrease the computational complexity of a formal system by the means of introducing context? Context localizes relevant worlds and specific computable functions define the world. Thus extraneous deductions are instant credits reducing complexity. The what is context question is reviewed in Section 3.3. from Section 4 on we explore relations between contexts. Decontextualization is possible and might be necessary to address structural deductions. It might further be implied by paraconsistent logics[24]. Section 3.3 defines meta-contextual reasoning and a brief view to defining inter-context relations. Virtual tree computing is presented as a partial evaluation technique. Morph Gentzen computing provides the basis to type inferencing with visual object computing and rapid prototyping.

2. Intelligent Languages

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 [1]. A set of function symbols in the language, referred to by Agent Function Set, is a set of function symbols modeled in the computing world by Agents. The objects, messages passing actions, and implementing agents are defined by syntactic constructs, with agents appearing as functions, expressed by an abstract language that is capable of specifying modules, agents, and their communications. We have to put this together with syntactic constructs that could run on the tree computing

theories. Sentential logic is the standard formal language applied when defining basic models. The language 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 for is a structure with a set A . There are structures defined for 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 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 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. Since the function symbols appearing might be invented by an activated agent without being defined in advance, intelligent Syntax allows us to program with nondeterministic syntax. The parsing problems are quite challenging. Trees connect by message sequences hence carry parsing sequences with them. Thus the present computational linguistics theory is a start to programming with VAS and Nondeterministic Syntax.

2.1. Intelligent Context Free Parsing

We define intelligent context free grammars as follows.

Definition 2.1. A language L is intelligent context free, abbreviated by ICF, iff L is intelligent and there is a context free grammar defining L .

A preliminary parsing theory might be defined once we observe the correspondence between String Functions and context. Let us define string intelligent functions.

Definition 2.2. A language is String Intelligent iff it is an intelligent language and all agent functions in the language are string functions. The following start to

the ICF theory is reported at [22].

Theorem 1. *String Intelligent languages are Context Free ICF.*

Proof. Follows from Definition 2.1 and abstract syntax on signature trees as presented at ADJ 1973. \square

3. Functorial Linguistic Abstraction

3.1. Categorical Grammars

Defining a category for languages allows us to define lifts, for example, from context. The linguistics abstraction techniques proposed allows us to lift from context to structures for analogical reasoning and proofs with free proof trees [12]. For example, the G-diagrams for models technique is applied at two levels for reasoning at meta-context. Models definable with G-diagrams allow free proof trees to be defined for meta-contextual reasoning with intelligent trees. The diagrams further define $D \langle A, G \rangle$ categorical abstractions are defined for lifting from diagrams to categories for definable models. A third application for G-diagrams is for encoding situations- thus abstracting from Possible Worlds Context. Categorical grammar is as close as computational linguistics has come to what we might want to refer to by Linguistics Abstraction. The term categorical, however, is not quite in the same sense as categories in the present paper. There are recent techniques for structurally transforming abstract syntax by applying logical rules, for example functional composition, and abstraction. They are called categorical grammars [13] and the Lambek calculus for grammars [14]. These techniques have formed a basis for defining natural deduction like rules for grammars and proof techniques for abstract syntax trees by Koenig [14]. There are a number of references to the present author in the paper due to him having put forth the present area for computational logic to show where it has been thus far.

3.2. Functors and VAS Syntax

Intelligent functions can represent agent functions, as in artificial intelligence agents, or represent languages with only abstract definition known at syntax. For example, a function F_i can be an agent corresponding to a language L_i . L_i can in turn involve agent functions amongst its vocabulary. Thus context might be defined at L_i with its string and splurge functions. An agent F_i might be as abstract as a functor defining functions and context with respect to a set and a linguistics model as we have defined in [4]. Since the languages represented by the agent functions can have arbitrary grammars not known to the signatures defined amongst the agent set,

nondeterministic syntax computing is definable by the present linguistics theory. This area is explored in [4]. Form and context are definable by viewing computational linguistics by agent function sets.

3.3. Models and Syntax

In [1,7] the author presented computing with intelligent trees and objects, where intelligent tree rewriting as a formal algebraic and model-theoretic computing technique might be defined from the abstract syntax trees and language constructs. The generalized diagrams were defined by this author to encode the model-theoretic semantics of a language from its abstract syntax. We present language designs with linguistics constructs that make it easier to identify G-diagram models and define automatic implementations from abstract syntax [4,10]. There is a theory in principle for building models from syntax for first order logic. However, 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 the linguistics abstraction techniques proposed allows us to lift from context to structures for analogical reasoning and proofs with free proof trees [12]. For example, the G-diagrams for models technique is applied at two levels for reasoning at meta-context. Models definable with G-diagrams allow free proof trees to be defined for meta-contextual reasoning with intelligent trees. The diagrams further define $D \langle A, G \rangle$ categorical abstractions are defined for lifting from diagrams to categories for definable models. A third application for G-diagrams is for encoding situations- thus abstracting from Possible Worlds Context. Categorical grammar is as close as computational linguistics has come to what we might want to refer to by Linguistics Abstraction. The term categorical, however, is not quite in the same sense as categories in the present paper. There are recent techniques for structurally transforming abstract syntax by applying logical rules, for example functional composition, and abstraction. They are called categorical grammars [13] and the Lambek calculus for grammars [14]. These techniques have formed a basis for defining natural deduction like rules for grammars and proof techniques for abstract syntax trees by Koenig[10]. There are a number of references to the present author in the paper due to him having put forth the present area for computational logic only to show where it has been thus far. In [1,7] we have presented computing with intelligent trees and objects, where intelligent tree rewriting as a formal algebraic and model-theoretic computing technique might be defined from the abstract syntax trees and language constructs. The generalized diagrams were defined by this author to encode the model-theoretic semantics of a language from its abstract syntax. We present language designs with linguistics constructs that make it easier to identify G-diagram models and define automatic implementations from abstract syntax [4,10]. There is a theory in principle for building models from

syntax for first order logic. However, 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.

4. Meta-Contextual Reasoning

4.1. What is Context

Context an inherent characteristic of natural language that ultimately decides the formal power of natural language?

What Theorem 1.3, the ICF theorem appears to imply, in view of Section 3.1 and 3.2 is that the latter assertion is not really true. The abstract linguistics put forth by the present paper and [4,23] have surprising implications. Utterance rich with abstractions, metaphores and string intelligent functions, i.e., functions and functors transcending context, is definable by a context free grammar. An important technical point is that the agents are represented by function names that appear on the free syntax trees of implementing trees [4]. The trees defined by the present approach have function names corresponding to computing agents. The computing agent functions have a specified module defining their functionality. The definitions are either applied at proof trees or syntax tree implementation time. Computing with intelligent trees [1], G-diagrams for their models, and $D\langle A, G \rangle$ categories are introduced in [6] for meta- contextualreasoning. Meta-contextual reasoning is defined by lifting from syntax and clausal theories to proof theory with G-diagrams for intelligent trees $D\langle A, G \rangle$ categories- categories fro models definable by G-diagrams. Proof abstraction [5] and planning with free proof trees [8] are another technique for meta-contextual reasoning. Relations between contexts can be defined by what context relevant functions are applied as to the context they correspond to and the context in which they appear. Intelligent signature functions transferring context around also define inter- context relations. A computer system can automatically infer the relation between some given set of contexts from the inter-context relevant functions.

4.2. KR, $D\langle A, G \rangle$ Models, and Context

Defining a category from the generalized diagram below is a second order lift from context.

Definition 4.1 Let M be a structure for a language L , call a subset X of M a generating set for M if no proper substructure of M contains X , i.e. if M is the closure of $X \cup c(M)$: c is a constant symbol of L . An assignment of constants to M is a pair $\langle A, G \rangle$, where A is an infinite set of constant symbols in L and G : A

M , such that $G(a)$: a in A is a set of generators for M . Interpreting a by $g(a)$, every element of M is denoted by at least one closed term of $L(A)$. For a fixed assignment $\langle A, G \rangle$ of constants to M , the diagram of M , $D\langle A, G \rangle(M)$ is the set of basic (atomic and negated atomic) sentences of $L(A)$ true in M . (Note that $L(A)$ is L enriched with set A of constant symbols.)

Definition 4.2. A G -diagram for a structure M is a diagram $D\langle A, G \rangle$, such that the G in definition 4.1 has a proper definition by specified functions.

The G -diagram $D\langle A, G \rangle$ defines a linguistics abstraction from content from which a linguistics model might be defined for reasoning [3,5]. Abstract model theory as a second order lift is defined by a category $D_j A_j G_j$. The $D\langle A, G \rangle$ category is the category for models definable from $D\langle A, G \rangle$. A preliminary functorial model theory is defined for $D\langle A, G \rangle$ in [6]. Knowledge representation has two significant roles: to define a model for the AI world, and to provide a basis for reasoning techniques to get at implicit knowledge.. An ordinary diagram is the set of atomic and negated atomic sentences that are true in a model. Generalized diagrams are diagrams definable by a minimal set of functions such that everything else in the models closure can be inferred, by a minimal set of terms defining the model. Thus providing a minimal characterization of models, and a minimal set of atomic sentences on which all other atomic sentences depend. Our primary focus will be the relations amongst KR, AI worlds, and the computability of models. To keep the models which need to be considered small and to keep a problem tractable, such that the models could be computable and within our reach, are important goals [7,18,16]. We show that we can apply G -diagram functions to localize reasoning to the worlds affected by some relevant functions to a specific reasoning aspect. To prove Godel's completeness theorem, Henkin defined a model directly from the syntax of the given theory. This structure is obtained by putting terms that are provably equal into equivalence classes, then defining a free structure on the equivalence classes. The reasoning enterprise requires more general techniques of model construction and extension, since it has to accommodate dynamically changing world descriptions and theories. The minimal set of function symbols is the functions set with which a model can inductively be defined.

5. Generalized Diagrams and Relevant Worlds

5.1. Generalized Diagrams

In order to point out the use of the generalized method of diagrams we present a brief view of the problem of planning within the present formulation. The diagram of a structure in the standard model-theoretic sense is the set of atomic and negated atomic sentences that are true in that structure [6]. The generalized diagram (G -

diagram) (author since 1987s) is a diagram in which the elements of the structure are all represented by a minimal family of function symbols and constants. Hence it is sufficient to define the truth of formulas only for the terms generated by the minimal family 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 functionsymbols. 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. Hence a diagram can be considered as a basis for a defining 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 Consider the primitive first order language (FOL) $L = c, f(X), p(X), q(X)$. Let us apply Prolog notation convention for constants and variables) and the simple theory for all X : $p(X) \ q(X), p(c)$, and indicate what is meant by the various notions.

[model] = $\{p(c), q(c), q(f(c)), q(f(f(c))), \dots, \{p(c) \ \& \ q(c), p(c) \ \& \ p(X), p(c) \ \& \ p(f(X)), \dots\}, \{p(c) \ \vee \ p(X), p(c) \ \vee \ p(f(X)), p(c) \ - \ p(c) \dots\}$,

[diagram] = $\{\{p(c), q(c), p(c), q(f(c)), q(f(f(c))), \dots\}, \dots, q(X)\}$,
 i.e. diagram = the set of atomic formulas of a model Thus the diagram is [diagram]=
 $\{p(c), q(c), q(f(c)), q(f(f(c))), \dots, q(X)\}$.

5.2. Visual Virtual Trees

Linguistics knowledge representation and its relation to context abstraction are defined in brief. Scheller[7,23] are further applications on specific visual linguistics. Nourani, e.g. [20] has put forth new visual computing techniques fro intelligent multimedia context abstraction with linguistics components as indicated at Section 4.2. In the present paper we also instantiate proof tree leaves with free Skolemized trees. Thus virtual trees, at times as intelligent trees, are substituted for the leaves. By a virtual tree we intend a term made of constant symbols and named but not always prespecified Skolem function terms. In virtual planning with G-diagrams that part of the plan that involves free Skolemized trees is carried along with the proof tree for a plan goal. We can apply predictive diagram KR to compute queries and discover data knowledge from observed data and visual object images keyed with diagram functions. Model-based computing Nourani [28] which can be applied to automated data and knowledge engineering with keyed diagrams. Specific computations can be carried out with predictive diagrams[17]. For cognition, planning, and learning the robot's mind a diagram grid can define state. The starting space applicable project was meant for autonomous robots space journeys. The designs in the authors papers are ways for a robot to update its mind state based on what it encountered on its

path. What the robot believes can be defined on a diagram grid. The degree to which a robot believes something is on the grid. It can get strengthened or weakened as a function of what the robot learns as progress is brought on. Robot's Mind State- The array grid entries are pointing to things to remember and the degree the robot believes them. The grid model is a way to encode the world with the model diagram functions.

5.3. Morph Gentzen VR

The IM Morphed Computing Logic for computing for multimedia is new projects with important computing applications since the authors Nourani 1996 papers [15]. The basic principles are a mathematical logic where a (Gentzen 1943) or natural deduction (Prawitz 1972) systems is defined by taking arbitrary structures and multimedia objects coded by diagram functions. Multimedia objects are viewed as syntactic objects defined by functions, to which the deductive system is applied. Thus we define a syntactic morphing to be a technique by which multimedia objects and hybrid pictures are holographically mapped via their defining functions to a new hybrid picture. Functorial topological structures can be defined without difficulty. The deduction rules are a Gentzen system augmented by Morphing, and Trans-morphing. The logical language has function names for hybrid pictures. The MIM Morph Rule - An object defined by the functional n -tuple $\{f_1, \dots, f_n\}$ can be Morphed to an object defined by the functional n -tuple $\{h(f_1), \dots, h(f_n)\}$, provided h is a homomorphism of abstract signature structures. The MIM TransMorph Rules- A set of rules whereby combining hybrid pictures p_1, \dots, p_n defines an Event p_1, p_2, \dots, p_n with a consequent hybrid picture p . Thus the combination is an impetus event. By trans-morphing hybrid picture's corresponding functions a new hybrid picture is deduced. The techniques can be applied to arbitrary topological structures.

The languages and MIM rules are applied to algebraic structures. The deductive theory is a Gentzen system in which hybrid pictures are named by parameterized functions; augmented by the MIM morph and transform rules. The Model theory is defined from Intelligent syntax languages Nourani [1,22]. A computational logic for intelligent languages is presented in brief with a soundness and completeness theorem in [29]. The idea is to do it at abstract models syntax trees without specifics for the shapes and topologies applied. We start with L1, and further on might apply well behaved infinitely languages. A soundness and completeness theorem has been put forth Nourani [30] stating the MIM Soundness and Completeness:

Theorem 2. (see Nourani 1997) *Morph Gentzen Logic is sound and complete.*

Proof. Plain Morph Gentzen Logical Completeness has two proofs: (A) There is a conventional proof route whereby we start with the completeness theorem for ordinary Gentzen systems. From it we can add on the morph rules and carryout a proof based on what the morph rules preserve on models. Again intricate models

are designed with positive diagrams following Section 4.3.1. The Morph Rule - A structure defined by the functional n -tuple $\{f_1, \dots, f_n\}$ can be Morphed to a structures definable by the functional n -tuple $\{h(f_1), \dots, h(f_n)\}_i$, provided h is a homomorphism of abstract signature structures. The TransMorph Rules- A set of rules whereby combining structures A_1, \dots, A_n defines an Event A_1, A_2, \dots, A_n with a consequent structure B . The rules preserve Morph Gentzen completeness based on Section 4.3.1. (B) A-There is a direct proof which applies positive diagrams, and canonical models for L_{ω}^1, ω fragments as the authors' papers in mathematical logic since 1980. \square

The direct proof has to apply what the author had developed since 1980s on infintary logic fragments with what might come close to intuitionistic logic. For example, (Gödel 1932) intuitionistic propositional logic that has disjunction property and Gentzen [1935] proved the disjunction property for closed formulas of intuitionistic predicate logic. Kleene [1945, 1952] proved that intuitionistic first-order number theory and the related existence property: If $\exists x A(x)$ is a closed theorem, then for some closed term t , $A(t)$ is a theorem. Intuitionistic systems were the predecessors to Beth's tableaux, Rasiowa and Sikorski's topological models, and Kleene's recursive realizabilities. These areas are further explored on relizability and Beth tableaux in (Nourani 2000, and 2007) and onto Kripke's [1965] possible-world semantics (Nourani Scandinivina AI 1991), with respect to which intuitionistic predicate logic is complete and consistent on classical model theory. \square

6. Computing Models

Canonical Models from models to set theory had been stated for arbitrary structures as follows.

Definition 6.1. Let $(M, a)_c$ in C be defined such that M is a structure for a language L and each constant c in C has the interpretation a in M . The mapping $c \rightarrow a \in C$ is an assignment of C in M . We say that $(M, a)_c$ in C is canonical model for a presentation P on lanaguage L , iff the assignment $c \rightarrow a$ maps C onto M , i.e. $M = (a : c \text{ in } C)$.

Generic diagrams allow us to define virtual tree canonical models with specific functions. For free Skolemization what comes to mind is generic model expansion. It is relevant to our methods of modeling, planning, and reasoning. A virtual tree, or virtual prrof tree, is a proof tree that is constructed with agent languages with free Skolem functions.

Definition 6.2. A generic diagram for a structure M is a diagram such that the there is a proper diagram definition with specific function symbols, for example $\Sigma \subset 1$ Skolem functions.

Theorem 3. *For the virtual proof trees defined for a goal formula from the G-diagram there is an initial model satisfying the goal formulas. It is the initial model definable by the G-diagram.*

Proof. In planning with GF-diagrams plan trees involving 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 a model in which the goals are satisfied. There is analogy to SLD proofs. We can view on the one hand, SLD resolution type proofs on ground terms, where we go from $p(0)$ to $p(f(c))$; or from $p(f(c))$ to $p(f(g(c)))$. Whereas, while doing proofs with free Skolemized trees we are facing proofs of the form $p(g(\dots))$ proves $p(f(g(\dots)))$ and generalizations to $p(f(x))$ proves For all x , $p(f(x))$. Since the proof trees are either proving plan goals for formulas defined on the GF-diagram, or are computing with Skolem functions defining the GF-diagram, the model defined by the GF-diagram applies and it is initial for the proofs. \square

For free Skolemization what comes to our logician mind's side of things is Generic model expansion. It is relevant to our methods of modeling, planning, and reasoning. A free proof tree is a proof tree that is constructed with free Skolem functions and from GF-diagram. The idea is that if the free proof tree is constructed for a goal formula, the GF-diagram defines a model satisfying the goal formula satisfied. The model is the initial model of the AI world for which the free Skolemized trees were constructed. Thus we have transformed the model theoretic problems of computing to that of defining computing with Generalized Diagrams. There are recent papers since 1992 [ADJ 73, Nourani Hoppe 94], where we have presented this theory and technique. Partial deductions in the present approach correspond to proof trees ,Nourani 2003 [16],that have free Skolemized trees in their representation. These concepts will be developed in forthcoming papers of this author. In the present approach, the free proof tree technique, as we shall further define, leaves could be virtual, where virtual leaves are free Skolemized trees. The idea is that if the free proof tree is constructed for a goal formula, the G-diagram defines a model satisfying the goal formula satisfied. The model is the initial model of the AI world for which the free Skolemized trees were constructed. For plans with free Skolemized trees we can apply the Hilbert epsilon technique to define computing models. What applying Hilbert's epsilon implies is that there is a model M for the set of formulas such that we can take an existentially quantified formula $w[X]$ and have it instantiated by a Skolem function that can answer the membership question to the model. Whether or not Hilbert had intended it this way or not is not relevant at present. The issue in our approach, however, is that we are not so much concerned with existentially quantified formulas. We start with some Skolem functions to define Initial models. Thus we have Hilbert models for which the Skolem functions implicitly define membership to the set defining a model. We have planning applications with VR in which there

is goal formula to be satisfied with perhaps existential quantifiers. Since we are interested in model theoretic techniques for handling proofs with the method of free proof trees we propose the following model-theoretic view, which we refer to by the Hilbert Model Theorem for Skolemized virtual tree computing.

Theorem 4. *The Hilbert's epsilon technique implies there is a model M for the set of formulas such that we can take an existentially quantified formula $w[X]$ and have it instantiated by a Skolem function which can answer the satisfiability question for the model.*

Proof. We start with some Skolem functions to define the models. The equivalence defined by Hilbert's epsilon function is instantiated by the Skolem functions defining GF-diagrams. Thus we have Hilbert models for which the Skolem functions implicitly define membership to the set defining a model. While doing proofs with free Skolemized trees we are facing proofs of the form $p(g(\dots))$ proves $p(f(g(\dots)))$ and generalizations to $p(f(x))$ proves For all x , $p(f(x))$. Since the proof trees are either proving plan goals for formulas defined on the G-diagram, or are computing with Skolem functions defining the GF-diagram, the model defined by the GF-diagram applies for the proofs. \square

7. Prediction and Discovery

Minimal prediction is an artificial intelligence technique defined since the authors model-theoretic planning project. It is a cumulative nonmonotonic approximation (Nourani 1999c) 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 $f: q \rightarrow 0,1$ is defined, or there exists a formula p in $D(M)$, such that $T \cup p$ 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 f_1, \dots, f_n 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 g_1, \dots, g_l that appear at free trees. The f terms and g terms, tree congruences, and predictive diagrams then characterize partial deduction with free trees. The predictive diagrams are applied to discover models to the intelligent game trees. Prediction is applied to plan goal satisfiability and can be combined with plausibility (Nourani 1991) probabilities, and fuzzy logic to obtain, for example, confidence intervals.

Let us see what predictive diagrams do for knowledge discovery knowledge management. Diagrams allow us to model-theoretically characterize incomplete KR. To

key into the incomplete knowledge base. Generalized predictive diagrams whereby specified diagram functions a search engine can select onto localized data fields. 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 f_1, \dots, f_n that inductively define the model. The functions are keyed onto the inference and knowledge base to select via the areas keyed to, designated as Sis in figure 4. Visual object views to active databases might be designed with the above. The trees defined by the notion of provability implied by the definition might consist of some extra Skolem functions g_1, \dots, g_n , that appear at free trees. The f terms and g terms, tree congruences, and predictive diagrams then characterize deduction with virtual trees as intelligent predictive interfaces. Data discovery from KR on diagrams might be viewed as satisfying a goal by getting at relevant data which instantiates a goal. The goal formula states what relevant data is sought. We have presented planning techniques, which can be applied to implement discovery planning. In planning with G-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 a model in which the goals are satisfied. The model is the initial model of the AI world for which the free Skolemized trees were constructed. Partial deductions in this approach correspond to proof trees that have free Skolemized trees in their representation. While doing proofs with free Skolemized trees we are facing proofs of the form $p(g(\dots))$ proves $p(f(g(\dots)))$ and generalizations to $p(f(x))$ proves for all x , $p(f(x))$. Thus the free proofs are in some sense an abstract counterpart of the SLD. Logically and theoretically the grid is minimally defined by the G-diagram functions. The grid model is a way to encode diagrammatic reasoning[21]. We have applied predictive diagrams since 1994 to partial deduction and cumulative defaults. An predictive diagram [18,20] 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 $f: q \rightarrow 0,1$ is defined, or there exists a formula p in $D[M]$, such that $T \cup p$ proves q ; or that T proves q by minimal prediction.

7.1. New Applications on Incomplete Model Diagram

According to Poole an explanation problem can be stated formally as: given Facts Φ consistent formulas, known to be true, Defaults Δ : possible hypotheses, that we accept as part of an explanation and Observations G : which are to be explained, An observation g in G is explainable, if there exists ground hypotheses $\Omega\Delta$, such that:

1. $\Phi, \cup\Omega \models g$ and
2. $\Phi, \cup\Omega$ is consistent.

Examples base set applications: closed-world assumption (CWA) and nonmonotonic logics. Base set Δ . $T(\Delta)$ is the cosure of T under logical entailment. Sup-

pose Δ includes the following: Thing (Tweety); Bird(x) \rightarrow Thing(x); Ostrich(x) \rightarrow Bird(x) Flying-Ostrich(x) \rightarrow Ostrich(x).

For example, if we want to state that nothing except for birds can fly, that is all birds, except ostriches fly and that no ostriches, except for flying ostriches can fly, we can do that with the following.

Thing(x) & not Bird(x) \rightarrow not flies(x) Bird(x) & not ostrich(x) \rightarrow Fliex(x)
 Ostrich(x) & not flying-ostrich(x) \rightarrow not Flies(x) Flying-ostrich (x) \rightarrow Flies(x)
 Thing(x) & not Bird(x) & Ostrich(x) & not Flying-ostrich not (x) Bid(x) flying-ostrich(x) \rightarrow ostrich(x).

Theorist is a proof procedure for predictive reasoning based on Loveland MESON system that Poole has further developed. The author has carried that area on virtual trees and partial deduction since 1992 (Nourani-Hoppe 1995, for example). Minimal prediction on virtual tress constrcuts hypotheses on generic Skolemized trees, where each hypothesis is a set of atomic literals pi; such that when some particular theory T is augmented with pi, entails the set of goal litreals G, where entailments are carried on with virtual trees that are modeled on generic model diagrams. Therefore, $T \models pi \cup G$. pi must be a subset of a set of ground atomic virtual tree predictibles A. In addition we must ensure $T \cup p \subset i$ is consistent under \models . The set of all possible hypotheses is $\Delta = \delta \subset i$. More general predictions are defined by applying more complex virtual tree parametrized formulas as predictibles based on generic model diagrams. First order existential formuals can form the predictibles.

To state specific applications let us state the compactness theorem from standard logic and model theory. Let us say that a predition is minimal iff the predictive hypothesis are based on a minimal subsets to Δ . Flying-Ostrich(x) \rightarrow Ostrich(x) For example, if we want to state that nothing except for birds can fly, that is all birds, except ostriches fly and that no ostriches, except for flying ostriches can fly, we can do that with the following. flies(x) Fliex(x) Flies(x) Flies(x) Theorist is a proof procedure for predictive reasoning based on Loveland MESON [18] system that Poole has further developed. The author has carried that area on virtual trees and partial deduction since 1992 (Nourani-Hoppe 1995, for example). Minimal prediction on virtual tress constrcuts hypotheses on generic Skolemized trees, where each hypothesis is a set of atomic literals i; such that when some particular theory T is augmented with $\Delta \subset i$, it entails the set of goal litreals G, where entailments are carried on with virtual trees that are modeled on generic model diagrams. Therefore, $T \models |Delta \subset i \cup G$. $\Delta \subset i$ must be a subset of a set of ground atomic virtual tree predictibles A. In addition we must ensure $T \cup \Delta \subset i$ is consistent under \models . The set of all possible hypotheses is $\Delta = \Delta \subset i$. More general predictions are defined by applying more complex virtual tree parametrized formulas as predictibles based on generic model diagrams. First order existential formuals can form the predictibles. To state specfic applications let us state the compactness theorem from standard logic and model theory. Let us say that a predition is minimal iff the predictive hypothesis are based on a minimal subsets to Δ . 1. $T \cup \Omega \models g$ 2. $T \cup \Omega$ is consistent.

Therefore, either there is a function $f : g \rightarrow 1,0$ defined direct; or by applying Godel's completeness theorem a time or two, there is a proof for g either direct or by minimal prediction. Therefore, there is an predictive diagram for the logical consequences to G .

Theorem 5. *Compactness A formula A in a theory T is valid iff it is valid in some finitely axiomatized part of T .*

Corollary 1. *A theory T has a model iff every finitely axiomatized part of T has a model.*

Theorem 6. *A set of first order observations G is explainable iff there exists an predictive diagram for the logical consequences to G .*

Proof. Applies the above with Godel's completeness and the compactness theorem from model theory.

(\rightarrow). The diagram $D[M]$, where M is a model for T , and for any formula q in M , indicates either the function $f : q \rightarrow 0,1$ is defined, or there exists a formula p in $D[M]$, such that $T \cup pg$; or that T proves g by minimal prediction. Hence for a formula g there exists ground hypotheses $D \subset \text{et}\Delta$ on which $f : g \rightarrow 1,0$ is defined such that $T \cup pg$; or there exists a formula p such that $T \cup p$ proves g , or T proves g by minimal prediction, hence the compactness theorem above renders $T \cup g$ consistent. Thus g in G explainable.

(\leftarrow) When $g \cup p$ is explainable, there exists ground hypotheses $\Delta \subset i \subset \text{set}\Delta$, such that: 1. $T \cup \Delta \subset I \models g$. 2. $T \cup \Delta$ is consistent. Therefore, either there is a function $f : g \rightarrow 1,0$ defined direct; or by applying Godel's completeness theorem a time or two, there is a proof for g either direct or by minimal prediction. Therefore, there is an predictive diagram for the logical consequences to G . \square

Acknowledgments

Acknowledgement of support (Frau Ina Lauth, Lauth transmedia GmbH, Munich, colleague Prof. Oliver Schulte, SFU, Burnaby).

References

- [1] Nourani, C.F., Slalom Tree Computing, AI Communications, The European AI Journal, December 1996, IOS Press, Amsterdam.
- [2] Genesereth, M.R. and N.J. Nilsson, Logical Foundations of Artificial Intelligence, Morgan-Kaufmann, 1987.

- [3] Nourani, C.F, Automatic Models From Syntax, 1993, Proc. XV Scandinavian Linguistics Conference, Oslo, Norway.
- [4] Functorial Model Theory and Infinite Language Categories, ASL, San Francisco, January 1995(see ASL Bulletin 1996, Wisconsin Meeting).
- [5] Mosteic, I and C. Holsbaur, Extending Explanation Based Generalization by Abstraction Operators, Machine Learning EWSL- 91, Springer-Verlag, LNAI, vol.482
- [6] Nourani, C.F., Linguistics Abstraction, April 1995, Brief Overview, ICML96, May 1996, Catalunya, Tarragona, Spain.
- [7] Scheler, G., Some Parallels between Visual Linguistics and Linguistics Processing, Proc. 18th Annual Conference of the Cognitive Science Society, La Jolla, July 1997.
- [8] Nourani, C.F.Planning and Plausible Reasoning in AI, Proc. Scandinavian Conference in AI, May 1991, Denmark, 150- 157, IOS Press.
- [9] Nourani, C.F., Diagrams, Possible Worlds, and the Problem of Reasoning in Artificial Intelligence, Logic Colloquium, Padova, Italy, 1988.
- [10] Koenig, E., Parsing Categorical Grammars, Report 1.2.C, Espirit Basic Research Action 3175, Dynamic Interpretation of Natural Language (DYANA), 1990.
- [11] Nourani, C.F. 2004 Versatile Abstract Syntax, MetaContextual Logic and Virtual Reality Computing Logic, the third international conference on Information in Tokyo, November 29 - December 2, 2004. <http://www.information-iii.org/conf/info2004.html>.
- [12] 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.
- [13] - Goguen, J.A. J.W Thatcher, E.G. Wagner, and J.B. Wright, An Introduction to Categories, Algebraic Theories and Algebras, IBM Research Report, RC5369, Yorktown Heights, N.Y., April 1975.
- [14] Lambek, J. The Mathematics of Sentence Structure, American Mathematical Monthly, 65, 154-170.
- [15] Nourani, C.F., MIM Logik, 1977, Summer Logic Colloquium, Prague, August 1998.

- [16] Nourani, C.F 2003 KR, Predictive Model Discovery, and Schema Completion 7th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2002) Orlando, USA, July 14-18, 2003
- [17] Nourani, C.F., Free Proof Trees and Model-theoretic Planning February 23, 1995, Automated Reasoning AISB, Sheffield, England, April 1995.
- [18] , Loveland D.W., Automated Theorem Proving: A Logical Basis. North Holland 1978.
- [19] Poole, D, R. Goebel, and R. Aleliunas. Theorist: A Logical Reasoning System For Defaults and Diagnosis. In N. Cercone and G. McCalla editors, The knowledge Frontier: Essays in the represneation of Knowledge. Springer-Verlag 1987.
- [20] Nourani, C.F., Intelligent Multimedia- New Computing Techniques and Its Applications, Ferbruary 28, 1997. CSIT'99, Proceedings of 1st International Workshop on Computer Science and Information Technologies, January 18-22, 1999, Moscow, Russia. Ch. Freytag and V. Wolfengagen (Eds.): MEPhI Publishing 1999, ISBN 5-7262-0263-5.
- [21] Nourani, C.F., Compuability, KR, and Reducibiltiy, ASL, Toronto, May 1998. BSL, vol. 4, no. 4, December 1998, page 457.
- [22] Nourani, C.F. (1995d), Intelligent Languages- A Preliminary Syntactic Theory, May 15, 1995, Mathematical Foundations of Computer Science;1998, 23rd International Symposium, Brno, Czech Republic, August The satellite workshop on Grammar systems. Silesian University, Faculty of Philosophy and Sciences, Institute of Computer Science.
- [23] Scheler, G, Feature-based Perception of Semantic Concepts to appear in Freksa(ed.) Computation and Cognition. Springer, 1997.
- [24] Nourani, C.F., Functorial Syntax and Paraconsistent Logics, ASL, New Orleans, May 1999.
- [25] Chomsky, N. (1965). Aspects of the theory of syntax. MIT Press. Chomsky, N. (1988) Language and problems of knowledge. MIT Press.
- [26] Chomsky, N. The Minialist Program, MIT Press, 1996.
- [27] Nourani, C.F., Syntax Trees, Intensional Models, and Modal Diagrams For Natural Language Models, Revised July 1997. Uppsala Logic Colloquium, August1998, Uppsala University, Sweden.

- [28] Nourani, C.F., Intelligent Trees, Thought Models, And Intelligent Discovery, MODEL-BASED REASONING IN SCIENTIFIC DISCOVERY (MBR'98), Pavia, Italy, December 17-19, 1998.
- [29] G, Beweisbarkeit und Unbewiesbarkeit von Anfangsfallen der transfiniten Induktion in der reinen Zahlentheorie, Math Ann 119, 140-161, 1943.
- [30] Nourani, C.F., 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.
- [31] Nourani, C.F. and T. Hoppe, 1995, G-Diagrams For Models, Free Proof Trees, And Predictive Automated Reasoning, February 21, 1995, Automated Reasoning-AISB, England, April 1995.