

05171 Abstracts Collection
**Nonmonotonic Reasoning, Answer Set
Programming and Constraints**
— Dagstuhl Seminar —

Gerhard Brewka¹, Ilkka Niemelä², Torsten Schaub³, Mirosław Truszczyński⁴,
and Joost Vennekens⁵

¹ Univ. Leipzig, DE

brewka@informatik.uni-leipzig.de

² Helsinki Univ. of Technology, FI

³ Univ. Potsdam, DE

torsten@cs.uni-potsdam.de

⁴ Univ. of Kentucky, US

mirek@cs.engr.uky.edu

⁵ K.U. Leuven, BE

joost@cs.kuleuven.be

Abstract. From 24.04.05 to 29.04.05, the Dagstuhl Seminar 05171 “Nonmonotonic Reasoning, Answer Set Programming and Constraints” was held in the International Conference and Research Center (IBFI), Schloss Dagstuhl. During the seminar, several participants presented their current research, and ongoing work and open problems were discussed. Abstracts of the presentations given during the seminar as well as abstracts of seminar results and ideas are put together in this paper. The first section describes the seminar topics and goals in general. Links to extended abstracts or full papers are provided, if available.

Keywords. Knowledge representation, nonmonotonic reasoning, logic programming, answer-set programming, constraints

**Executive Summary: The Dagstuhl Seminar 05171 –
Nonmonotonic Reasoning, Answer Set Programming and
Constraints**

We provide a brief overview of the seminar and comment on most the important research themes that emerged.

Keywords: Knowledge representation, nonmonotonic reasoning, logic programming, answer-set programming, constraints

Joint work of: Brewka, Gerhard; Niemelä, Ilkka; Schaub, Torsten; Truszczyński, Mirosław

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2005/260>

“Do What I Meant, Not What I Said” Debugging Answer Set Programs

Martin Brain (University of Bath, GB)

This talk presents a discussion on what debugging means in the context of answer set programming. An attempt will be made to persuade the listeners that any debugging performed by a computer can be reduced to one of three fundamental questions. Algorithms to resolve these questions will be presented. Finally the speaker will (rashly) assert that the first step towards usable debugging systems is complete and outline areas of possible development.

Joint work of: Brain, Martin; De Vos, Marina

P2P Systems with Trust Relations

Loreto Bravo (Carleton Univ. - Ottawa, CDN)

The problem of answering queries posed to a peer who is a member of a peer-to-peer data exchange system is studied.

The answers have to be consistent wrt to both the local semantic constraints and the data exchange constraints with other peers; and must also respect certain trust relationships between peers.

A semantics for peer consistent answers under exchange constraints and trust relationships is introduced and some techniques for obtaining those answers, using answer sets semantics, are presented.

Keywords: Peer-to-peer systems, answer sets semantics, trust relations

Joint work of: Bravo, Loreto; Bertossi, Leopoldo

Prioritized Component Systems

Gerhard Brewka (Universität Leipzig, D)

We introduce a flexible framework to specify problem solutions (outcomes) and preferences among them. The proposal combines ideas from answer-set programming (ASP), answer-set optimization (ASO) and CP-nets. The problem domain is structured into components. ASP techniques are used to specify values of components, as well as global (inter-component) constraints among these values. ASO methods are used to describe preferences among the values of a component and CP-net techniques to represent inter-component dependencies and corresponding preferences. Both CP-nets and ASO programs are special cases of the resulting framework.

Keywords: Preference handling, answer set programming, CP-nets

Joint work of: Brewka, Gerhard; Niemelä, Ilkka; Truszczynski, Mirek

See also: Proc. AAAI-05, Philadelphia

Towards Static Analysis of Answer Set Programs

Stefania Costantini (University of L'Aquila, I)

In this talk we propose a static analysis methodology for Answer Set Programming. The method is based on: identifying the cycles contained in the program, showing that stable models of the overall program are composed of stable models of suitable sub-programs, corresponding to the cycles; defining the Cycle Graph, where each vertex corresponds to one cycle, and each edge corresponds to one handle, which is a literal containing an atom that, occurring in both cycles, actually determines a connection between them; providing a procedure for checking consistency on the cycle graph without actually computing the answer sets. Based on these results, one is able to check (in principle even on the non-grounded version of the program) several properties, among which whether a given atom may belong to some answer set.

Keywords: Answer set programming, graph representations

Extending Answer Set Programming for the Semantic Web

Thomas Eiter (TU Wien, A)

The Semantic Web as a next-generation web is a scientific challenge which, if successful, will greatly enhance our possibilities for information access and processing, making available a wealth of information. In this context, several logic-based languages are considered to provide a basis for advanced information access which involves a great deal of reasoning beyond simple pattern matching. As we feel, Answer Set Programming (ASP) has something to offer for this goal, but on the other hand, lacks in its basic design features which are desired for Semantic Web applications. Among them are higher-order predicates and the possibility to integrate external information, e.g., coming from other reasoning engines, into an answer set program. In this talk, we report on an ongoing project at TU Wien which addresses such issues, and looks into extensions of ASP which provide the basic support needed for advanced reasoning in Semantic Web applications.

Keywords: Answer set programming; semantic web; hybrid systems; description logics; higher-order predicates

Joint work of: Eiter, Thomas; Ianni, Giovambattista; Lukasiewicz, Thomas; Schindlauer, Roman; Tompits, Hans

Modularity, Magic Sets, and their Application in Data Integration

Wolfgang Faber (University of Calabria, I)

We propose a generalization of the well-known Magic Sets technique to datalog programs with (possibly unstratified) negation under stable model semantics.

Our technique produces a new program the evaluation of which is generally more efficient (due to a smaller instantiation), while preserving soundness under cautious reasoning. Importantly, if the original program is consistent, then full query-equivalence is guaranteed for both brave and cautious reasoning, which turn out to be sound and complete. This technique can be combined with a previously introduced method for disjunctive programs (without negation).

In order to formally prove the correctness of our Magic Sets transformation, we introduce a novel notion of modularity for datalog with negation under the stable model semantics, which is relevant per se. We prove that such a module can be evaluated independently from the rest of the program, while preserving soundness under cautious reasoning (and completeness under brave reasoning). For consistent programs, soundness and completeness are guaranteed for both brave and cautious reasoning.

Our Magic Sets optimization offers an effective method for enhancing the performance of data-integration and systems in which query-answering is carried out by means of cautious reasoning over datalog programs with negation (which are guaranteed to be consistent). In fact, preliminary results of experiments in the EU project INFOMIX, show that Magic Sets are fundamental for the scalability of the system.

Keywords: Magic sets, modularity, data integration

Full Paper: <http://www.wfaber.com/research/papers/icdt2005.pdf>

Answer Sets for Propositional Theories

Paolo Ferraris (Univ. of Texas at Austin, USA)

Equilibrium logic, introduced by David Pearce, extends the concept of an answer set from logic programs to arbitrary sets of formulas. Logic programs correspond to the special case in which every formula is a “rule” — an implication that has no implications in the antecedent (body) and consequent (head).

The semantics of equilibrium logic looks very different from the usual definitions of an answer set in logic programming, as it is based on Kripke models. In this paper we propose a new definition of equilibrium logic which uses the concept of a reduct, as in the standard definition of an answer set.

Second, we apply the generalized concept of an answer set to the problem of defining the semantics of aggregates in answer set programming. We propose a

semantics for programs with aggregates that includes aggregates in the style of Faber, Leone, and Pfeifer containing arbitrary formulas and also choice rules.

Keywords: Answer sets, equilibrium logic, stable models, propositional theories, aggregates, weight constraints

The `nomore++` Approach to Answer Set Solving

Martin Gebser (Universität Potsdam, D)

We present a new answer set solver called `nomore++`.

A distinguishing feature is that it treats heads and bodies equitably as computational objects.

Apart from its operational foundations, we show how it improves on previous work through its hybrid lookahead and its support-driven strategy.

We underpin our claims by selected experimental results.

Keywords: Answer set programming; normal logic programs

Joint work of: Anger, Christian; Gebser, Martin; Linke, Thomas; Neumann, Andre; Schaub, Torsten

On Conformant Planning with Answer Set Prolog

Michael Gelfond (Texas Tech University - Lubbock, USA)

I briefly describe a family of conformant planners based on logic programming and theory of action and change. Various planners from this family will be compared with other state-of-the-art conformant planners in terms of efficiency, range of applicability, and degree of elaboration tolerance.

Keywords: Planning, theories of action and change, answer set programming

Joint work of: Son, Tran Cao; Tu, Phan Huy; Gelfond, Michael; Morales, Ricardo

Answer Set Programming and Combinatorial Voting

Rafal Grabos (Universität Leipzig, D)

We show how Logic Programming with Ordered Disjunction (LPOD), the extension of answer set programming for handling preferences, may be used for representing and solving collective decision making problems. We present the notion of combinatorial vote problem in the context of LPOD and define various types of vote rules, used as decision criteria for determining optimal candidate for a group of voters.

Keywords: Decision making, answer set programming, preferences

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2005/262>

Extended Conceptual Logic Programs

Stijn Heymans (Vrije Universiteit Brussel, B)

We present extended conceptual logic programs (ECLPs), for which reasoning is decidable and, moreover, can be reduced to finite answer set programming. ECLPs are useful to reason with both ontological and rule-based knowledge, which is illustrated by simulating reasoning in an expressive description logic (DL) equipped with DL-safe rules. Furthermore, ECLPs are more expressive in the sense that they enable nonmonotonic reasoning, a desirable feature in locally closed subareas of the Semantic Web.

Keywords: Answer set programming, open domains, description logics

Joint work of: Heymans, Stijn; Van Nieuwenborgh, Davy; Vermeir, Dirk

Enhancing Answer Set Programming with Templates

Giovambattista Ianni (TU Wien, A)

The work aims at extending Answer Set Programming (ASP) with the possibility of quickly introducing new predefined constructs and to deal with compound data structures: we show how ASP can be extended with template predicate definitions. We present language syntax and give its operational semantics. We show that the theory supporting our ASP extension is sound, and that program encodings are evaluated as efficiently as ASP programs. Examples show how the extended language increases declarativity, readability, compactness of program encodings and code reusability.

Keywords: Answer set programming, modular logic programming

Joint work of: Ianni, Giovambattista; Calimeri, Francesco; Santoro, Maria Carmela; Pietramala, Adrian

Translating Normal Logic Programs into Propositional Theories

Tommi Janhunen (Helsinki University of Technology, FIN)

The goal of this research is to combine the knowledge representation capabilities of normal logic programs (NLPs) with the efficiency of modern SAT solvers. To this end, we look for polynomial and faithful but non-modular translations from normal programs into propositional theories. In addition to summarizing existing approaches in this talk, we present a new method for transforming normal logic programs into sets of clauses. This transformation is based on a novel characterization of stable models in terms of level numberings and it uses atomic normal programs, which are free of positive body atoms, as an intermediary representation. The corresponding translation function possesses a unique combination of properties: (i) a bijective relationship is established between stable models and classical models, (ii) the models coincide up to the set of atoms $hb(P)$ appearing in a program P , and (iii) the length of the translation as well as the translation time are of order $len(P) \times \log_2(|hb(P)|)$ where $len(P)$ denotes the length of P in symbols. Our preliminary experiments with an implementation of the transformation, namely translators called LP2ATOMIC and LP2SAT, and SAT solvers such as CHAFF and RELSAT suggest that our approach becomes competitive when the task is to compute not just one but all stable models for a normal program possessing sufficiently many stable models.

Keywords: Propositional logic, normal logic programs, stable models, supported models, faithful translations, modularity

Modelling and Implementing a Knowledge Base for Checking Medical Invoices with DLV

Gabriele Kern-Isberner (Universität Dortmund, D)

Checking medical invoices, done by every health insurance company, is a labor-intensive task. Both speed and quality of executing this task may be increased by the knowledge-based decision support system ACMI which we present in this paper.

As the relevant regulations also contain various default rules, ACMI's knowledge core is modelled using the answer set programming paradigm. It turned out that all relevant rules could be expressed directly in this framework, providing for a declarative and easily extendable and modifiable knowledge base.

ACMI is implemented using the DLV system.

Keywords: Answer sets, default rules, health insurance, rule schemas

Joint work of: Kern-Isberner, Gabriele; Beierle, Christoph; Dusso, Oliver

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2005/261>

Strong Equivalences for Logic Programs with Preferences

Kathrin Konczak (Universität Potsdam, D)

Recently, strong equivalence for Answer Set Programming has been studied intensively, and was shown to be beneficial for modular programming and automated optimization.

In this work we define the novel notion of strong equivalence for logic programs with preferences. Based on this definition we give, for several semantics for preference handling, necessary and sufficient conditions for programs to be strongly equivalent.

These results provide a clear picture of the relationship of these semantics with respect to strong equivalence, which differs considerably from their relationship with respect to answer sets.

Finally, based on these results, we present for the first time simplification methods for logic programs with preferences.

Keywords: Answer set programming, nonmonotonic reasoning, preferences, equivalences

Joint work of: Konczak, Kathrin; Faber, Wolfgang

A Model-Theoretic Counterpart of Loop Formulas

Joohyung Lee (Univ. of Texas at Austin, USA)

In an important recent paper, Lin and Zhao introduced the concept of a loop formula, and showed that the answer sets for a logic program are exactly the models of Clark's completion of the program that satisfy the loop formulas. Just as supported sets are a model-theoretic account of completion, "externally supported" sets we propose, are a model-theoretic counterpart of loop formulas. This reformulation of loop formulas shows that they are related to assumption sets (Sacca and Zaniolo) and to unfounded sets (Van Gelder, Ross and Schlipf; Leone, Rullo and Scarcello), invented many years earlier.

Keywords: Logic programming, answer set programming, nonmonotonic reasoning, loop formulas, completion

Full Paper: <http://www.cs.utexas.edu/apps/murf/papers/mtclf.pdf>

See also: IJCAI'05, to appear

The DLV System: Ongoing Enhancements and Applications

Nicola Leone (University of Calabria, I)

We overview the most relevant enhancements under development in the DLV system. On the optimization side, we describe relevant improvements to DLV instantiator, a novel magic-set technique for efficient query evaluation, and heuristics for solving SigmaP2-hard problems.

For extensions, we present an advanced mechanism allowing to plug-in external functions in DLV, in the form of built-in predicates. They provide the basis for extending the language with functions symbols and complex terms (like sets and lists). On the application side, we describe the advanced features for database connectivity, the novel DLV API, and the challenging DLV applications for data-integration (INFOMIX system) and for Ontology Representation and Reasoning.

Keywords: Disjunctive logic programming, answer set programming, non-monotonic reasoning, disjunctive datalog, heuristics, optimizations

Joint work of: Leone, Nicola; The DLV Team

Cmodels-3: SAT-based Disjunctive Answer Set Programming System

Yuliya Lierler (Universität Erlangen-Nürnberg, D)

Using SAT solvers as inference engines in answer set programming solvers showed to be the promising approach in building efficient systems.

Nowadays SAT based answer set programming systems successfully work with nondisjunctive programs. We propose the way to use the SAT solvers for finding answer sets for disjunctive logic programs. This work implements two different ways of SAT solver invocation used in nondisjunctive answer set programming. The algorithms are based on the definition of completion for disjunctive programs and the extension of loop formula definition to the case of disjunctive programs. We propose the necessary modifications to the algorithms known for nondisjunctive programs in order to adapt them to the case of disjunctive programs and demonstrate their implementation based on system Cmodels.

Keywords: Disjunctive answer set programming, satisfiability, implementation of answer set programming systems

On the Relation between Answer Set and SAT Procedures (or, between smodels and cmodels)

Marco Maratea (University of Genova, I)

Answer Set Programming (ASP) is a declarative paradigm for solving search problems. State-of-the-art systems for ASP include smodels, dlv, cmodels, and assat.

Our goal is to study the computational properties of such systems, e.g., to formally characterize under which conditions different systems have a same behavior. We begin our study with smodels and cmodels. We show that though these two systems are apparently different, they are equivalent on a significant class of programs, called tight. By equivalent, we mean that they explore search trees with the same branching nodes, (assuming, of course, a same branching heuristic).

Given our result and that cmodels search engine is based on the Davis Logemann Loveland procedure (DLL) for propositional satisfiability (SAT), we are able to establish that many of the properties holding for DLL also hold for cmodels and thus for smodels.

For instance we show that there exist classes of formulas which are exponentially hard for cmodels, but “easy” for smodels.

We also discuss how our results extend to other systems.

Finally, we extended cmodels in order to obtain a unique platform with a variety of reasoning strategies, and conducted an extensive experimental analysis, on tight and non tight programs. Our results show that the combinations of reasoning strategies that currently dominate in SAT, each in a different class of problems, also lead to the best performances in ASP in the same classes.

Keywords: Answer set programming, automated reasoning, propositional satisfiability

Joint work of: Giunchiglia, Enrico; Maratea, Marco

Normal Form Theorem for Logic Programs with Cardinality Constraints

Victor W. Marek (University of Kentucky, USA)

We discuss proof schemes, a kind of context-dependent proofs for logic programs. We show usefulness of these constructs both in the context of normal logic programs and their generalizations due to Niemela and collaborators. As an application we show the following result. For every cardinality-constraint logic program P there is a logic program P' with the same heads, but with bodies consisting of atoms and negated atoms such that P and P' have same stable models. It is worth noting that another proof of same result can be obtained from the results by Lifschitz and collaborators.

Keywords: Proof scheme, cardinality constraints

Joint work of: Marek, Victor W.; Remmel, Jeffrey B.

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2005/259>

Algorithms for ID-Logic Model Generation

Maarten Mariën (University of Leuven, B)

We investigate model generation techniques for propositional ID-Logic. We develop a theoretical framework for model generation in this logic, present an algorithm and prove its correctness. As ID-Logic is an integration of classical logic and logic programming, our algorithm integrates techniques from SAT and ASP. We report on a prototype system, called MidL. We have implemented several variants of the basic algorithm in MidL and compare their performances.

Keywords: ID-Logic, model generation, SAT, answer set programming

Joint work of: Mariën, Maarten; Mitra, Rudradeb; Denecker, Marc

Computer Aided Security Requirements Engineering with ASP

Fabio Massacci (Università di Trento, I)

Security Requirements Engineering is emerging as a branch of Software Engineering, spurred by the realization that security must be dealt with early on during the requirements phase. Methodologies in this field are challenging as they must take into account subtle notions such as trust (or lack thereof), delegation, and permission; they must also model entire organizations and not only systems-to-be.

In this talk I'll introduce Secure Tropos, a formal framework for modeling and analyzing security requirements developed jointly with P. Giorgini, J. Mylopoulos and N. Zannone that extends the state-of-the-art Tropos/i* Requirements Engineering Methodology. Secure Tropos is founded on few notions: actors, goals, dependency, ownership, trust, and delegation (of permission and execution). I'll show how ASP can be used to provide a formal for these notions and for supporting the security engineer in the design.

In the last part of the talk, I'll demonstrate the Computer Aided Security Requirements Engineering Tool that actually supports the methodology and uses the ASP solvers as reasoning back-ends (See <http://sesa.dit.unitn.it/sttool/>).

Keywords: Security, asp

See also: <http://www.ing.unitn.it/~massacci/SecureTropos>

Tools for Modeling and Solving Search Problems

Artur Mikitiuk (Univ. of Texas at Tyler, USA)

We will describe a language of propositional logic with pseudo-boolean constraints to model search problems that are specified in terms of boolean combinations of pseudo-boolean constraints. We will also present software tools (a grounder program `psgrnd`) that allow one to use SAT or SAT(PB) solvers to compute solutions to instances of search problems represented in the language of our logic.

Keywords: SAT, pseudo-boolean constraints, grounder

Joint work of: East, Deborah; Iakhiaev, Mikhail; Mikitiuk, Artur; Truszczynski, Mirosław

ALBA: A Logical Framework for Intelligent Profiling

Alessandra Mileo (Universit di Milano, I)

The growth of the World Wide Web emphasized the need of providing some tools in charge of helping the user to find interesting information on the Web, by determining which Web Pages on a particular topic would be interesting for him/her.

Development of such tools is strictly related to the problems of i) categorization and ii) user profiling. I want to focus on how user interests can be summarized and expressed in a profile, and how this profile can be used to help a user browsing the Web.

All previous approaches addressing the problem of learning and revising user profile, used similar concept of what a "profile" is.

In the project presented here, a user profile can be defined as a set of user-specific preferences settled by the user himself, telling us his/her interests in a set of topics.

The solution I propose is however focused on "intelligent user profiling".

By the term "intelligent" here I mean that the profile is structured in such a way to be automatically updated and learned as user's interests become more and more clear by his behaviour.

In other words, an intelligent profile is a set of preferences that are initially given by the user, and that will be automatically revised by an inference engine which "learns" user's interests on the basis of user's browsing behaviour. This may allow to capture not only explicit preferences, but also some hidden interests ignored even by the user.

My idea is to introduce an alternative way of representing user preferences in a profile through Answer Set Programming, so as to merge some of the advantages of the others approaches and in the meantime, overcome some of their main limitations.

User's browsing behaviour will be automatically captured in a log file; information are then extracted in form of logic rules and used by an Asp Learner to update the profile.

This learning procedure does not need for large data set to train on, in order to acquire knowledge about user's preferences; in fact the initial profile provided by the user already gives the system an idea of what he likes/dislikes, and this knowledge is then revised just by analysing user's behaviour and by selecting some "meaningful" actions among all actions user did within a certain time slot.

In this way, the system is supposed to save user time: it learns faster what a user is interested in, just by carefully observing user's significant reactions and behaviours, so as to be able to suggest where interesting information can be found.

Keywords: Profiling, machine learning, preferences, cluster, log file

Progress and Problems in SAT Solving

David Mitchell (Simon Fraser University, CDN)

Until the 1970's, propositional logic was considered a subject in the foundations of theorem proving. SAT became important to computer science in the 1970's, but was almost universally taken to be of purely theoretical interest until very recently.

Today, the best SAT solvers are effective enough that they have become a standard industrial tool in hardware verification, are used as the engines in solvers for a variety of other representation languages.

We describe the main ideas and techniques that have gone into the development of the family of modern SAT solvers which includes Chaff and siege. Our account is historical, beginning with foundations from the early 20th century. We emphasize the role of resolution, which provides a uniform way to view algorithm behaviour, as well as the importance of implementation details.

We also mention some current, as well as historical, problems.

Possibilistic Stable Models

Pascal Nicolas (Université d'Angers, F)

We present the main lines of a new framework that we have defined in order to improve the knowledge representation power of Answer Set Programming paradigm.

Our proposal is to use notions from possibility theory to extend the stable model semantics by taking into account a certainty level, expressed in terms of necessity measure, on each rule of a normal logic program.

First of all, we introduce possibilistic definite logic programs and show how to compute the conclusions of such programs both in syntactic and semantic ways.

The syntactic handling is done by help of a fix-point operator, the semantic part relies on a possibility distribution on all sets of atoms and the two approaches are shown to be equivalent.

In a second part, we define what is a possibilistic stable model for a normal logic program, with default negation. Again, we define a possibility distribution allowing to determine the stable models.

We end our presentation by showing how we can use our framework to addressing inconsistency in Answer Set Programming.

Keywords: Non monotonic reasoning, uncertainty, possibility theory

Joint work of: Nicolas, Pascal; Garcia, Laurent; Stéphan, Igor

Boolean Equation Systems and Answer Set Programming

Ilkka Niemelä (Helsinki University of Technology, FIN)

We study the applicability of answer set programming to solving Boolean equation systems. We develop a novel characterization of solutions for variables in disjunctive and conjunctive Boolean equation systems. Based on this we devise a mapping from Boolean equation systems with alternating fixed points to normal logic programs such that the solution of a given variable of an equation system can be determined by the existence of a stable model of the corresponding logic program.

The technique can be used to model check alternating formulas of modal mu-calculus.

Keywords: Answer set programming, alternating Boolean equation systems, modal mu-calculus

Full Paper: <http://www.springerlink.com/openurl.asp?genre=article&issn=0302-9743&volume=3392&spage=134>

Joint work of: Keinänen, Misa; Niemelä, Ilkka

Circ2dlp: Translating Parallel Circumscription into Disjunctive Logic Programming

Emilia Oikarinen (Helsinki University of Technology, FIN)

In contrast to stable model semantics of disjunctive logic programs where all atoms are subject to minimization, parallel circumscription enables the use of varying and fixed atoms in addition to atoms being minimized. Parallel circumscription allows elegant and concise representation of certain problems that are awkward to formalize using stable model semantics. We have implemented a translator called circ2dlp for translating parallel circumscription into a disjunctive logic program. Using the translator circ2dlp efficient ASP solvers such as GnT and dlvs can be used to solve problems encoded as parallel circumscription.

Keywords: Parallel circumscription, disjunctive logic programming stable model semantics, translator

Joint work of: Oikarinen, Emilia; Janhunen, Tomi

Action Description of Protein Folding in Answer Set Programming.

Ramon P. Otero (University of La Coruña, E)

Protein folding is a central problem in molecular biology. Proteins are long chains of elements in sequence; elements —called amino-acids— are taken from a set of twenty possible. This chain folds in space to a particular form/shape. It has been shown experimentally that the three-dimensional shape of proteins only depends on the amino-acid sequence (Anfinsen 1973). Characterizing this dependence is the protein folding problem. When observing the 3D shapes of proteins some regular 3D sub-structures can be identified. Segments of the chain fold in a regular helix, 3.6 aminoacids at each turn—alpha helix. Another regular sub-structure is an almost flat and linear shape called beta strand. These sub-structures give rise to an intermediate folding problem called the protein secondary structure prediction problem (PSSP). Almost any method has been applied to this problem, prominently from machine learning. Unfortunately, no predictive model was identified even for reduced subproblems like the identification of alpha helices. This work considers a seldom tried approach viewing folding as a dynamic process, then using an action description for it. In life conditions proteins fold as they are constructed in the ribosome, one amino-acid at a time following the sequence. The hypothesis is that the lack of success in folding prediction may be due to modeling a dynamic domain with unsuitable formalisms. We show some preliminary action descriptions able to predict alpha helices as good as best alternative current methods. This work is also a first step for the application of learning action descriptions (Otero 03) to the problem. This learning method sound and complete for action is able to take into account previous knowledge about the domain during induction, which improves the result.

Keywords: Applications of action descriptions in ASP, learning in ASP

A Logic for Reasoning about Paraconsistent Answer Sets

David Pearce (Universidad Rey Juan Carlos - Madrid, E)

We present an alternative model theory for answer sets based on the possible worlds semantics proposed by Routley (1974) as a framework for the propositional logics of Fitch and Nelson. By introducing a falsity constant or second negation into Routley models, we show how paraconsistent as well as ordinary answer sets can be represented via a simple minimality condition on models.

This means we can define a paraconsistent version of equilibrium logic or paraconsistent answer sets (PAS) for propositional theories. The underlying logic N9 of PAS belongs to the lattice of logics studied by Odintsov (2004). We axiomatise the logic, showing it to be the least conservative extension of the logic of here-and-there with strong negation, representable via the full twist-structure on the 3-element Heyting algebra. The latter means that N9 can be viewed as a 9-valued logic. In addition, we show that the logic suffices to characterise the strong equivalence of programs in the paraconsistent case and can thus serve as a useful mathematical foundation for PAS. We conclude by showing that the logic has the Interpolation Property.

Joint work of: Odintsov, Sergei; Pearce, David

An Application of Revision Programming to von Neumann-Morgenstern Approach in Conflict Resolution

Inna Pivkina (New Mexico State University, USA)

Revision programming is a formalism for describing and enforcing constraints on knowledge bases. This work applies revision programming to conflict resolution.

In a 1944 book that started game theory (and mathematical approach to conflict resolution), von Neumann and Morgenstern proposed the notion of a solution. When the situation changes, the old solution is often no longer a solution, so it needs to be updated. In practical applications, it is usually desirable to keep the solution change "minimal" in some reasonable sense. For a straightforward formalization of this minimality, checking whether a change is minimal is NP-hard.

We show that by representing the notion of a solution as a revision program, we can produce a reasonable notion of minimality for which there exists a feasible algorithm for checking minimality of update.

This representation also allows to find NM-solutions using existing answer set programming solvers.

Keywords: Revision programming, answer set programming, conflict resolution, game theory

Joint work of: Pivkina, Inna; Kreinovich, Vladik

Semantic Web Languages and Semantic Web Services as Application Areas for Answer Set Programming

Axel Polleres (Universität Innsbruck, A)

In the Semantic Web and Semantic Web Services areas there are still unclear issues concerning an appropriate language. Answer Set Programming and ASP engines can be particularly interesting for Ontological Reasoning, especially in the light of ongoing discussions of non-Monotonic extensions for Ontology Languages. Previously, the main concern of discussions was around OWL and Description Logics. Recently many extensions and suggestions for Rule Languages and Semantic Web Languages pop up, particularly in the the context of Semantic Web Services, which involve the meta-data description of Services instead of static data on the Web only. These languages involve SWRL, WSML, SWSL-Rules, etc. I want to give an outline of languages, challenges and initiatives in this area and where I think Answer Set Programming research can hook in.

Keywords: Semantic Web, Semantic Web Services, rule languages, RDF, RDFS, OWL, WSMO, WSML, OWL-S, SWSL, SWSF

Extended Abstract: <http://drops.dagstuhl.de/opus/volltexte/2005/263>

Set Based Logic Programming

Jeffrey Remmel (Univ. California - San Diego, USA)

We propose a set of desiderata for extensions of Answer Set Programming to capture domains where the objects of interest are infinite sets and yet we can still process ASP programs effectively. We propose two different schemes to do this. One is to extend cardinality type constraints to set constraints which involve codes for finite, recursive and recursively enumerable sets. A second scheme to modify logic programming to reason about sets directly. In this setting, we can also augment logic programming with certain monotone inductive operators so that we can reason about families of sets which have structure such a closed sets of a topological space or subspaces of a vector space. We observe that under such conditions, the classic Gelfond-Lifschitz construction generalizes to at least two different notions of stable models.

Keywords: ASP, codes for infinite sets, stable model generalizations

Joint work of: Remmel, Jeffrey; Marek, Victor

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2005/266>

Enumerating Maximal Sets w.r.t. Monotone Property by Minimal Hitting Set Computation

Ken Satoh (NII - Tokyo, J)

In this talk, I give an algorithm which enumerates maximal sets w.r.t. monotone property by using minimal hitting set.

Monotone property is the one where if a set has the property, every subset of the set has the same property. I will talk an application of the algorithm to computing a minimal belief revision and a maximal set of soft constraints.

Keywords: Minimal hitting set, soft constraint, maximal consistent set, monotone property

Joint work of: Satoh, Ken; Uno, Takeaki

Platypus: A Platform for Distributed Answer Set Solving

Torsten Schaub (Universität Potsdam, D)

We propose a model to manage the distributed computation of answer sets within a general framework, incorporating a variety of software and hardware architectures to allow its easy use with a diverse cadre of computational elements. Starting from a generic algorithmic scheme, we develop a platform for distributed answer set computation, describe its current state of implementation, and give some experimental results.

Keywords: Answer set programming, distribution, solver

Joint work of: Gressmann, Jean; Mercer, Robert E.; Schaub, Torsten; Tichy, Richard; Thiele, Sven

On the Distribution of Programs with Stable Models

John Schlipf (University of Cincinnati, USA)

We report on experimental work to determine the the distribution of programs with stable models among randomly generated normal propositional logic programs. Earlier work, with a distribution analogous to the usual distribution for studying 3-SAT, surprisingly showed no non-monotonicity. Here, under what we feel to be a very natural distribution, we get obvious nonmonotonicity. Accordingly, we believe we also get far more insight into properties likely to block having stable models.

Our programs had rules with 0 or 2 positive subgoals and 0 or 1 negative subgoals. This is natural in part because it is easy, given a normal logic program, to generate one in this form where the stable models of the original program are

exactly the projections of stable models of the new program. Also, this distribution, with 0 positive subgoals and 1 negative subgoal in every rule, duplicates earlier work on the distribution of digraphs with centers. In our preliminary work with more possible rule forms, this distribution seemed to reflect most of the complexities in the distribution.

We have (unfortunately for graphing) four parameters: (1) the number of variables, (2) the ratio of rules to variables, (3) the probability that a rule will have positive subgoals, and (4) the probability that a rule will have a negative subgoal.

Our results show sharp nonmonotonic effects in how changing the parameters affects the probability a randomly generated program will have a stable model. We also have preliminary information about what part of the distribution space generates the hardest problems.

Keywords: Normal logic programs, stable models, distribution, probability

Joint work of: Wong, Douglas; Schlipf, John; Truszczynski, Mirek

A Framework for Representing and Solving NP Search Problems

Eugenia Ternovska (Simon Fraser University, CDN)

NP search and decision problems occur widely in application areas, and a number of general-purpose methods for solving them have been developed. The dominant approaches include propositional satisfiability (SAT), constraint satisfaction problems (CSP), and answer set programming (ASP).

We propose a declarative constraint programming framework which we believe combines many strengths of these methods, while addressing weaknesses in each of them. We formalize our approach as a model extension problem, which is based on the classical notion of extension of a structure by new relations.

We show that a parameterized version of this problem captures exactly the NP search problems.

We also discuss features of an effective modelling language suitable for representing problems in our framework.

Joint work of: Ternovska, Eugenia; Mitchell, David

On Solution Correspondences in Answer-Set Programming, I: Strong and Uniform Equivalence for Non-Ground Programs

Hans Tompits (TU Wien, A)

Recent research in nonmonotonic logic programming under the answer-set semantics focuses on different notions of equivalence. In particular, strong and uniform equivalence are proposed as useful tools for optimizing (parts of) a logic program. Whereas most previous research in this direction addressed only ground logic programs (i.e., programs without variables), in this paper, we deal with the more general case of non-ground programs. More specifically, we discuss languages with both finite and infinite vocabularies and provide semantical characterizations capturing the essence of equivalence, generalizing the concepts of SE-models and UE-models, respectively, as originally introduced for propositional programs. We furthermore show that, for infinite vocabularies, uniform equivalence between disjunctive programs is undecidable, and we provide decidability results and precise complexity bounds for strong equivalence, for both finite and infinite vocabularies, and for uniform equivalence for finite vocabularies, thereby correcting a previous complexity bound for strong equivalence from the literature.

Joint work of: Eiter, Thomas; Fink, Michael; Tompits, Hans; Woltran, Stefan

Applying Solvers of Pseudoboolean Constraints to Compute Answer Sets of Programs with Weight Constraints

Miroslaw Truszczynski (University of Kentucky, USA)

We present a method to compute stable models of logic programs with weight atoms. Our method reduces the problem to that of computing models of sets of pseudo-boolean constraints. The reduction is based on generalizations of the concepts of the program completion and a loop formula. A distinct feature of our approach is that it uses off-the-shelf pseudo-boolean constraint solvers as back-end search engines. Experimental results show that the method we developed is often faster than `smodels` and `cmmodels`. It performs especially well in combination with local-search pseudo-boolean constraint solvers.

Keywords: Knowledge representation, logic programming

Joint work of: Liu, Lengning; Truszczynski, Miroslaw

Compiling Propositional Theories into Logic Programs

Agustin Valverde Ramos (Univ. de Malaga, E)

We study reductions of propositional theories to logic programs under answer set semantics, or more accurately equilibrium logic. Specifically we are concerned with the question of how to transform a propositional theory into an equivalent logic program and what are the complexity constraints on this process. We want the transformed program to be equivalent in a strong sense so that theory parts can be transformed independent of the wider context in which they might be embedded. It was only recently established by Cabalar & Ferraris (2004) that propositional theories are indeed equivalent (in a strong sense) to logic programs. Here this result is extended with the following contributions. (i) We show how to effectively obtain an equivalent program starting from an arbitrary theory. (ii) We show that in general such a transformation is NP-hard if we require the resulting program to share precisely the vocabulary or signature of the initial theory. (iii) Extending previous work we show how polynomial transformations can be obtained if one allows the resulting program to contain new atoms. The program obtained is still in a strong sense equivalent to the original theory, and the answer sets of the theory can be retrieved from it.

Keywords: ASP, equilibrium logic, program transformations

Joint work of: Cabalar, Pedro; Pearce, David; Ramos, Agustin Valverde

An Analysis of Dependencies in ID-Logic

Joost Vennekens (University of Leuven, B)

We study properties of ID-logic, an extension of classical logic with non-monotone inductive definitions. The semantics of this formalism can be given an elegant algebraic characterization in the framework of approximation theory, which provides a powerful setting in which to analyse this logic. We present a theoretical analysis of the basic structure of expressions in this logic, based on the fundamental concept of a dependency relation. We use this to generalize certain splitting results for this logic and to clarify the relation between this logic and certain known classes of inductive definitions. These results help to illustrate the use of this logic for knowledge representation. We also complement this semantical analysis with corresponding syntactic criteria.

Keywords: Knowledge representation, mathematical foundations

Joint work of: Vennekens, Joost; Denecker, Marc

Relevance and Forgetting in Answer Set Programming

Kewen Wang (Griffith University - Brisbane, AU)

The study of forgetting for reasoning has attracted considerable attention in AI.

However, much of the work on forgetting, and other related approaches such as independence, irrelevance and novelty, has been restricted to the classical logics. In this talk we will describe a detailed theoretical investigation of the notion of forgetting and relevance in the context of answer set programming. We first provide a semantic definition of forgetting under the answer sets for extended logic programs. We then discuss the desirable properties and some motivating examples. An important result of this study is an algorithm for computing the result of forgetting in a logic program. Furthermore, we present a modified version of the algorithm and show that the time complexity of the new algorithm is polynomial with respect to the size of the given logic program if the size of certain rules is fixed. We show how the proposed theory of forgetting can be used to characterize the logic program updates.

Keywords: Answer sets, logic programs

On Solution Correspondences in Answer-Set Programming, II: A General Framework (and Some Characterisations for the Ground Case)

Stefan Woltran (TU Wien, A)

We introduce a general framework for specifying program correspondence under the answer set semantics.

The framework allows to define different kinds of equivalence notions, including previously defined notions like strong and uniform equivalence, in which programs are extended with rules from a given context, and correspondence is determined by means of a binary relation.

In particular, refined equivalence notions based on projected answer sets can be defined within this framework, where not all parts of an answer set are of relevance.

We study general characterizations of inclusion and equivalence problems, introducing novel semantical structures. Furthermore, we deal with the issue of determining counterexamples for a given correspondence problem, and we analyse the computational complexity of correspondence checking.

Keywords: Equivalence, computational complexity

Joint work of: Eiter, Thomas; Tompits, Hans; Woltran, Stefan

Random Generated Logic Programs

Yuting Zhao (IRST - Trento, I)

We studied the following problems on some classes of random generated logic programs under the answer set semantics: the existence of an answer set and the hardness of finding one answer set, if any. Firstly we showed how the probabilities of having an answer set change with the changing of ratio L/N , where L and N are the number of rules and atoms in a program respectively, in the following classes of logic programs: (i) with fixed body-length rules of 3 and more literals; (ii) with mixed body-length rules of 3 and more literals; (iii) with mixed body-length rules of 1-3 literals. Secondly we indicated the hard-job-region in each class of programs, and conjectured the relations between the hard-job-regions and the above probabilities.

Keywords: Answer set programming, random generated logic program, phase transition, hard-job-region

Social Modelling using Answer Set Programming

Marina de Vos (University of Bath, GB)

In this talk we will discuss the use of the Answer Set Programming (ASP) paradigm for modelling and analysing specifications of normative and institutional properties of multi-agent systems. We provide a general framework for the specification and validation of such systems in ASP. We discuss the features of this approach using the example of the institution of property and the scenario of exchange.

Joint work of: Ciffe, Owen; de Vos, Marina; Padget, Julian