Abstracts of Current Literature

Selected Dissertation Abstracts

Compiled by:
Susanne M. Humphrey, National Library of Medicine, Bethesda, MD 20209
Bob Krovetz, University of Massachusetts, Amherst, MA 01002

The following are citations selected by title and abstract as being related to computational linguistics or knowledge representation, resulting from a computer search, using the BRS Information Technologies retrieval service, of the Dissertation Abstracts International (DAI) database produced by University Microfilms International.

Included are the UM order number and year-month of entry into the database; author; university, degree, and, if available, number of pages; title; DAI subject category chosen by the author of the dissertation; and abstract. References are sorted first by DAI subject category and second by author. Citations denoted by an MAI reference do not yet have abstracts in the database and refer to abstracts in the published Masters Abstracts International.

Unless otherwise specified, paper or microform copies of dissertations may be ordered from University Microfilms International, Dissertation Copies, Post Office Box 1764, Ann Arbor, MI 48106; telephone for U.S. (except Michigan, Hawaii, Alaska): 1-800-521-3042, for Canada: 1-800-268-6090. Price lists and other ordering and shipping information are in the introduction to the published DAI. An alternate source for copies is sometimes provided at the end of the abstract.

Logic Programming Semantics: Techniques and Applications

Baudinet, Marianne
Stanford University Ph.D. 1989, 170 pp.
Computer Science
University Microfilms International
ADG89-19402

It is generally agreed that providing a precise formal semantics for a programming language is helpful in fully understanding the language. This is especially true in the case of logic-programming–like languages for which the underlying logic provides a well-defined but insufficient semantic basis. Indeed, in addition to the usual model-theoretic semantics of the logic, proof-theoretic deduction plays a crucial role in understanding logic programs. Moreover, for specific implementations of logic programming, e.g. PROLOG, the notion of deduction strategy is also important.

In this thesis, we provide semantics for two types of logic programming languages and develop applications of these semantics. First, we propose a semantics of PROLOG programs that we use as the basis of a proof method for termination properties of PROLOG programs. Second, we turn to the temporal logic programming language TEMPLOG of Abadi and Manna, develop its declarative semantics, and then use this semantics to prove a completeness result for a fragment of temporal logic and to study TEMPLOG's expressiveness.

In our PROLOG semantics, a program is viewed as a function mapping a goal to a finite or infinite sequence of answer substitutions. The meaning of a program is then given by the least solution of a system of functional equations associated with the program. These equations are taken as axioms in a first-order theory in which various program properties, especially termination or nontermination properties, can be proven. The method extends to PROLOG programs with extra-logical features such as cut.

For TEMPLOG, we provide two equivalent formulations of the declarative semantics: in terms of a minimal temporal Herbrand model and in terms of a least fixpoint. Using these semantics, we are able to prove that TEMPLOG is a fragment of temporal logic that admits a complete proof system. The fixpoint semantics also enables us to study TEMPLOG's expressiveness. For this, we focus on the propositional fragment of TEMPLOG. We prove that propositional TEMPLOG has essentially the expressiveness of finite automata or regular languages, and that its extension with stratified negation has the expressiveness of Buchi automata or $Ω$-regular languages.

Understanding Coreference in a System for Solving Physics Word Problems

Bulko, William Charles
The University of Texas at Austin Ph.D.
1989, 209 pp.
Computer Science, Physics, General
University Microfilms International
ADG89-20668

In this thesis, a computer program (BEATRIX) is presented that takes as input an English statement of a physics problem and a figure associated with it, understands the two kinds of input in combination, and produces a data structure containing a model of the physical objects described and the relationships between them. BEATRIX provides a mouse-based graphic interface with which the user sketches a picture and enters English sentences; meanwhile, BEATRIX creates a neutral internal representation of the picture similar to this, which might be produced as the output of a vision system. It then parses the text and the picture representation, resolves the references between objects common to the two data sources, and produces a unified model of the problem world. The correctness and completeness of this model have been validated by applying it as input to a physics problem-solving program currently under development.

Two descriptions of a world are said to be coreferent when they contain references to overlapping sets of objects. Resolving coreferences to produce a correct world model is a common task in scientific and industrial problem-solving: because English is typically not a good language for expressing spatial relationships, people in these fields frequently use diagrams to supplement textual descriptions. Elementary physics problems from college-level textbooks provide a useful and convenient domain for exploring the mechanisms of coreference.

Because flexible, opportunistic control is necessary in order to recognize coreference and to act upon it, the understanding module of BEATRIX uses a blackboard control structure. The blackboard knowledge sources serve to identify physical objects in the picture, parse the English text, and resolve coreferences between the two. We believed that BEATRIX demonstrates a control structure and collection of knowledge that successfully implements understanding of text and picture by computer. We also believe that this organization can be applied successfully to similar understanding tasks in domains other than physics problem-solving, where data such as the output from vision systems and speech understanders can be used in place of text and pictures.

Tensor Manipulation Networks: Connectionist and Symbolic Approaches to Comprehension, Learning, and Planning

Dolan, Charles Patrick
University of California, Los Angeles Ph.D.
1989, 475 pp.
Computer Science, Physiology, Philosophy
University Microfilms International
ADG89-19909

It is a controversial issue as to which of the two approaches, the Physical Symbol System Hypothesis (PSSH) or Parallel Distributed Processing (PDP), is a better characterization of the mind. At the root of this controversy are two questions: (1) what sort of computer is the brain, and (2) what sort of programs run on that computer? What is presented here is a theory that bridges the apparent gap between PSSH and PDP approaches. In particular, a computer is presented that adheres to constraints of PDP computation (a network of simple processing units), and a program is presented that at first glance is only suitable for a PSSH computer but that runs on a PDP computer. The approach presented here, vertical integration, shows how to construct PDP computers that can process symbols and how to design symbol systems so that they will run on more brainlike computers.

The type of computer presented here is called a tensor manipulation network. It is a special type of PDP network where the operation of the network is interpreted as manipulations of high rank tensors (generalized vector outer products). The operations on tensors in turn are interpreted as operations on symbol structures. A wide range of tensor manipulation architectures are presented with the goal of inducing constraints on the symbol structures that it is possible for the mind to possess.

As a demonstration of what is possible with constrained symbol structures, a program, CRAM, that uses and acquires thematic knowledge is presented. CRAM is able to read, in English, single-paragraph, fablelike stories and either give a thematically relevant summary or generate planning advice for a character in the story. CRAM is also able to learn new themes through the combination of existing, known themes encountered in the fables CRAM reads.

Cram demonstrates that even the most symbolic cognitive tasks can be accomplished with PDP networks, if the networks are designed properly.

Construction of a Sound Nonnumeric Probabilistic Reasoner

Neufeld, Eric Michael
University of Waterloo (Canada) Ph.D. 1989
Computer Science
This item is not available from University Microfilms International
ADG05-66059

The AI community has long sought a nonnumeric formalism for reasoning in the presence of uncertainty. Probability was abandoned in the beginning for several reasons: the distributions were difficult to obtain, computation was exponential in most cases, and results were considered to be counter-intuitive.

Alternatives to probability included ad hoc "reasoners," novel numeric and symbolic uncertainty formalisms, and extensions of mathematical logic. The first two alternatives had semantic difficulties. "Nonmonotonic" extensions of mathematical logic kept running into variations of the lottery paradox as a consequence of rules that accepted not quite certain conclusions as certain.

We present a framework for representation and reasoning under uncertainty that does not demand numeric probability distributions and that is not a victim of the lottery paradox. The formalism is called an inference graph. Nodes encode events, and arcs encode both probabilistic inequalities and information about statistical independence. We have implemented an efficient inference graph interpreter. If one accepts this formalism as a partial account of defaults, then this interpreter produces the expected answer for nearly every problem we have encountered in the nonmonotonic literature. Where it disagrees, one can show that the expected "answer" is wrong in a statistical sense.

Its greatest advantage is that beliefs encoded in the graph can (in principle) be verified by performing an experiment in the real world; they possess a rigorous semantics that none of the nonmonotonic formalisms can claim.

Deriving Rules for Medical Expert Systems Using Natural Language Parsing and Discourse Analysis

Rinaldo, Frank Joseph
Illinois Institute of Technology Ph.D. 1989, 114 pp.
Computer Science, Language, Linguistics
University Microfilms International
ADG89-22174

A rule knowledge base for a rule-based expert system is generated automatically by processing input text in the form of published papers. This system 'reads' the input text and using sublanguage analysis techniques performs syntactic, semantic, and discourse analysis of the text. The syntactic analysis phase uses the Linguistic String Parser developed at New York University. Some of the semantics are integrated in with the LSP. The final semantic interpretation and discourse analysis is determined by a computer program written in C. The information derived is stored internally in a framelike structure that is used to resolve ambiguity and determine the proper context of the phrase and/or sentence being analyzed. This internal structure is then traversed with the assistance of a medical knowledge base to generate production rules for the expert system.

Use of Prior Knowledge in Integration of Information from Technical Materials

Kubes, Milena
Mc Gill University (Canada) Ph.D. 1989
Education, Psychology
This item is not available from University Microfilms International
ADG05-65470

This study was designed to examine the ability to use prior knowledge in text comprehension and knowledge integration. The focus of the research was on effects of different degrees of subjects' theoretical knowledge in the domain of biochemistry on their comprehension of written technical materials describing experimental procedures and results, and the ability to integrate such new text derived information with prior theoretical knowledge considered by experts to be relevant to the topic. Effects of cues on the accessibility and use of prior knowledge were also examined. Pre-test questions testing the extent of subjects' prior knowledge of photosynthesis, and a "cue article" specifically designed to prime subjects' relevant prior knowledge of photosynthesis served as cues in the study.

A theoretical model of experts' knowledge was developed from a semantic analysis of expert-produced texts. This "expert model" was used to evaluate the extent of students' theoretical knowledge of photosynthesis and its accessibility while applying it to the experimental tasks. College students and university graduate students served as subjects in the study, permitting a contrast of groups varying in prior knowledge of and expertise in chemistry.

Statistical analyses of data obtained from coding subjects' verbal protocols against text propositions and the expert model revealed that prior knowledge and comprehension contribute significantly to predicting knowledge integration, but they are not sufficient for this process to take place. It appears that qualitative aspects and specific characteristics of subjects' knowledge structure contribute to the process of integration, not simply the amount of accumulated knowledge. There was also evidence that there are specific inferential processes unique to knowledge integration that differentiate it from test comprehension. Cues manifested their effects on performance on comprehension tasks and integrative tasks only through their interactions with other factors. Furthermore, it was found that textual complexity placed specific constraints on students' performance: the application of textual information to the integrative tasks and students' ability to build conceptual frame representations based on text propositions depended on the complexity of the textual material. (Abstract shortened with permission of author.)

A Fast Parallel Algorithm for N-ARY Unification with AI Applications

Cline, Marshall Peter
Clarkson University Ph.D. 1989, 319 pp.
Engineering, Electronics and Electrical, Computer Science
University Microfilms International
ADG89-18761

Unification is the central primitive used in all resolution-based automated theorem proving systems (Robinson 1965a) and logic programming (Kowalski 1974) environments. Almost all the efforts in this area have been focused on the special case of unifying just two terms (binary unification [Kowalski 1979]), which is only sufficient when the theorem prover's input language is restricted (ex: to Horn logic [Horn, 1951]; Henschen 1974]) or when additional inference rules are provided (such as Factoring [Wos 1964]). Fast (linear time) binary unification algorithms have existed for a decade (Martelli 1977; Paterson 1978b), but when more than two terms are unified (n-ary unification), the typical solution is quadratic, the original algorithm being exponential (Robinson 1965a).

We show that n-ary unification can be reduced to binary unification, resulting in efficient sequential and parallel algorithms. In particular, if N is the size of the input graph (nodes + edges), the parallel algorithm has a time cost of O(N) with processors proportional to the number of terms in the unified set. Adopting the popular computational complexity view that sublinear unification is impossible even with infinite processors

(Yasuura 1984; Dwork 1984), our algorithm is "popularly optimal" (Dwork 1986). It is the literature's first linear parallel n-ary unifier.

The sequential algorithm is also asymptotically efficient: it has a time cost bounded by O(N logN), which is within a factor of logN from optimal. As well as theoretical efficiency, our system has minimal startup and overhead costs, pragmatic concerns that have plagued other unification algorithms (DeChampeaux 1986a; Martelli 1982; Escalada 1988).

Asymptotic time cost analysis of unification in the recent Literature has contained ambiguities. We show how different authors have stated the same time cost for their algorithms, yet the actual efficiencies vary all the way from linear to exponential time. We present a new categorization schema intended to remove this confusion.

A flexible binary unifier is developed that is combined with an implementation of our n-ary unification theory. The n-ary unifier is ported to a number of hardware platforms, including several monoprocessors and a tightly coupled MIMD multiprocessor (the Sequent B21). A suite of tests is performed on the implementations, results confirming the time cost analysis. We also discuss the relative fitness of SIMD architectures (such as the Connection machine) for a class of algorithms such as ours.

The simplicity of our algorithm not only implies a very small startup cost, but it also means the theory might be realizable in hardware. We discuss the importance of this (the Literature's first hardware n-ary unifier), then present a solution. We keep the presentation at the conceptual level, broad enough to allow implementations to be based on any number of existing hardware primitives.

Finally, the new n-ary unification framework is shown to be flexible, promising other avenues of application.

Prosodic Constituency in the Lexicon

Inkelas, Sharon

Stanford University Ph.D. 1989, 385 pp.
Language, Linguistics
University Microfilms International
ADG89-19437

The goal of this thesis is to argue for the existence in the lexicon of a hierarchy of prosodic constituents, coextensive with the domains of lexical phonological rules. These form the lexical half of the prosodic hierarchy of constituents whose postlexical members include the phonological word, phonological phrase, and intonational phrase (Selkirk 1978). I argue in support of lexical prosodic structure, as distinct from the copresent morphological structure, by showing that mismatches occur between the two. Two cases of misalignment between rule domains and morphological structure are discussed: those compounds whose members form individual domains for rules, and invisibility effects (in which some member of the morphological string is excluded from the corresponding rule domain). Construing these phenomena as mismatches between prosodic and morphological structure provides a much more explanatory account than is possible in a framework that posits only a single structure.

A further consequence of the introduction of prosodic structure into the lexicon is the ensuing possibility that morphemes might subcategorize for attachment to prosodic constituents. Lieber (1980) characterized dependent (bound) morphemes as subcategorizing for a morphological sister. I show that lexical dependence can actually be factored into two dimensions: prosodic and morphological. By crossing these two independent dimensions we derive a four-way typology of morphemes corresponding to the recognized categories of affix, bound root, free stem, and clitic. In particular, the assignment of a prosodic subcategorization frame to clitics yields a number of correct predictions about the distribution of clitics across categories and within sentences.

Three sources of prosodic structure are posited. The most general is a mapping algorithm that parses strings into prosodic constituents on the basis of their morphological constituency. A second, more specific course is a rule of compounding that imposes a particular prosodic constituent structure on its output. The third source is prosodic subcategorization frames. These not only constrain the distribution of prosodically bound morphemes, but also contribute structure to the representation. Where these various sources of prosodic constituency make conflicting predictions, the Elsewhere Condition causes the most specific to take precedence. These three mechanisms not only generate correspondence between morphological prosodic structure, but are also capable of deriving all and only the attested mismatches between them.